

SAP HANA Platform SPS 12
Document Version: 1.2 – 2018-01-24

SAP HANA Application Lifecycle Management

Content

1	SAP HANA Application Lifecycle Management.	5
1.1	SAP HANA Application Lifecycle Management Home Screen.	7
2	Preparing to Use SAP HANA Application Lifecycle Management.	11
2.1	SAP HANA Application Lifecycle Management Tasks.	12
2.2	SAP HANA Application Lifecycle Management Roles.	14
2.3	Assign User Roles.	16
2.4	Maintain the Delivery-Unit Vendor ID.	18
3	Setting Up the Transport.	20
3.1	Transport Scenarios in SAP HANA Application Lifecycle Management.	22
3.2	Setting Up and Using Native SAP HANA Transport.	26
	Register a System for a Transport Route.	27
	Create a Transport Route.	29
	Start the Transport.	32
	Setting Up Native SAP HANA Transport for SAP HANA Cloud Platform.	33
3.3	Setting Up and Using CTS Transport.	37
	Configure SAP HANA Systems for CTS Transport.	38
	Change CTS Configuration.	40
	Export Delivery Units for CTS Transport.	42
	Export Changelists for CTS Transport.	43
3.4	Remarks on the Transport.	45
3.5	Check the Transport Logs.	47
4	Maintaining Delivery Units.	49
4.1	SAP HANA Delivery Units.	50
4.2	SAP HANA Delivery Unit Naming Conventions.	51
4.3	The SAP HANA Delivery-Unit Lifecycle.	51
4.4	Create a Delivery Unit.	52
4.5	Assign Packages to a Delivery Unit.	54
4.6	Export a Delivery Unit.	55
4.7	Import a Delivery Unit.	57
5	Maintaining Products.	59
5.1	Create a Product.	60
5.2	Assign a Delivery Unit to a Product.	61
6	SAP HANA Change Recording.	63

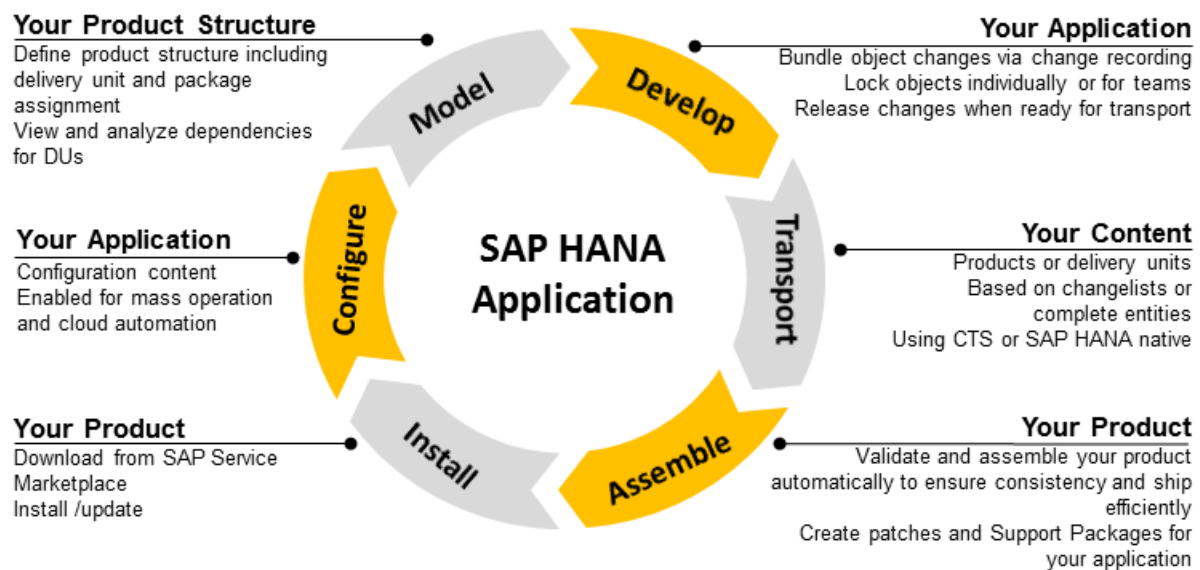
6.1	Technical Details of Initial Change Recording Setup.	64
6.2	Predecessor Changelists in SAP HANA Change Recording.	65
6.3	Consequences of Package - DU Reassignments.	66
6.4	Setup of the Transport Landscape.	68
	Transport Modes in Change Recording.	70
	Recommendations on the Transport of Changelists.	72
	CTS and Changelists.	74
	Native SAP HANA and Selected Changelists.	75
	Native SAP HANA and Complete DU.	76
6.5	Enable SAP HANA Change Recording.	77
6.6	Create Changelists.	78
6.7	Assign Objects to Changelists.	80
	Assign objects in SAP HANA Web-based Development Workbench.	81
	Move objects from one changelist to another.	81
6.8	Approve Contributions to Changelists.	82
6.9	Release Changelists.	83
6.10	Filter and Search for Changelists.	84
	Filter Changelists.	84
	Search for Changelists.	85
7	Assembling Add-On Products and Software Components.	86
8	Installing and Updating SAP HANA Products and Software Components.	88
8.1	Installing and Updating SAP HANA Products.	89
8.2	Installing and Updating SAP HANA Software Components.	91
8.3	Installation and Update Options.	93
9	Installing and Updating Products and Software Components in SAP HANA XS Advanced Model.	95
9.1	Prerequisites and Authorizations.	98
9.2	Set Up a Virus Scan for Installation Archives	98
9.3	Checks Before Installing or Updating Products or Software Components in SAP HANA XS Advanced Model.	99
9.4	Installation and Update Options in XS Advanced Model.	102
9.5	Examples: Installing and Updating Products and Software Components in XS Advanced Model	104
9.6	Display installed Products and Software Components in XS Advanced Model.	108
9.7	Uninstall Products and Software Components in SAP HANA XS Advanced Model.	110
10	Using hdbalm.	114
10.1	Using hdbalm with SAP HANA Cloud Platform.	116
10.2	hdbalm Commands, Options, and Variables.	117
10.3	Enable SSL for hdbalm.	121

10.4	Proxy Support for hdbalm.	122
10.5	hdbalm install Command.	123
	Examples: hdbalm install Command.	125
10.6	hdbalm assemble Command.	127
10.7	hdbalm import Command.	128
10.8	hdbalm transport Command.	129
10.9	hdbalm log Command.	130
10.10	hdbalm product Command.	131
10.11	hdbalm du Command.	132
10.12	hdbalm dependencies Command.	134
10.13	hdbalm package Command.	135
10.14	hdbalm admin Command.	136
11	Configuring SAP HANA Applications with the Process Engine.	138
11.1	Process Engine Roles.	138
11.2	Tutorial: Execute a Configuration Service with Process Engine.	139
11.3	Troubleshooting	142
12	FAQs on SAP HANA Application Lifecycle Management.	143
13	SAP HANA Repository Translation Tool.	147
13.1	SAP HANA Repository Translation Tool (RTT) Parameters.	148
13.2	Configure the Repository Translation Tool.	150
13.3	Create Text-Strings Packages for Translation.	151
13.4	Export Text-Strings Files for Translation.	152
13.5	Import Translated Text-Strings Files.	152
14	Maintaining Translation Text Strings.	154
14.1	Create and Edit Text Translations.	155
	Online Translation Tool Details.	157
14.2	Export and Import Translated Text.	159

1 SAP HANA Application Lifecycle Management

SAP HANA application lifecycle management supports you in all phases of an SAP HANA application lifecycle, from modeling your product structure, through application development, transport, assemble, and install.

The following graphic illustrates the phases in a product lifecycle of an SAP HANA application:



Phases of SAP HANA Application Lifecycle Management

- **Model**
You define your product structure to provide a framework for efficient software development. This includes creating delivery units and assigning packages to delivery units. The delivery units are then bundled in products.
- **Develop**
You perform software development in repository packages. SAP HANA application lifecycle management supports you with change tracking functions.
- **Transport**
You can transport your developed content in different ways according to your needs. You can choose between transporting products or delivery units, based on changelists or complete entities. The transport type can be native SAP HANA transport or transport using Change and Transport System (CTS). You can also export delivery units, and import them into another system.
- **Assemble**

The developed software plus the metadata defined when modeling your product structure as well as possible translation delivery units are the basis for assembling your add-on product. You can also build Support Packages and patches for your product.

- **Install**

You can install SAP HANA products that you downloaded from SAP Support Portal or that you assembled yourself.

- **Configure**

If the SAP HANA product delivers configuration content, you can use the process engine of SAP HANA application lifecycle management to automate configuration tasks.

All phases of SAP HANA application lifecycle management are documented in the *SAP HANA Application Lifecycle Management Guide*. The tasks related to the **Install** and **Configure** phases of SAP HANA application lifecycle management are relevant for system administrators and are therefore also documented in the *SAP HANA Administration Guide*. The tasks related to software development are documented in the *SAP HANA Developer Guide (For SAP HANA Studio)*.

Availability of SAP HANA Application Lifecycle Management

SAP HANA application lifecycle management is installed with SAP HANA as automated content. You can access the SAP HANA application lifecycle management functions in different ways:

- Using the **XS user interface SAP HANA Application Lifecycle Management** which is available in the following places:
 - On the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.
 - Using a link in SAP HANA Web-based Development Workbench.
For example, to open the home screen, choose ► [Navigation Links](#) ► [Lifecycle Management](#) ► in the SAP HANA Web-based Development Workbench Editor tool.
 - Using the context menu in SAP HANA studio.
For example, to open the home screen from, choose ► [Lifecycle Management](#) ► [Application Lifecycle Management](#) ► [Home Screen](#) ► from the context menu for a particular system in the [SAP HANA Administration Console](#) perspective in SAP HANA studio.
 - Using the tile catalog in SAP HANA cockpit.
There are tiles available both in the [SAP HANA Application Lifecycle Management](#) and in the [SAP HANA Application Installation and Update](#) groups in SAP HANA cockpit that you can customize according to your needs.
- Using the **command line tool** `hdbalm`.
The file is shipped with the SAP HANA client installation. If you leave the default installation options unchanged `hdbalm` is located in the `..\\sap\\hdbclient` directory.

You cannot perform all application lifecycle management tasks with one tool. For example, assembling products and software components can only be done using the `hdbalm` tool, whereas the full set of transport functions is available only in the XS user interface. Whenever a function is available in the XS user interface it is documented there. When used in SAP HANA studio, the functions are the same as in the XS user interface. Therefore, these options are not separately documented.

Note

For information about the SAP HANA platform lifecycle management tools, see the *SAP HANA Server Installation and Update Guide* and *SAP HANA Platform Lifecycle Management* in the *SAP HANA Administration Guide*.

Related Information

[Using hdbalm \[page 114\]](#)

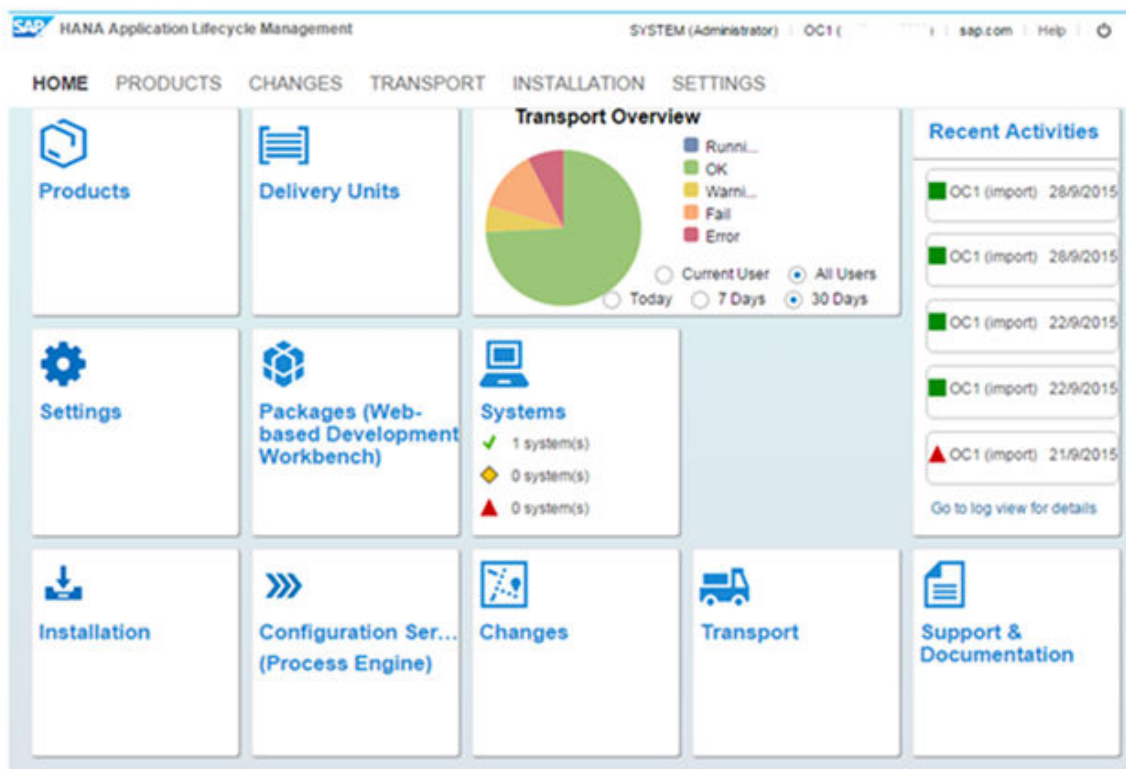
1.1 SAP HANA Application Lifecycle Management Home Screen

The home screen of the SAP HANA Application Lifecycle Management XS user interface provides you with a single point-of-access to all application lifecycle management functions required for SAP HANA. You open SAP HANA Application Lifecycle Management using a Web browser.

You can either click on the tiles or the tabs on the top of the screen to access the required functions.

The row on the very top of the screen contains the title, and further information, such as the log-on user and role, the ID and the Web server host of the SAP HANA system, as well as the vendor name. The [Help](#) link provides links to context-sensitive help that comes with the application.

The row on top of the tiles provides links to the most important views that are also available using the links.



- [Maintaining Products \[page 59\]](#)
- [Maintaining Delivery Units \[page 49\]](#)
- [Check the Transport Logs \[page 47\]](#)
- [Setting Up the Transport \[page 20\]](#)
- [Register a System for a Transport Route \[page 27\]](#)
- [Installing and Updating SAP HANA Products and Software Components \[page 88\]](#)
- [Configuring SAP HANA Applications with the Process Engine \[page 138\]](#)
- [SAP HANA Change Recording \[page 63\]](#)
- [Setting Up the Transport \[page 20\]](#)

Tiles in SAP HANA Application Lifecycle Management Home Screen

Tile Title	Description	More Information
<i>Products</i>	Opens the <i>Products</i> view that allows you to create and manage products and product instances, as well as assigning delivery units.	Maintaining Products [page 59]
<i>Delivery Units</i>	Opens the <i>Delivery Units</i> view that allows you to create and manage delivery units and to assign packages.	Maintaining Delivery Units [page 49]

Tile Title	Description	More Information
Transport Overview	Opens the Logs view that allows you to check the transport logs.	Check the Transport Logs [page 47]
Recent Activities	Provides an overview of the last transport activities that were performed in this system. Clicking on an activity opens the log entry of the activity in the Logs view.	Check the Transport Logs [page 47]
Settings	Opens the Settings view that allows you to enable or disable change recording and to establish transport settings for your system.	Setting Up the Transport [page 20] Enable SAP HANA Change Recording [page 77]
Packages	Opens the SAP HANA Web-based Development Workbench that allows you to create and manage SAP HANA Repository packages.	<i>Maintaining Repository Packages in SAP HANA Developer Guide for SAP HANA Web Workbench</i>
Systems	<p>Opens the Systems view that allows you to register HTTP destinations as source systems for the transport.</p> <div> <p>i Note</p> <p>This tile is visible only if the system is configured for native SAP HANA transport.</p> </div>	Register a System for a Transport Route [page 27]
Installation	Opens the Install and Update SAP HANA Products and Software Components view.	Installing and Updating SAP HANA Products and Software Components [page 88]
Configuration Services (Process Engine)	Opens the process engine of SAP HANA application lifecycle management that allows you to start and monitor configuration services that are provided by the application.	Configuring SAP HANA Applications with the Process Engine [page 138]
Changes	<p>Opens the Changes view that allows you to create and manage changelists.</p> <div> <p>i Note</p> <p>The link on this tile only works if change recording is enabled.</p> </div>	SAP HANA Change Recording [page 63]

Tile Title	Description	More Information
<i>CTS Export</i>	<p>Opens the <i>CTS EXPORT</i> view that allows you to export delivery units or changelists for CTS transport.</p> <p>i Note This tile is visible only if the system is configured for CTS transport.</p>	<ul style="list-style-type: none"> • Export Delivery Units for CTS Transport [page 42] • Export Changelists for CTS Transport [page 43]
<i>Transport</i>	<p>Opens the <i>Transport</i> view that allows you to create transport routes that specify the configuration details for the native SAP HANA transport, such as the source system and which content is to be transported. You can also start the transport from there.</p> <p>i Note This tile is visible only if the system is configured for native SAP HANA transport.</p>	Create a Transport Route [page 29]
<i>Support & Documentation</i>	Opens a dialog box that contains support information about the SAP HANA application lifecycle management XS user interface and links to documentation.	

2 Preparing to Use SAP HANA Application Lifecycle Management

Before you can use SAP HANA Application Lifecycle Management, you must assign predefined roles to users who need to perform application lifecycle management tasks and set the content vendor ID for the SAP HANA system.

Prerequisites

To use the SAP HANA Application Lifecycle Management tool, you must ensure that the following prerequisites are met:

- An SAP HANA system is available.
- SAP HANA XS is up and running on the SAP HANA system.
- You have system privileges on the SAP HANA system (for example, to add users).

For the prerequisites required to use `hdbalm`, see the *Prerequisites* section in the *Using hdbalm* topic. The link to the topic is in the *More Information* section.

Context

With SAP HANA Application Lifecycle Management, your authorization level determines which tasks you are able to perform. Authorization levels are granted by assigning the appropriate role, for example, `sap.hana.xs.lm.roles::Administrator`. If you do not have the required level of access, in the Web-based tool, certain buttons are disabled, and certain options are hidden. In `hdbalm`, you receive an error message informing you that authorization is missing.

Procedure

1. Assign copies of the necessary roles to the users who perform application lifecycle management tasks for example, SAP HANA application lifecycle management administrator tasks.

Note

This step must be performed in the SAP HANA studio or SAP HANA Web-based Development Workbench by a user with administrator privileges.

Access to features and options in the SAP HANA Application Lifecycle Management is based on user privileges, which are assigned in user roles, for example, administrator or transport manager.

➔ Recommendation

Do not use the repository roles delivered with SAP HANA directly, but instead use them as templates for creating your own roles. Furthermore, if repository package privileges are granted by a role, we recommend that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

2. Set the Vendor ID.

The vendor ID sets the namespace in SAP HANA where your application development takes place, for example, "sap.com" or "com.mycompany".

i Note

The namespace `sap` is restricted; you must not develop your own applications in this namespace. Place your packages in your own namespace.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Using hdbalm \[page 114\]](#)

2.1 SAP HANA Application Lifecycle Management Tasks

The responsibility for the tasks you perform with the SAP HANA Application Lifecycle Management is shared between dedicated lifecycle management roles.

The responsibility for common application-lifecycle management performed with the SAP HANA Application Lifecycle Management is shared between the various lifecycle management roles, which must be copied and assigned to the SAP HANA users who start the SAP HANA Application Lifecycle Management. For example, the [Administrator](#) role enables access to all options and tools in the SAP HANA Application Lifecycle Management. To start a transport operation based on a defined route, you only need the privileges assigned with the user role [ExecuteTransport](#). The [Display](#) role enables a user to view details of the delivery units, routes, and transports, but cannot make any changes.

➔ Recommendation


Do not use the repository roles delivered with SAP HANA directly, but instead use them as templates for creating your own roles. Furthermore, if repository package privileges are granted by a role, we recommend that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

The following table provides an overview of the most common tasks performed with the SAP HANA Application Lifecycle Management and which user role is required to perform the specified task.

Note

The SAP HANA systems you use as destination for delivery unit (DU) transport operations as well as the SAP HANA users required to complete the operation must be set up by an SAP HANA administrator.

SAP HANA Application Lifecycle Management Task/Role Matrix

Task	Administrator	Execute Transport	Display	Developer	Development Expert
Create product, DU, and so on	X				
Export DU	X	X			
Import DU	X				
Add transport destination	X				
Setup transport route	<div>  Note For this task, you need additional roles as described in <i>SAP HANA Application Lifecycle Management Roles</i>. </div>				
Execute transport	X	X			
View log files for transport, export, import operations	X	X	X	X	X
Configure change recording	X				
Create / delete changelists	X				X
Assign objects to changelists	X			X	X
Approve contributions to changelists	X			X (your own contributions)	X (your own and foreign contributions)
Release changelists	X				X

Task	Administrator	Execute Transport	Display	Developer	Development Expert
Add / delete contributors to changelists	X				X
Install/Update SAP HANA products and software components	X				

Related Information




[SAP HANA Application Lifecycle Management Roles \[page 14\]](#)

2.2 SAP HANA Application Lifecycle Management Roles

To grant users the privileges they require to perform tasks with the SAP HANA Application Lifecycle Management, you must assign them the privileges of one or more dedicated application lifecycle management roles.

The following table lists the roles that can be used as templates for users who want to perform lifecycle-management-related tasks using SAP HANA Application Lifecycle Management tool. The roles are hierarchical and interlinked. For example, by default, the [Administrator](#) role grants the privileges included in all other roles; the [ExecuteTransport](#) role grants the privileges assigned in the [Transport](#) and [Display](#) roles.

Note

Some lifecycle-management tasks require interaction with external tools, and this requires additional privileges, which you can grant by assigning the appropriate roles. For example, to register an HTTP destination as part of the setup of a transport route, you need to supply logon credentials for an existing technical user on the source system - the system defined in the HTTP destination configuration. To maintain logon credentials, you can use the [SAP HANA XS Administration Tool](#), which requires the privileges granted by the delivered role `sap.hana.xs.admin.roles::HTTPDestAdministrator`. To display all available roles, start the SAP HANA studio, and in the [SAP HANA Systems](#) view, expand the node  [Security](#)  [Roles](#) .

Recommendation

Do not use the repository roles delivered with SAP HANA directly, but instead use them as templates for creating your own roles. Furthermore, if repository package privileges are granted by a role, we recommend that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted

on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

SAP HANA Application Lifecycle Management Roles

Role Name	Description
<code>sap.hana.xs.lm.roles::Administrator</code>	Full read/write access to all the features in the SAP HANA Application Lifecycle Management tool, including the access privileges granted to all other user roles available in the SAP HANA Application Lifecycle Management, for example, Display , ExecuteTransport , and Transport .
<code>sap.hana.xs.lm.roles::Developer</code>	A role based on this role template is required if change recording is activated: Enables the user to work on a changelist to which he is assigned and to approve own contributions to the changelist. This role includes the privileges of the Display role.
<code>sap.hana.xs.lm.roles::DevelopmentExpert</code>	A role based on this role template is required if change recording is activated: Enables the user to perform all actions involved in change recording (for example, create, assign objects to, release, delete, assign other users to a changelist, approve own or foreign contributions). This role includes the privileges of the Display and the Developer roles.
<code>sap.hana.xs.lm.roles::Display</code>	View-only access; some features and options are hidden. A user with a role based on this role template can view all information available but cannot make any changes or trigger any transport operations.
<code>sap.hana.xs.lm.roles::ExecuteTransport</code>	Users with a role based on this role template can view all information as well as trigger predefined transport operations. However, they cannot register or maintain systems, create transport routes, or edit details of a product, a delivery unit, or a package.
<code>sap.hana.xs.lm.roles::Transport</code>	For technical users only. Do not assign a role based on this role template to normal users; the required privileges are part of the ExecuteTransport role. The Transport role grants the privileges required for export or import actions during a transport operation. The credentials and privileges of a technical user with the Transport role cannot be used for interactive logons, for example, to start the SAP HANA Application Lifecycle Management.
<code>sap.hana.xs.lm.roles::SLP_display</code>	For technical users used for HTTP-based deployment when using CTS Transport . Users with a role based on this role template can perform all supported read requests for SL protocol services.
<code>sap.hana.xs.lm.roles::SLP_CTS_deploy_admin</code>	For technical users used for HTTP-based deployment when using CTS Transport . Users with a role based on this role template can perform all supported requests for CTS Deploy SL protocol service.
<code>sap.hana.xs.lm.roles::SLP_CTS_ping_admin</code>	For technical users used for HTTP-based deployment when using CTS Transport . Users with a role based on this role template can perform all supported requests for CTS Ping SL protocol service.

Additional Roles

Role Name	Description
<code>sap.hana.ide.roles::EditorDeveloper</code>	<p>Inspect, create, change, delete and activate SAP HANA repository objects</p> <p>A role based on this role template is required when you select the Packages tile to maintain SAP HANA Repository Packages in SAP HANA Web-based Development Workbench.</p> <p>To create new packages, additional privileges may be required. For more information, see <i>Define Repository Package Privileges</i> in the <i>SAP HANA Developer Guide for Web-based Development Workbench</i>.</p>
<code>sap.hana.xs.admin.roles::HTTPDestAdministrator</code>	<p>Full access to HTTP destination configurations (display and edit)</p> <p>A role based on this role template is required when you register a system for a transport route.</p>
<code>sap.hana.xs.admin.roles::RuntimeConfAdministrator</code>	<p>Full access to the configuration settings for SAP HANA XS application security and the related user-authentication providers</p> <p>A role based on this role template is required when you register a system for a transport route.</p>

Related Information

[Register a System for a Transport Route \[page 27\]](#)

[Process Engine Roles \[page 138\]](#)

2.3 Assign User Roles

Access to features and functionality in SAP HANA Application Lifecycle Management is based on roles and privileges; the role you have determines the tasks you can perform.

Prerequisites

To assign privileges to users of the SAP HANA Application Lifecycle Management, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have administrator/system privileges on the SAP HANA system (for example, you can add/maintain database users).

Context

In the SAP HANA Application Lifecycle Management, the availability of features, screens, tabs, and UI controls (for example, [Add](#), [Edit](#), [Save](#), or [Delete](#) buttons) is based on user privileges. For the sake of convenience, the specific privileges required to use the features provided with a particular tool have been collected into a selection of specific roles, which you can use as a template to assign the relevant privileges to the user who needs to use a particular tool.

Note

To start the SAP HANA Application Lifecycle Management, you must have a role based on the the privileges granted by one of the dedicated ALM roles, for example, `sap.hana.xs.lm.roles::Display`.

To assign the required privileges to people who want to use the features provided by the SAP HANA ALM tool, perform the following steps:

Procedure

Create a new user or assign a copy of the required application lifecycle management role to an existing user who needs to perform a specific task.

For more information on the how to assign roles to users, see *Provisioning Users* in the *SAP HANA Administration Guide*. For more information on the available application lifecycle management roles, see *SAP HANA Application Lifecycle Management Roles*. The links can be found in the *Related Information* section.

Recommendation

Do not use the repository roles delivered with SAP HANA directly, but instead use them as templates for creating your own roles. Furthermore, if repository package privileges are granted by a role, we recommend that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

The user can now use SAP HANA application lifecycle management to perform the required task.

Related Information

[SAP HANA Application Lifecycle Management Roles \[page 14\]](#)

[SAP HANA Application Lifecycle Management \[page 5\]](#)

2.4 Maintain the Delivery-Unit Vendor ID

In SAP HANA, the [vendor ID](#) is used primarily to define the identity of the company developing a software component that it plans to ship for use with SAP HANA, for example, "sap.com". To create a delivery unit, it is a prerequisite to maintain a vendor ID in your system.

Prerequisites

To set the vendor ID, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have been assigned the privileges granted by a role based on the SAP HANA XS `sap.hana.xs.lm.roles::Administrator` user role template.

Context

Before creating your own first delivery unit, you must set the identity of the vendor in the development system's configuration. To maintain details of the delivery-unit vendor ID, perform the following steps:

Procedure

1. Start the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

Note

To start the SAP HANA Application Lifecycle Management, you must use the logon credentials of an existing database user, who has the appropriate user role assigned.

2. Choose the [SETTINGS](#) tab.
3. Maintain details of the vendor ID.

In the [SETTINGS](#) tab, perform the following steps:

- a. Choose [Change Vendor](#).
 - b. In the [Set Vendor](#) dialog, enter the name of the new vendor, for example, [mycompany.com](#).
 - c. Choose [OK](#) to save the changes.
- The new vendor ID appears in the [Vendor](#) box.

Note

The vendor ID is required to create a delivery unit.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

3 Setting Up the Transport

You can choose to perform transports in native SAP HANA mode using transport routes or using the Change and Transport System (CTS).

Prerequisites

To set up the transport you want to use for the entire SAP HANA system, you must ensure the following prerequisites are met:

- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role template.
- You have the privileges granted by roles based on the following SAP HANA XS role templates:
 - `sap.hana.xs.admin.roles::HTTPDestAdministrator`
 - `sap.hana.xs.admin.roles::RuntimeConfAdministrator`
- You have decided which transport scenario you want to use for this system, [Native SAP HANA transport](#) or [CTS Transport](#). For more information on the scenarios, see *Transport Scenarios in SAP HANA Application Lifecycle Management*. The link to the topic is in the *Related Information* section.

Context

Use the following steps to set up your transports using either a native SAP HANA system to transport to a single system or using CTS to transport through a transport landscape defined in CTS.

Note

Bear in mind that exports for native SAP HANA are executed on the target system while exports using CTS are started on the source system.

Note

If you want to transport SAP HANA content for SAP HANA applications that are closely connected with ABAP applications in terms of content (ABAP for SAP HANA applications), you can also use SAP HANA Transport for ABAP (HTA) as transport tool.

For more information, search for *SAP HANA Transport for ABAP* or *Transport Scenarios for SAP HANA Content* documentation in the SAP NetWeaver documentation on SAP Help Portal at <https://help.sap.com>.

You usually perform these steps once after you have set up the system.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. Choose the *SETTINGS* tab.
3. In the *Transport* section, make the selection that suits your needs.
 - Ensure that *Enable Native SAP HANA Transport* is selected, if you want to use this option. This is the default setting.
 - Select *Enable CTS Transport*, if you want to use this option.
In the *Switch Transport Mode* popup, choose *Yes* to confirm your choice.

Results

If you have selected *CTS Transport*, then the *TRANSPORT* tab changes to *CTS EXPORT*.

If you have changed from *CTS Transport* to *Native SAP HANA Transport*, then the *CTS EXPORT* tab changes to *TRANSPORT*.

To transport objects using *CTS Transport* or *Native SAP HANA Transport*, you must make further settings. Follow the instructions for the transport type that you want to use.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Setting Up and Using Native SAP HANA Transport \[page 26\]](#)

[Setting Up and Using CTS Transport \[page 37\]](#)

[Transport Scenarios in SAP HANA Application Lifecycle Management \[page 22\]](#)

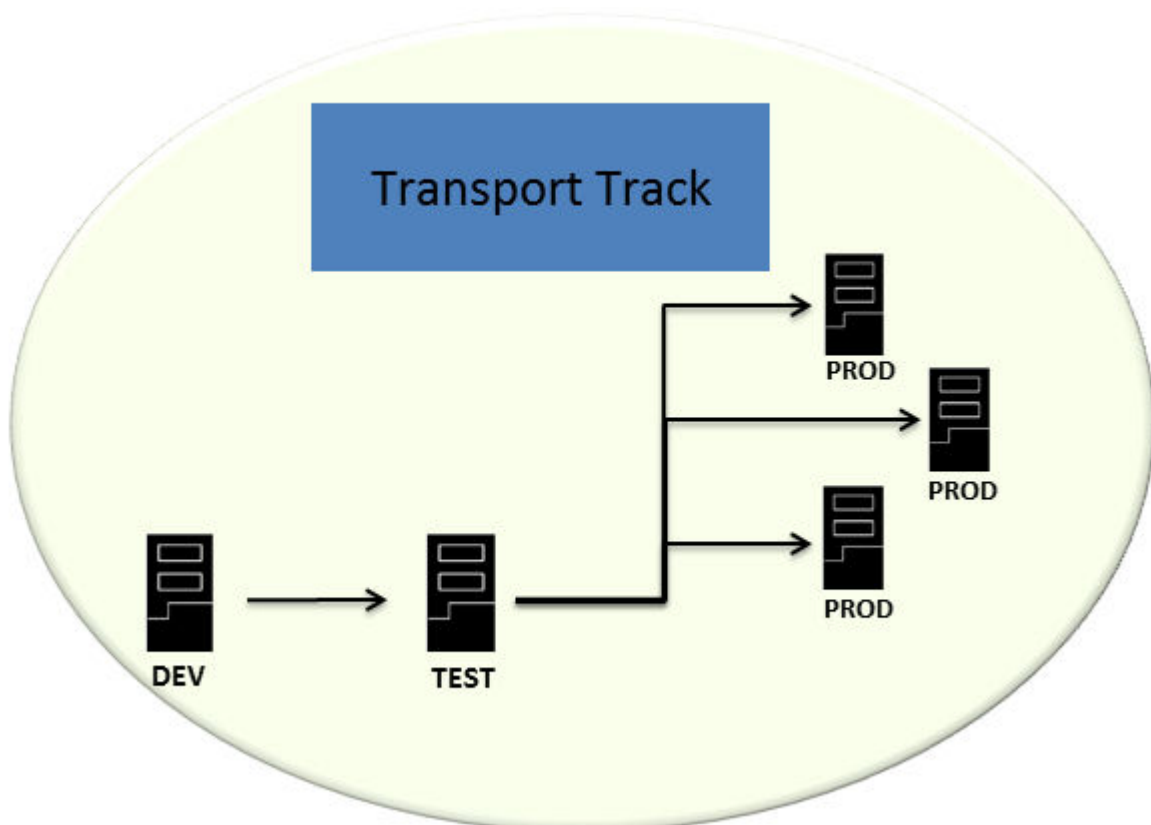
3.1 Transport Scenarios in SAP HANA Application Lifecycle Management

When you develop SAP HANA applications, you must decide on the transport scenario that you want to use and its impact on the setup of the transport landscape, as well as on whether or not you want to use change recording.

Transport Scenarios

A transport track often consists of a development system, a test system, and one or multiple production systems.

The development system is characterized by ongoing development, many changes, a lot of versions, and so on. It is not used productively. After transporting the development entities to the test system, they can be tested there. After successful testing, the software can be transported to the production systems. In the production systems, the software is not changed. Instead, a defined state of the development entities is used productively in these systems.



There are different scenarios available for transport in SAP HANA application lifecycle management: **Native SAP HANA transport** and **Change and Transport System (CTS) transport**. The following table provides a quick overview of the scenarios available. Links to more information are located in the *Related Information* section of this topic.

Transport Scenarios

Transport Scenario	More information
Native SAP HANA	<p>This scenario is suitable for the transport of native SAP HANA objects in which no ABAP development is involved and in which no other non-SAP HANA transports are done using CTS. This scenario is useful for smaller transport tracks with about 3-to-5 systems.</p> <p>You can use native SAP HANA transport to transport content from development to test and from test to production. Using native SAP HANA transport for the entire transport track, including the transport to the production systems, offers the following advantages:</p> <ul style="list-style-type: none"> • You have one tool in place that you can use for the entire transport track. • You can transport both products and delivery units. • If you use change recording, you can transport very small changes through this track using individual changelists. <div style="background-color: #fff9c4; padding: 10px; margin-top: 10px;"> <p>i Note</p> <p>With native SAP HANA transport, performing an import means that you are on the target system and you pull an export from the source system using a configured transport route. Therefore, each import in a production system triggers a separate export from the test system. If you have multiple production systems, you must ensure that the test system remains unchanged while you are performing imports in the production systems. If the test system changes during this process, then the multiple imports into several production systems can result in different states of the software in these systems. Bear in mind that you need to trigger the import separately on each target system.</p> <p>If you cannot guarantee that the test system remains unchanged, you can opt to assemble the development into an installable archive on the test system, and install it in the production systems, as described in the <i>Distribution using assembly and installation</i> section of this topic.</p> </div> <p>For more information on native SAP HANA transport, see <i>Set Up and Use Native SAP HANA Transport</i>. For more information on the setup of the transport landscape, if you have enabled change recording, see <i>Setup of the Transport Landscape</i>.</p>

Transport Scenario	More information
CTS	<p>This scenario is suitable for the transport of native SAP HANA objects in system landscapes in which the Enhanced CTS is already in use. This means that CTS is used to manage transports of other non-ABAP and non-SAP HANA applications. This scenario is particularly well suited for transporting SAP HANA content as part of an SAP solution (such as SAP BI, Mobile, and so on), since you often need to create a connection between the SAP HANA content and other non-ABAP and non-SAP HANA development objects.</p> <p>CTS is integrated into the SAP tools for change control (Change Request Management and Quality Gate Management of SAP Solution Manager).</p> <p>You can use CTS transport to transport SAP HANA content through the transport track, from development to test and from test to production. Using CTS transport for the entire transport track, including the transport to the production systems, offers the following advantages:</p> <ul style="list-style-type: none"> • You have one tool in place that you can use for the entire transport track. • You have a defined landscape (in CTS). • If you use change recording, you can transport very small changes through this track using individual changelists. <div style="background-color: #fff9c4; padding: 10px; margin-top: 10px;"> <p>i Note</p> <p>If you use CTS transport, you can only transport delivery units, because CTS transport does not support the transport of products.</p> </div> <p>For more information on CTS transport, see <i>Set Up and Use CTS Transport</i>. For more information on the setup of the transport landscape if you have enabled change recording, see <i>Setup of the Transport Landscape</i>.</p>

i Note

To bundle transports of ABAP and SAP HANA objects, for example, for SAP HANA content for ABAP for SAP HANA applications, you can use HANA Transport Container (HTC) or SAP HANA transport for ABAP (HTA), depending on your SAP NetWeaver release. For more information, search for *Transport Scenarios for SAP HANA Content* in the SAP NetWeaver documentation on SAP Help Portal at <http://help.sap.com>.

Distribution using assembly and installation

In addition to the transport scenarios available, SAP HANA application lifecycle management offers the possibility to assemble the developed software and install it in the production systems. This way, you have a well-defined state of the software including all the metadata, such as product, delivery unit, and support package versions in the production systems.

You can use this option in the following situations:

- You use CTS transport, but you need to transport products to the production system.
- You use native SAP HANA transport and you want to guarantee maximum consistency in the production systems.
- Your production systems cannot be reached using transport routes (CTS or SAP HANA application lifecycle management transport routes).

You can assemble the software in the test system or in an additional system, the consolidation system, for example. To do this, you must model the delivery unit or the product in the system in which you perform the assembly to supply the required metadata, such as version information and dependencies, for example. Assembly is supported for delivery units and not for individual objects or changelists. Once the metadata is available in the system you can perform the assembly using the commandline tool `hdbalim`.

After performing installation tests, you can install the assembled software in the production systems. You install the software separately in each production system.

➔ Recommendation

If it is important for you to have the information about versions in your production systems, use assembly and installation to distribute software to the production systems. If you want to supply your production systems with new software at regular intervals, use one of the transport scenarios (CTS or native SAP HANA).

Change Recording

You need to decide whether you want to record your changes in the development system and transport them using your transport track. The following table provides a quick overview of the change recording options. Links to more information are located in the *Related Information* section of this topic.

Options for Change Recording

Landscape Option	Description
Change recording enabled	<p>The change recording function is delivered with the SAP HANA server installation and provides a light-weight and out-of-the-box method to keep track of your software changes. If you use change recording, you use SAP HANA Web-based Development Workbench or SAP HANA studio for your SAP HANA development. Your software changes are recorded in changelists and you use one of the transport scenarios described above to transport your development entities.</p> <p>If you use change recording, see <i>Setup of the Transport Landscape</i> for the recommended landscape scenarios.</p>
Change recording not enabled	<p>If you do not use change recording, you still use SAP HANA Web-based Development Workbench or SAP HANA studio for your SAP HANA development and you use one of the transport scenarios described above to transport your development entities. However, choosing this option means that you do not have any method to track changes, for example, if errors occur, you need to use a different means to find the source of the errors.</p>

Related Information

[Setting Up the Transport \[page 20\]](#)

[Setting Up and Using Native SAP HANA Transport \[page 26\]](#)

[Setting Up and Using CTS Transport \[page 37\]](#)

[SAP HANA Change Recording \[page 63\]](#)

[Setup of the Transport Landscape \[page 68\]](#)

[Assembling Add-On Products and Software Components \[page 86\]](#)

3.2 Setting Up and Using Native SAP HANA Transport

You use the *Native SAP HANA Transport* option to transport native SAP HANA objects.

Prerequisites

- You have enabled your system for *Native SAP HANA Transport*.
- You have planned your transport scenario. This includes deciding which systems are required for transports, and if you want to use change recording in the development system. For more information, see *Transport Scenarios in SAP HANA Application Lifecycle Management*. If you have enabled change recording, see also *Setup of the Transport Landscape*.

Context

If you transport native SAP HANA objects only and you do not have any ABAP transports or other non-SAP HANA transport activities, then you should use the native SAP HANA transport.

You perform native SAP HANA transports on the **target system**, pulling the content from the source system into the target system.

To use native SAP HANA transport, you must perform the following tasks.

Procedure

1. Register your SAP HANA source systems.

On the SAP HANA target system, you must make the source system for transports known to the target system. This includes entering the host and the XS engine port of the source system and maintaining the transport destination in the *SAP HANA XS Administration Tool*.

2. Create transport routes.

A transport route defines the connection details, content and mode for the transport between SAP HANA source and target systems.

3. Execute the transport on the specified transport routes.

Execute a transport operation that exports delivery units or a product from the source SAP HANA system (defined in an HTTP destination) and imports them into the local (target) SAP HANA system.

You can trigger a transport with the privileges granted by a role based on the `sap.hana.xs.lm.roles::ExecuteTransport` role template, you do not need to have the privileges of a role based on the *Lifecycle Management Administrator* role template.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Setting Up the Transport \[page 20\]](#)

[Register a System for a Transport Route \[page 27\]](#)

[Create a Transport Route \[page 29\]](#)

[Start the Transport \[page 32\]](#)

[Transport Scenarios in SAP HANA Application Lifecycle Management \[page 22\]](#)

[Setup of the Transport Landscape \[page 68\]](#)

3.2.1 Register a System for a Transport Route

In the context of a SAP HANA transport route, the system you register is an HTTP destination representing the source system where the object you want to transport is located, for example a delivery unit (DU).

Prerequisites

To register a system for a transport route, you must ensure the following prerequisites are met:

- You are logged on to the SAP HANA system that is the **target** of the transport route.
- A technical user must already exist on the source (HTTP destination) system you register in this step. The technical user for SAP HANA application lifecycle management transport must not be an SAP HANA restricted user and it requires privileges granted by a role based on the SAP HANA XS `sap.hana.xs.lm.roles::Transport` user role template.
- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role template.
- You have the privileges granted by roles based on the following SAP HANA XS role templates:
 - `sap.hana.xs.admin.roles::HTTPDestAdministrator`
 - `sap.hana.xs.admin.roles::RuntimeConfAdministrator`

Context

To create and register an HTTP destination as part of the setup of a transport route, you must have the privileges required to create an HTTP destination configuration and, in addition, maintain the logon credentials of an existing technical user on the destination system.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. Choose the **TRANSPORT** tab.
3. Choose the **System** tab.
4. Register a new system.

Choose **Register** to start the registration process and enter the system details in the **Register System** dialog.

- **Host**

The name of the source SAP HANA system, where the delivery units you want to transport are located. If you want to use a tenant database on a multiple-container system, make sure that you specify the host of the tenant database here. You do this by adding the system ID (SID) of the tenant database as an alias to the host name of the SAP HANA system. Use the following format:

```
<Host name of SAP HANA system>-<SID of tenant database>
```

For more information about where to get the SID of the tenant database, see *Configure HTTP Access to Multitenant Database Containers* in the *SAP HANA Administration Guide*. A link to the guide is in the *Related Information* section.

Example

- **Host:** `101234-DB1.mycompany.com` OR
- **Host:** `101234-DB1`
- **XS Engine HTTP(S) Port**
The port number of the XS Engine associated with the SAP HANA instance running on the host specified in **Host**.

Choose **Next** to continue registering the system.

The **Configure Destination** panel appears.

5. Maintain the HTTP destination.

Choose **Maintain Destination** to display details of the HTTP destination you want to maintain in the **SAP HANA XS Administration Tool**.

Note

You must have the SAP HANA XS **HTTPDestAdministrator** role to complete this step.

6. Maintain details of the required technical user.

The technical user is required for the execution of the transport on the destination system.

- a. In the **Authentication** panel of the **HTTP Destination** screen (in the **SAP HANA XS Administration Tool**), select the method used to authenticate the user on the destination system, or tenant database, for example, **Basic**.

If you want to use single sign-on, see *Maintaining Single Sign-On for SAP HANA XS Applications* in the *SAP HANA Administration Guide* for more information. The link to the guide is in the *Related Information* section.

- b. Enter the name of the technical user described in the *Prerequisites* section. No check is made at this point to ensure the validity of the user name (or the corresponding password) on the destination system.
 - c. Enter a password for the technical user.
7. Choose [Save](#) to make the changes to the HTTP destination configuration and close the [SAP HANA XS Administration Tool](#).
8. Choose [Finish](#) to create the new HTTP destination.

i Note

Before the changes are saved, a check is made to ensure a logon is possible on the destination system with the user name and password provided. If the check fails, then a message appears with details.

Results

You have registered an HTTP destination for communication with the source system, or with a tenant database on this system.

If you use a tenant database, the [SID](#) column displays the system in the following format: `<Name of tenant database>@<Name of SAP HANA system>`.

You can now create transport routes for the registered system.

You can modify registered systems by choosing [Edit](#) for a selected system.

You can delete registered systems by choosing [Remove](#) for a selected system. If you want to do this, you must make sure that all transport routes which use this system are removed beforehand.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

3.2.2 Create a Transport Route

A transport route defines the configuration details which specify the source and target systems for a transport operation.

Prerequisites

To create a transport route for SAP HANA objects, you must ensure the following prerequisites are met:

- You are logged on to the target SAP HANA system.
- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role template.
- You have registered the source system for the transport route that you want to configure on the target system.

Context

A transport route specifies the source and target systems for a transport operation as well as additional details about the objects to transport and the transport mode, for example, transport based on changelists or transport based on complete delivery units. You can use the transport route to transfer a delivery unit between a source system (defined in an HTTP destination on the target system) and a target system, which is the local SAP HANA system that you are logged onto as the application lifecycle administrator.

To create a transport route, perform the following steps:

Procedure

1. Open SAP HANA Application Lifecycle Management.

SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. Display the **TRANSPORT** tab.

Choose the **Transports** tab.

3. Create a new transport route.

Choose **Create** and use the **Create Transport Route** dialog to enter details of the new SAP HANA transport route. If change recording is active in the system, some options are different.

- **Name**

Enter a name for the transport route.

- **Source System**

Use the drop-down list to select the System ID (SID) of the SAP HANA source system on which the delivery unit to transport is located. All systems that you have registered before as source systems are available in the list.

After you have selected the source system the delivery units or the product instance changes to reflect the content of the selected source system. The transport mode may also change between **Complete Delivery Units** and **Selected Changelists** or **All Changelists**.

- **Content**

Choose the content that you want to transport. If you select the **Delivery Units** option, delivery units are transported. If you select the **Product Instance** option, product instances are transported. Depending on which option you select, the system displays a list of delivery units or product instances that exist on the source system.

You can select one or more (with the **CTRL** key) delivery units to include in the transport, but you can select only one product instance.

- If change recording is **not active** in the selected **source** system, the [Complete Delivery Units Mode](#) is preselected. You cannot make any changes. The system always transports complete delivery units of the selected delivery units or product instance.
 - If change recording is **active** in the selected source system, you have the following options for the [Mode](#):
 - [Selected Changelists](#)
This transports all objects that are part of released changelists of the selected delivery units or the product instance that were not yet transported to the target system. When you start the transport, a dialog box with a list of changelists appears where you can select the changelists that you want to transport.
 - [All Changelists](#)
This transports all changelists that are in status released for the selected delivery units or the product instance. This transport mode corresponds to a transport of the complete delivery unit or product instance.
- For more information on the transport modes available with change recording, see *Transport Modes in Change Recording*. The link to the topic is in the *Related Information* section.
- Enter a meaningful comment in the [Comment](#) field to enable others to differentiate between the transport routes. This is especially important if you work with a multitude of transport routes.

4. Save the details of the new transport route.

Choose [OK](#) to finish creating the new transport route.

Results

You have created a new transport route. You can now start the transport on this transport route.

You can modify transport routes by choosing [Edit](#) for a selected transport route.

You can delete transport routes by choosing [Remove](#) for a selected transport route.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[SAP HANA Delivery Units \[page 50\]](#)

[Register a System for a Transport Route \[page 27\]](#)

[Transport Modes in Change Recording \[page 70\]](#)

3.2.3 Start the Transport

A transport operation enables you to move a delivery unit (DU) or a product between a source system (defined in an HTTP destination) and a target system, which is the local SAP HANA system that you are logged onto as the application lifecycle administrator.

Prerequisites

To execute a transport using a defined SAP HANA transport route, you must ensure the following prerequisites are met:

- You can log on to the target system defined in the SAP HANA transport route.
- A technical user with valid logon credentials exists on the source system specified in the SAP HANA transport route.
- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::ExecuteTransport` role template.

Context

To transport a DU or a product between a source system and a target system, perform the following steps:

Procedure

1. Open the SAP HANA Application Lifecycle Management on the SAP HANA target system.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`
2. Choose the **TRANSPORT** tab.
3. Choose the **Transports** tab.
4. Select the transport route on which you want to execute the transport.
5. Choose **Start Transport**.

SAP HANA Application Lifecycle Management displays the details of the transport you want to start in the **Start the Transport** dialog.

Choose **OK** to start the transport.

i Note

If change recording is active in the selected source system and if transport mode **Selected Changelists** is set in the transport route, the **Transport of Changelists** dialog box appears. Select the changelists that you want to transport and choose **Next**.

If changelists exist in the same package that were released earlier than the ones that you want to attach to the transport request, the predecessor changelists are also included in the transport request. You

must always transport the predecessors with the selected changelists. For more information, see *Predecessor Changelists in SAP HANA Change Recording*. If you do not want to transport predecessor changelists, you must modify your selection.

To see the objects of a changelist, ensure that the *Show Objects* field is selected and select the *Change ID* in the list.

The list of objects appears in *Objects in the selected Changelist(s)* section of the screen.

6. Choose *Next*.
7. Verify the DU/product name and the changelists and then choose *Transport and Close*.

Results

If no errors occurred, the `Transport completed successfully` message appears at the bottom of your screen. Check the transport logs by selecting the status message in the *Last Transport* column.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)
[Create a Transport Route \[page 29\]](#)
[Transport Modes in Change Recording \[page 70\]](#)
[Check the Transport Logs \[page 47\]](#)
[Predecessor Changelists in SAP HANA Change Recording \[page 65\]](#)

3.2.4 Setting Up Native SAP HANA Transport for SAP HANA Cloud Platform

If you want to perform native SAP HANA transports using SAP HANA Application Lifecycle Management on SAP HANA systems in the SAP HANA Cloud Platform, you must set up a trust relationship so that the SAP HANA source system can be reached by the SAP HANA target system. This is required because systems in the SAP HANA Cloud Platform can only be reached using HTTPS communication.

Context

To set up the trust relationship you must exchange a certificate. This includes the following steps:

1. Export a certificate of the SAP HANA source system to a file.
2. Make the certificate that you exported from the source system known on the target system.
To do this, create a trust store on the SAP HANA target system and import the exported certificate file into the new trust store.

3. Make the configuration settings for native SAP HANA transport on the target system (register the SAP HANA source system and create transport routes).

Export a Certificate on the Source System

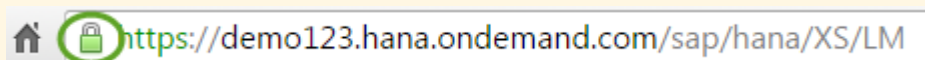
Procedure

1. Start SAP HANA Application Lifecycle Management on the source system on SAP HANA Cloud Platform.
Example: `https://demo123.hana.ondemand.com/sap/hana/XS/LM`
2. Use the Web browser's functions to export the certificate and save it to a file. The certificate must exist in the *DER encoded binary* format.

Example

The following is an example of how to export the certificate if you use Google Chrome.

1. Click on the Lock icon in the browser bar:



2. Choose the *Connection* tab.
3. Choose the *Certificate Information* link.
The certificate with which you are currently logged on is displayed.
4. In the *Certificate* dialog box, choose the *Details* tab.
5. Choose *Copy to File...* for the certificate.
The *Certificate Export Wizard* opens.
6. Select the *DER encoded binary* format and choose *Next*.
7. Browse for a location where you want to store the file, enter a file name and choose *Next*. If the export was successful, you get a success message.

Make the Certificate Known on the Target System

Procedure

1. Start the SAP HANA XS Administration Tool of the SAP HANA target system of the SAP HANA Cloud Platform.
Example: `https://demo456.hana.ondemand.com/sap/hana/XS/admin`
2. Start the *Trust Manager*.
3. To create a new trust store, choose *Add*. Type a name for the new trust store and choose *OK*.

The trust store appears in the *Trust Store* pane. You can now import the certificate that you exported on the source system.

4. In the new trust store, go to the [Certificate List](#) tab and choose [Import Certificate](#).

After successful import the source system certificate is known in the target system.

For more information on creating a trust store and importing certificates, see *Managing Trust Relationships* in the *SAP HANA Administration Guide*.

Make the Configuration Settings for Native SAP HANA Transport

Prerequisites

All prerequisites are fulfilled as described in *Setting Up and Using Native SAP HANA Transport* and *Register a System for a Transport Route*.

Procedure

1. Start SAP HANA Application Lifecycle Management of the SAP HANA target system of the SAP HANA Cloud Platform..

Example: `https://demo456.hana.ondemand.com/sap/hana/XS/LM`

2. Choose **TRANSPORT** > [System](#) > [Register](#).

3. Enter the [Host](#) of the SAP HANA Cloud Platform source system (Example: **demo123.hana.ondemand.com**), and enter **443** as [XS Engine HTTP\(S\) Port](#). This is the only port that allows HTTPS communication.

Then choose [Next](#).

4. Choose [Maintain Destination](#).

The *SAP HANA XS Administration Tool* opens.

5. Choose [Edit](#) to be able to make changes. Make the following settings:

- Proxy Details:

Field	Value
Proxy Type	HTTP
Proxy Host	If you use a proxy, enter your proxy host (Example: proxy).
Proxy Port	If you use a proxy, enter your proxy port (Example: 8080).
User	No entry
Password	No entry

- Authentication Details:

Field	Value
<i>SSL Enabled</i>	On
<i>Trust Store</i>	Select the trust store that you created beforehand (which contains the certificate) from the dropdown list.
<i>SSL Authentication Type</i>	Select <i>Client Certificate</i> .
<i>Authentication</i>	Select <i>Basic</i> .
<i>User</i>	Enter <Your user>
<i>Password</i>	Enter <Your password> .

6. Save your entries. Close the SAP HANA XS Administration Tool and choose *Finish* to finish registering the system.

Before the changes are saved, a check is made to ensure a logon is possible on the destination system with the user name and password provided.

Results

You have registered an HTTPS destination for communication with the source system. You can now create a transport route for the registered system.

For more information about how to configure the connectivity for SAP HANA XS in cloud environments, search for *Connectivity for SAP HANA XS (Enterprise Version)* in the *Connectivity Service* guide of the *SAP Cloud Platform* documentation at <https://help.sap.com/cp>.

Related Information

[Setting Up and Using Native SAP HANA Transport \[page 26\]](#)

[Register a System for a Transport Route \[page 27\]](#)

[Documentation on SAP HANA Cloud Platform](#)

3.3 Setting Up and Using CTS Transport

You use the [CTS Transport](#) option to transport SAP HANA objects in transport landscapes where CTS is already in place.

Prerequisites

- You have enabled [CTS Transport](#) in your SAP HANA source system.
- You have planned your transport scenario. This includes deciding which systems are required for transports, and if you want to use change recording in the development system. For more information, see *Transport Scenarios in SAP HANA Application Lifecycle Management*. If you have enabled change recording, see also *Setup of the Transport Landscape*.
- You have performed all configuration steps that are necessary on the AS ABAP to be able to perform SAP HANA transports. For more information, search for *Transporting Non-ABAP Objects in Change and Transport System* in the *SAP NetWeaver* documentation on SAP Help Portal, or see the guide *How To ... Configure SAP HANA for CTS*. The link to the guide is in the *Related Information* section.

Context

In the following cases we recommend that you use CTS transport:

- If you already use CTS for transports of ABAP or other non-ABAP objects to manage transports of SAP HANA objects using the CTS infrastructure.
- If you use a change control solution (Change Request Management or Quality Gate Management in SAP Solution Manager) to manage your transports.

You perform CTS transport activities on the **source system**, transporting the content from the source system to the target system.

To use CTS transport, you must perform the following tasks.

Procedure

1. Make the configuration settings for CTS transport in your SAP HANA source system.
On the SAP HANA source system, you configure an HTTP destination to the CTS communication system and the representation of the source system in the CTS communication system.
2. To transport your development artifacts using CTS, you have the following options:
 - a. Export delivery units with CTS
You can execute a CTS transport for complete delivery units that are assigned to CTS if change recording is not active in the SAP HANA system, or you can execute a transport of all released changelists of specific delivery units, if change recording is active.

b. Export changelists with CTS

You can execute a CTS transport for selected changelists of DUs that are assigned to CTS, if change recording is active in the SAP HANA system.

For more information on the transport of changelists, see *Transport Modes in Change Recording*.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Setting Up the Transport \[page 20\]](#)

[Configure SAP HANA Systems for CTS Transport \[page 38\]](#)

[Export Delivery Units for CTS Transport \[page 42\]](#)

[Export Changelists for CTS Transport \[page 43\]](#)

[How To ... Configure SAP HANA for CTS](#)

[Transport Modes in Change Recording \[page 70\]](#)

3.3.1 Configure SAP HANA Systems for CTS Transport

To use CTS transport, you need to configure both the SAP HANA system and the CTS communication system (AS ABAP). This chapter covers only the steps that you need to perform in the SAP HANA system.

Prerequisites

To configure the SAP HANA source system for CTS transport, you must ensure the following prerequisites are met:

- You are logged on to the SAP HANA source system.

Note

If you use multitenant database containers, make sure you are logged on to the correct source tenant database. For more information, see the guide *How-To Configure SAP HANA for CTS*. The link to the guide is in the *Related Information* section.

- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role template.
- You have the privileges granted by roles based on the following SAP HANA XS role templates:
 - `sap.hana.xs.admin.roles::HTTPDestAdministrator`
 - `sap.hana.xs.admin.roles::RuntimeConfAdministrator`
- You have enabled CTS transport in SAP HANA Application Lifecycle Management.

Context

Configuration is required on the SAP HANA source system only. If you use multitenant database containers, configuration is performed on the source tenant database. You have to configure the SID under which this system is known in CTS (CTS upload system ID) and you have to configure an HTTP destination to your CTS communication system to enable communication between the SAP HANA system and the CTS communication system.

The information about the target system for the transport is defined in the CTS communication system (AS ABAP).

Procedure

1. Open SAP HANA Application Lifecycle Management on the SAP HANA source system.
SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.
2. Choose the **SETTINGS** tab.
3. Configure the CTS system.
 - a. In the **Transport** section, choose **Configure CTS System**.
 - b. In the **CTS Upload System** dialog, under SID, you must enter the system ID of the SAP HANA source system in CTS. This is the representation of the source system in the CTS communication system. The SID of the SAP HANA system to which you are currently logged on is entered automatically. This is usually correct.
 - c. Optional: Enter a comment.
 - d. The number of the export file format is pre-selected. It corresponds to the export file format used in the current SAP HANA server. If you want to import the exported file in older SAP HANA server versions, select a lower file format from the list. For a mapping between the file format and the SAP HANA server version, see SAP Note 1984354. The link to the SAP Note can be found in the *Related Information* section.
4. Configure the HTTP destination to the CTS communication system.
 - a. Choose **Configure CTS Destination**.
The **HTTP Destination** maintenance in the **SAP HANA XS Administration Tool** opens. If it opens in display mode, switch to change mode if you need to change data.
 - b. Check that the data for the CTS communication system is displayed here.
 - c. In the **Path Prefix** field, enter the relative path to the appropriate Export Web Service as configured on the CTS communication system. For more information on the CTS Export Web Service, refer to the guide *How-To Configure SAP HANA for CTS*. The link to the guide can be found in the *Related Information* section.
 - d. Enter the data in the **Authentication** section as required by the security policy in your company.

➔ Tip

We recommend that you use **SAP Assertion Ticket**. For more information about how to set up your system to use assertion tickets, see *Authentication Assertion Tickets* in the *SAP NetWeaver Security Guide*.

If you want to use single sign-on, see *Maintaining Single Sign-On for SAP HANA XS Applications* in the *SAP HANA Administration Guide* for more information. The link to the guide is in the *Related Information* section.

- e. Choose [Save](#) to make the changes to the HTTP destination configuration and close the [SAP HANA XS Administration Tool](#).
5. Choose [Save](#) to conclude configuring the CTS system.

Note

Before the changes are saved, a check is made to ensure a logon is possible on the destination system with the authentication data provided. If the check fails, then a message appears in the dialog and you cannot save the data.

Results

You have made the relevant configuration settings in SAP HANA Application Lifecycle Management for CTS transport.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Setting Up the Transport \[page 20\]](#)

[How-To Configure SAP HANA for CTS guide](#) 

[SAP Note 1984354](#) 

3.3.2 Change CTS Configuration

If you need to make changes to the CTS configuration (CTS upload system or CTS communication system), you need to consider some important points.

Prerequisites

To change the configuration on the SAP HANA source system for CTS transport, you must ensure the following prerequisites are met:

- You are logged on to the SAP HANA source system.

Note

If you use multitenant database containers, make sure you are logged on to the correct source tenant database. For more information, see the guide *How-To Configure SAP HANA for CTS*. The link to the guide is in the *Related Information* section.

- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role template.
- You have the privileges granted by roles based on the following SAP HANA XS role templates:
 - `sap.hana.xs.admin.roles::HTTPDestAdministrator`
 - `sap.hana.xs.admin.roles::RuntimeConfAdministrator`
- You have configured CTS transport in SAP HANA Application Lifecycle Management.

Context

In general, we recommend that you do not change the CTS configuration. However, there are situations where it might be required. Only change the configuration if you are an expert user.


Making changes to the CTS configuration can cause inconsistencies in your system landscape, because changes to the configuration will not lead to a re-export of already exported changelists or delivery units. If you have also made changes to the transport landscape, you may have to manually re-export the complete delivery units to guarantee consistency, before performing new exports. To perform a re-export of DUs, choose ► [CTS EXPORT](#) ► [Delivery Units](#) ► [Attach to transport request](#) ► for the selected DUs.

If you only change the name of the upload system or the communication system, but the system and the transport routes remain unchanged, you can continue to perform the transport of selected changelists.

Procedure

1. To change the configuration of the CTS upload system, you can choose [Configure CTS System](#) and change details in the dialog, such as the [SID](#), or details of the HTTP destination.
2. If you want to change the CTS communication system, you must first delete the CTS configuration. When you have deleted the information, you can configure a new CTS communication system.
Concerning changes to the CTS communication system, read the recommendations described in SAP Note 1715802.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)
[Configure SAP HANA Systems for CTS Transport \[page 38\]](#)
[SAP Note 1715802](#) 

3.3.3 Export Delivery Units for CTS Transport

The export of delivery units (DUs) using CTS involves assigning them to CTS and exporting them.

Prerequisites

- To execute a transport using CTS, you must ensure that you have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::ExecuteTransport` role template.
- The SAP HANA system must be enabled and configured for CTS transport.

Context

To export DUs with CTS, perform the following steps:

Procedure

1. Open the SAP HANA Application Lifecycle Management.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`
2. Choose the *CTS EXPORT* tab.
3. Choose the *Delivery Units* tab if it is not open by default.
4. Choose *Assign Delivery Units*, if the DUs you want to transport are not assigned to CTS yet.
In the list of DUs displayed in the *Assign Delivery Units* dialog, select the *Assigned to CTS* checkbox to the right of the DUs you want to export, then choose *Save*. Only DUs that are assigned to CTS can be exported with CTS.
5. Select the DUs that you want to transport from the list and then choose *Attach to transport request*.
In the *Start CTS Export* dialog, you usually see a *Transport Request ID*, its description, and owner.
6. Verify the information and the transport request.

i Note

It depends on the configuration of your source system in CTS, whether a transport request is available for selection or not (TMS Parameter `WBO_GET_REQ_STRATEGY`).

To display more details, or to change properties, choose *Go to Transport Organizer UI*. Transport Organizer Web UI is used to manage transport requests. This includes creating transport requests, editing, and releasing them as well as changing details.

For more information, see the *How To... Configure SAP HANA for CTS* guide or *Managing Transports of Non-ABAP Objects* in the *CTS Plug-In* documentation. The links can be found in the *Related Information* section.

7. To start the export, choose [Export and Close](#) in the *Start CTS Export* dialog.

Results

The DUs are exported. If change recording is active in the system, the export is executed for all active objects that are part of released changelists of the DU(s). After the export, check the log. To do this, choose [Logs](#).

Depending on the configuration of your source system in CTS, you must release the transport request in the CTS communication system so that it can be imported in the target system, or it is automatically released (TMS Parameter `WBO_REL_REQ_STRATEGY`). To do this, choose [Open Transport Organizer](#) or use the link in the export log. For more information, see the *How To... Configure SAP HANA for CTS* guide.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Setting Up the Transport \[page 20\]](#)

[Configure SAP HANA Systems for CTS Transport \[page 38\]](#)

[How To... Configure SAP HANA for CTS !\[\]\(278d3fba7f1afbeedaef802f0c5d66fc_img.jpg\)](#)

[Check the Transport Logs \[page 47\]](#)

3.3.4 Export Changelists for CTS Transport

The export of changelists using CTS involves assigning the corresponding delivery units (DUs) to CTS, and exporting selected changelists of the DUs.

Prerequisites

- To execute an export of changelists for CTS transport, you must ensure that you have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::ExecuteTransport` role template.
- The SAP HANA system must be enabled and configured for CTS Transport.
- Change recording must be activated in the SAP HANA system and released changelists must exist.

Context

To export changelists, perform the following steps:

Procedure

1. Open SAP HANA Application Lifecycle Management.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`
2. Choose the *CTS EXPORT* tab.
3. Choose the *Released Changelists* tab.
4. If you have not yet assigned DUs to CTS, choose *Assign Delivery Units* and assign the DUs to CTS for which you want to export changelists.
5. Choose *Attach to Transport Request*.
6. Select the released changelists (assigned to CTS DUs) to export and choose *Next*.

The system executes a predecessor check. If changelists exist in the same package that were released earlier than the ones that you want to attach to the transport request the predecessor changelists are also included in the transport request. You must always transport the predecessors with the selected changelists. For more information, see *Predecessor Changelists in SAP HANA Change Recording*.

Note

If you do not want to export predecessor changelists, you must modify your selection.

A list of objects for each changelist appears in *Objects in the Changelist* section on the screen.

7. Choose *Next*.
8. Verify the information and the transport request..

Note

It depends on the configuration of your source system in CTS, whether a transport request is available for selection or not (TMS Parameter `WBO_GET_REQ_STRATEGY`).

To display more details, or to change properties, choose *Go to Transport Organizer UI*. Transport Organizer Web UI is used to manage transport requests. This includes creating transport requests, editing, and releasing them as well as changing details.

For more information, see the *How To... Configure SAP HANA for CTS* guide or *Managing Transports of Non-ABAP Objects* in the *CTS Plug-In* documentation. The links can be found in the *Related Information* section.

9. To start the export, choose *Export and Close* in the *Start CTS Export* dialog.

Results

The selected changelists are exported. After the export, check the export log. To do this, choose [Logs](#).

Depending on your CTS configuration, you must release the transport request in the CTS communication system so that it can be imported in the target system, or it is automatically released (TMS Parameter `WBO_REL_REQ_STRATEGY`). To do this, choose [Open Transport Organizer](#) or use the link in the export log. For more information, see the *How To... Configure SAP HANA for CTS* guide.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[How To... Configure SAP HANA for CTS](#) 

[Transport Modes in Change Recording \[page 70\]](#)

[Predecessor Changelists in SAP HANA Change Recording \[page 65\]](#)

3.4 Remarks on the Transport

If you transport the complete delivery unit (DU) or all changelists of a DU, either all active objects of a DU (if change recording is not enabled) or all active objects that are part of released changelists of the DU (if change recording is enabled) are transported.

During the transport process the following steps are executed:

1. On the source system: Export of the DU including its objects into a DU archive.
If change recording is not enabled, the DU archive contains all active objects of the DU. If change recording is enabled, the DU archive contains the last active version of each object of the DU that is part of a released changelist.
2. Transfer of the DU archive including its objects to the target system.
3. Import of the objects of the archive and activation of the objects.

This can lead to the following results:

- Native SAP HANA transport
If all objects of the archive can be successfully activated, the transport process finishes successfully (return code 0). If one transported object cannot be activated, then the import of all transported objects is rolled back. The transport process finishes with the return code 8. The DU archive is not in the system. If change recording is enabled in the target system, the imported changelist is also not in the system and no new changelist is created.
- CTS transport
If all objects of the archive can be activated successfully, then the transport finishes successfully (return code 0). If there are objects in the archive that cannot be activated, they are nevertheless imported in the target system, but they receive the status *broken*. If an object that references another object in the archive cannot be activated, the referenced object will also not be activated either and receives the status *broken*. All the objects of the archive which can be activated are imported and activated. The transport process

finishes with the return code 8. If change recording is enabled in the target system, the imported changelist is in the system and a new changelist is created.

Note

If objects outside of the transported archive reference objects in the transported archive, it is possible that the referencing objects cannot be activated after the import. These objects are also in status *broken* after the import. If you have broken objects after the import, you **must** repair them. SAP recommends that you perform the correction in the source system and then transport the repaired objects to the target system to ensure a consistent status in all systems.

Special Transport Situations

Packages that are unassigned from a DU can cause object deletions

If you transport the complete DU or all changelists of a DU, all active objects of packages that are assigned to the DU are exported into a DU archive (.tgz file). When the archive is imported in the target system, the objects that are part of the DU in the target system are overwritten by the content of the DU archive. If the imported DU contains fewer packages than the same DU on the target system, because some packages were unassigned from the DU (or re-assigned to another DU), the objects that are part of the missing packages will be deleted on the target system. The packages will be unassigned from the DU on the target system and are empty afterwards.

The following example illustrates this behavior for the transport of a complete DU or all changelists of a DU.

Example

DU DEMO_DU contains the following objects in the target system:

- Package demo.aaa with objects AAA.xsjs and AA.xsjslib
- Package demo.bbb with object BBB.xsjs
- Package demo.ccc with object CCC.xsjs

The transport of the DU archive of DEMO_DU contains the following objects:

- Package demo.aaa with objects AAA.xsjs
- Package demo.bbb with object BBB.xsjs

When the archive is imported in the target system, the following objects are deleted:

- Object AA.xsjslib of package demo.aaa
- Object CCC.xsjs of package demo.ccc

So the final state of DU DEMO_DU on the target system is the following:

- Package demo.aaa with object AAA.xsjs
- Package demo.bbb with object BBB.xsjs

Dependencies of objects between DUs can cause activation errors

For native SAP HANA transport the following is valid: If objects of one DU have dependencies on objects of another DU, then you must make sure that the dependent object is imported either at the same time or later

than the object on which it depends. Otherwise, the dependent object cannot be activated in the target system and the import will be rolled back. If you have circular dependencies, the dependent DUs must be imported simultaneously so that all objects can be activated successfully.

Example

DEMO_DU contains privileges (demo.aaa.xsprivileges). Application demo.bbb.Tester.hdbrole is part of DU INTERNAL_DEMO_DU and refers to the privileges of DEMO_DU. If the DU INTERNAL_DEMO_DU is transported first, activation would fail because the required .xsprivileges object is not yet in the target system.

To avoid this situation, assign both DUs to the same transport route and perform the import for both DUs at the same time. Alternatively, make sure that you import DEMO_DU before you import INTERNAL_DEMO_DU.

If you use CTS transport, SAP recommends that you assign DUs to the same transport request if objects in one DU have dependencies on objects in the other DU. Alternatively, you can import the transport request that contains the dependent object later than the transport request that contains the object on which it depends. However, if neither option is possible, the transport request that contains the dependent object will still be imported with CTS transport. If it is transported earlier than the object on which it depends, the job completes with return code 8, as described above. The dependent object will remain broken in the system. As soon as a transport request brings the object on which it depends, the broken object will be activated automatically.

Related Information

[Consequences of Package - DU Reassignments \[page 66\]](#)

3.5 Check the Transport Logs

The transport logs in SAP HANA application lifecycle management enable you to check the status of transport activities.

Context

The transport log allows you to monitor transport processes or analyze them after they are finished. SAP HANA application lifecycle management has an integrated log for all transport activities that take place on the SAP HANA system. This includes transport, export, and import activities.

The log is a list with multiple columns. An icon in the [Status](#) column gives a quick overview of the overall success of the transport activity or whether it is still running. Each transport activity has a unique transport action ID. If you click on the ID link, the detailed log view opens that contains more detailed information on the activity. The [RC](#) (return code) column contains the overall return code for the transport activity. The following return codes exist:

Return codes in the transport log

Return code	Description
0	Transport performed without errors.
4	Warnings were issued. All objects were transported successfully. There were however special actions for individual objects that may not have been intentional, for example, a warning is issued during the export if the transport contains an object deletion. Read the warnings.
8	Individual objects could not be transported successfully. You must analyze and correct the errors.
12	A critical error has occurred, probably not caused by the contents of the transport. You must inform your system administrator.

If there are transports with errors, analyze the cause of the error. The detailed view contains useful information. It also displays return codes for individual actions in the transport process. All actions with a return code of 8 or higher are highlighted in red.

You can perform the following tasks:

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. To access the transport logs, choose **TRANSPORT** > **Logs**.
3. You can filter the log display according to various criteria, such as the return code, user, source and target system, time, transport operation, or delivery unit name. By default, the filter criteria are hidden. To display them, choose [Show Filters](#).
4. You can display the detailed log view by clicking on the transport action ID.

You will get detailed information about the individual steps in the transport activity. You can see whether the individual steps were successful or not from the color of the entry, the return code, and the error code.

For example, for CTS transport, the number of the transport request is provided to which the DUs or changelists were assigned. Clicking on this link enables you to open Transport Organizer Web UI where you can check the content of the transport request.

5. If transport processes run for more than two hours, you can end them. To do this, choose [Stop Long Running Processes](#).

4 Maintaining Delivery Units

A delivery unit (DU) is a collection of packages that are to be transported together. You assign all the packages belonging to your application to the same DU to ensure that they are transported consistently together within your system landscape. Each DU has a unique identity.

Prerequisites

To maintain delivery units with the SAP HANA Application Lifecycle Management, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have been assigned the privileges granted by a role based on the SAP HANA `sap.hana.xs.lm.roles::Administrator` user role template.
- A vendor ID (repository namespace) is already defined.

Context

The identity of a delivery unit consists of two parts: a vendor name and a delivery-unit name. The combined ID ensures that delivery units from different vendors are easy to distinguish and follows a pattern that SAP uses for all kinds of software components.

To create and manage delivery units you first need to maintain the identity of the vendor, with whom the delivery units are associated, and in whose namespace the packages that make up the delivery unit are stored. As part of the vendor ID maintenance process, you must perform the following tasks:

Procedure

1. Understand delivery units.
You must be familiar with the conventions that exist for delivery-unit names and understand the phases of the delivery-unit lifecycle.
2. Maintain details of the vendor ID associated with a delivery unit.
Delivery units are located in the namespace associated with the vendor who creates them and who manages the delivery-unit's lifecycle.
3. Create a delivery unit.
Create a transportable “container” to hold the repository packages in application.
4. Assign packages to a delivery unit.
Add to a delivery unit the repository packages that make up your application.

5. Export a delivery unit.

You can export the contents of a delivery unit from the SAP HANA Repository to a compressed Zip archive, which you can download to a client file system.

6. Import a delivery unit.

You can import the contents of a delivery unit into the SAP HANA Repository, for example, from a compressed Zip archive, which you upload from a client file system.

Related Information

[Maintain the Delivery-Unit Vendor ID \[page 18\]](#)

[Create a Delivery Unit \[page 52\]](#)

[Export a Delivery Unit \[page 55\]](#)

[Import a Delivery Unit \[page 57\]](#)

4.1 SAP HANA Delivery Units

In SAP HANA, the delivery unit (DU) is the vehicle that SAP HANA application lifecycle management uses to ship one or more software components from SAP (or a partner) to a customer .

Delivery Units correspond to an "application" with versions, for which support packages and patches are delivered. Delivery units are not an equivalent to development classes; they are similar to Software Component Versions in the Product Availability Matrix (PAM). You can also use the delivery unit to transport repository content between SAP HANA systems, for example, between development systems or between development and productive systems.

Note

A governance plan exists for the naming of delivery units and the package structure.

A delivery unit is identified by the following key properties:

- **Vendor**
Used primarily to define the identity of the company producing the software component to deliver, for example, "sap". However, vendor can also be used to specify any customer implementing SAP HANA. To create a delivery unit, it is a prerequisite to maintain a vendor in your system.
- **Name**
Used to identify the software component to be delivered

Although a vendor might provide multiple versions of a delivery unit in parallel, only one version of a delivery unit can be installed in a HANA system at any one time. In addition, duplicate names are not allowed for delivery units.

SAP HANA treats delivery units and their versions as software components and software-component versions. Software components and software-component versions are usually shipped as part of a product (product version).

Note

If you try to create a delivery unit using a name that already exists in your system, you will receive a validation error. A validation error also occurs if the check for a valid **vendor ID** does not find a suitable entry.

4.2 SAP HANA Delivery Unit Naming Conventions

The delivery unit (DU) is the vehicle that SAP HANA application lifecycle management uses to ship software components from SAP (or a partner) to a customer. The DU is also the container you use to transport application content in your system landscape. In SAP HANA, the name of a DU must adhere to conventions and guidelines.

If you create a delivery unit, the name of the new delivery unit must adhere to the following conventions

- A delivery-unit name must contain only capital letters (A-Z), digits (0-9), and underscores (_).
- The name must start with a letter.
- The maximum length of a delivery-unit name must not exceed 30 characters

Note

The naming conventions for packages in a delivery unit differ from the naming conventions that apply to the delivery unit itself. For example, the maximum length of a package name is not restricted to 30 characters; however, it must be less than 190 characters (including the namespace hierarchy).

4.3 The SAP HANA Delivery-Unit Lifecycle

In SAP HANA, application lifecycle management includes all the activities you need to plan and perform to ensure that the software components you develop for SAP HANA are produced and shipped in a regulated way that meets the requirements laid out for the SAP HANA platform.

For example, you need to plan, manage, and maintain the application versions you want to ship, and you need to ensure that a stable plan exists to provide support for existing and all future versions of your application.

SAP HANA application lifecycle management uses the delivery unit (DU) as a vehicle to ship one or more software components from SAP (or a partner) to a customer. Delivery units correspond to an "application" with versions, for which support packages and patches are delivered. You can also use the delivery unit to transport content between SAP HANA systems, for example, between development systems or between development and production systems.

Note

Delivery units are not to be confused with development classes; delivery units are similar to software component versions in the Product Availability Matrix (PAM).

The Application-Development Lifecycle

Application development on SAP HANA requires a server-centric lifecycle for design-time objects, which are the development artifacts that you store in the SAP HANA repository. As an application developer, you check out design-time content from the repository and edit a copy of the checked-out artifact in the local file system on your personal computer (PC). The following steps provide a brief, high-level overview of the development lifecycle for design-time content:

1. Check out design-time content.
Check out the package containing the design-time artifacts that you want to work on (if the package already exists).
2. Edit the design-time content.
Edit the copies of the design-time artifacts, which are stored in your SAP HANA repository "workspace" on your local file system; the local copies of the design-time artifacts are created during the checkout process.
3. Commit changes to design-time content.
Committing the changes you have made to the design-time artifacts creates new versions of the artifacts in the SAP HANA repository. Note that identical (unchanged) versions of a file are not committed.
4. Activate changes to design-time content.
Activating the changes you have made to the design-time artifacts makes these changes available to applications and to other users, creating runtime objects where necessary.

4.4 Create a Delivery Unit

A delivery unit (DU) is a group of transportable packages that contain objects used for content delivery. You can use the SAP HANA Application Lifecycle Management to create a DU for your application content or your software component.

Prerequisites

To create a delivery unit with the SAP HANA Application Lifecycle Management, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have the privileges granted by a role based on the SAP HANA `sap.hana.xs.lm.roles::Administrator` user role template.
- The vendor ID is defined for the DU; the vendor ID defines the repository namespace in which the new DU resides.

Context

You use a DU to transport the design-time objects that are stored in the SAP HANA repository between two systems, for example, from a development system to a consolidation system. To create a new delivery unit using the SAP HANA application lifecycle management, perform the following steps.

Procedure

1. Open SAP HANA Application Lifecycle Management.

SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. Choose the *PRODUCTS* tab.
3. Choose the *Delivery Units* tab.
4. Choose *Create*.

The *New Delivery Unit* dialog box appears.

5. Enter details for the new DU.

When entering details, note the following points:

- *Name*
The field is mandatory and you must follow strict naming conventions, for example, use capital letters.
- *Vendor*
This field is mandatory. However, you cannot enter a vendor here; the box is populated by the value you enter when defining the vendor in the *SETTINGS* tab.
- *Version*
Version numbers must take the form “#.#.#”, for example, **1.0.5**, where:
 - **1** = the DU version number
 - **0** = the support package version (if required)
 - **5** = the patch version (if required)

i Note

The numbers you enter here refer to the application component that you are developing; the numbers do not refer to the patch or service-pack level deployed on the SAP HANA server.

6. Choose *Create*.
The new delivery unit is added to the SAP HANA repository in the namespace specified by the vendor ID and the application path.
7. Check the status bar at the bottom of the browser window for error messages. Choose the message link to display the message text.

Results

You have created a delivery unit.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[SAP HANA Change Recording \[page 63\]](#)

4.5 Assign Packages to a Delivery Unit

By default, a new delivery unit (DU) is empty; you must assign packages to it manually.

Prerequisites

To assign packages to a DU with the SAP HANA Application Lifecycle Management, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have the privileges granted by a role based on the SAP HANA `sap.hana.xs.lm.roles::Administrator` user role template.

Context

A DU contains one or more packages. You must assign the packages to the DU manually. You can also remove (unassign) packages from a DU and edit the details of a package. A package can only be assigned to one DU. To assign packages to a DU, perform the following steps:

Procedure

1. Open the SAP HANA Application Lifecycle Management.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`
2. Choose the *PRODUCTS* tab.
3. Choose the *Delivery Units* tab.
4. Select the DU to which you want to assign some packages.
The *Assigned Packages* panel displays the current contents of the selected DU.

➔ Tip

To remove (unassign) a package from a DU, select the package and choose *Unassign*.

5. Assign new packages to the DU.

Choose [Assign](#) and select the name of the package you want to assign to the DU.

i Note

Ensure that the [Select sub-packages](#) field is selected.

6. Choose [Check for Unassigned](#) to ensure that you have selected all packages and sub-packages that you want to assign to the DU.
If you have missed a sub-package, select it from this dialog box and choose [Assign](#).
7. Check the status bar at the bottom of the browser window for error messages. Choose the message link to display the message text.

Related Information

[Create a Delivery Unit \[page 52\]](#)

4.6 Export a Delivery Unit

You can export a delivery unit (DU), for example, to a file, for your application content or your software components using the SAP HANA Application Lifecycle Management.

Prerequisites

To export a delivery unit with the SAP HANA Application Lifecycle Management, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system
- You have the privileges granted by roles based on one of the following SAP HANA user role templates:
 - `sap.hana.xs.lm.roles::Administrator`
 - `sap.hana.xs.lm.roles::ExecuteTransport`

Context

A DU is a group of transportable objects used for content delivery. You can use a DU to transport the design-time objects that are stored in the SAP HANA repository between two systems, for example, from a development system to a consolidation system.

Note

If a system is configured to work with change recording, all activated objects must be approved before their release. Only released objects are exported from that system.

If a system is not configured to work with change recording, all active objects in a delivery unit are exported.

To export a DU (for example, from the SAP HANA repository to a file) using the SAP HANA Application Lifecycle Management, perform the following steps.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. Choose the *PRODUCTS* tab.
3. Choose the *Delivery Units* tab.
4. Select the DU you want to export.

In the *Delivery Units* list, locate and select the DU you want to export to a file.

Tip

You can check the details and contents of the selected DU in the *Details* and *Assigned Packages* panels respectively.

5. Export the selected DU.

To start the export, choose *Export*.

The *Export Delivery Unit to File* screen appears where you can select the file format for export. By default the newest file format is selected. If you want to import the DU into older SAP HANA servers, you can select an older file format. For more information on export file formats, you can choose the *?* icon on the screen and see SAP Note 1984354. The link to the SAP Note can be found in the *Related Information* section.

If you choose *Export*, a dialog appears that enables you to specify the location where you want to save the exported DU, for example, on a local file system.

Note

Depending on the browser settings, the import might start automatically or the file location might not be requested. For example, you have created a default location for all download operations.

Related Information

[SAP Note 1984354](#)

4.7 Import a Delivery Unit

You can import a delivery unit (DU), for example, from a file, for your application content or your software components using the SAP HANA Application Lifecycle Management.

Prerequisites

To import a delivery unit with the SAP HANA Application Lifecycle Management, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system
- You have the privileges granted by a role based on the SAP HANA `sap.hana.xs.lm.roles::Administrator` user role template.
- The package name of the DU does **not** exist in the system.

i Note

Package names are case-insensitive. If you have a package name with only upper-case and lower-case differences, the import fails because the system sees this as a duplicate. To import the DU, either delete the package from the system and then import the DU or rename the new package.

Context

A DU is a group of transportable objects used for content delivery. You can use a DU to transport the design-time objects that are stored in the SAP HANA repository between two systems, for example, from a development system to a consolidation system.

To import a delivery unit (for example, from a file to the SAP HANA repository) using the SAP HANA Application Lifecycle Management, perform the following steps.

Procedure

1. Open the SAP HANA Application Lifecycle Management.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`
2. Choose the *PRODUCTS* tab.
3. Choose the *Delivery Units* tab.

4. Choose *Import*

5. Select the DU you want to import.

Choose *Browse* to display a file explorer, which you can use to locate the DU you want to import, and choose *Open*.

➔ **Tip**

Exported DUs have the file extension `.tgz`, for example, `MyDU.tgz`.

The *Confirm Import of Delivery Unit* screen appears containing the list of objects included in that DU.

6. Confirm that this is the DU that you want to import.

Choose *Import* to import the selected delivery unit.

i Note

The import operation overwrites any identical objects in the target system with the content of the imported DU.

Related Information

[SAP HANA Change Recording \[page 63\]](#)

5 Maintaining Products

A product contains one or more delivery units. A delivery unit (DU) is a collection of packages that logically belong together. You assign delivery units to a product to ensure that they are transported consistently together within your system landscape.

Prerequisites

To maintain products with the SAP HANA Application Lifecycle Management, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have the privileges granted by a role based on the SAP HANA `sap.hana.xs.lm.roles::Administrator` user role template.
- A vendor ID is already defined.
- You have created at least one DU.

Context

To create and manage products, you first need to maintain the DUs which you assign to the product. A DU requires a vendor ID, the name of the vendor with whom the DUs are associated and in whose namespace in the SAP HANA repository the packages that make up the DU are stored. As part of the product maintenance process, you must perform the following tasks:

Procedure

1. Create a product.
2. Assign delivery units to a product.

Related Information

[Create a Product \[page 60\]](#)

[Assign a Delivery Unit to a Product \[page 61\]](#)

[Maintaining Delivery Units \[page 49\]](#)

[Maintain the Delivery-Unit Vendor ID \[page 18\]](#)

5.1 Create a Product

Use the SAP HANA Application Lifecycle Management to create a product and its components.

Prerequisites

To perform this task, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have the privileges granted by a role based on the SAP HANA XS `sap.hana.xs.lm.roles::Administrator` user role templates.
- The vendor ID is already defined for the delivery units you assign to the product.

Context

A product contains one or more delivery units, packages, and can contain applications associated with the packages. To use the SAP HANA Application Lifecycle Management to create a new product, perform the following steps:

Procedure

1. Open the SAP HANA Application Lifecycle Management.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`
2. Choose the *PRODUCTS* tab.
3. Choose the *Products* tab.
4. Choose *Create*.
The *New Product* dialog box appears.
5. Define the details of the new product.
 - a. Specify a *name* for the new product.
The *Vendor* box is populated with the value defined in the *SETTINGS* tab; you cannot enter or change the value here.
 - b. Optional: Enter a *Version*, and a *Description* for the product.
 - c. Optional: Create a default instance (with the instance ID *1*) for the product by selecting the corresponding check box and entering a name for the instance.
Each product version requires at least one product instance. The product instance indicates the entity that is to be installed. If you do not create the product instance now, you must do it later when you assign delivery units to the product. You can then enter an instance ID from **1** to **999**, and a name for the instance.

6. Create the new product.

Choose [Create](#) to add the new product to the list of products displayed in the [Products](#) tab.

The new product is empty at this stage; you must assign one or more delivery units to it.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Assign a Delivery Unit to a Product \[page 61\]](#)

5.2 Assign a Delivery Unit to a Product

A product can contain one or more product instances which can contain one or more delivery units. You must assign the delivery units (DU) manually to the product instances of the product.

Prerequisites

To assign DUs to a product, you must ensure the following prerequisites are met:

- You have access to an SAP HANA system.
- You have the privileges granted by a role based on the SAP HANA XS `sap.hana.xs.lm.roles::Administrator` user role template.
- The vendor ID is already defined for the DUs you assign to the product.

Context

To use the SAP HANA Application Lifecycle Management to assign an existing delivery unit to a product , perform the following steps.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. Choose the [PRODUCTS](#) tab.
3. Choose the [Products](#) tab.

-
4. Select the product to which you want to assign a DU or DUs.
In the list of products displayed on the left-hand side of the *Products* tab, select the product to which you want to assign a DU.
 5. If no product instance exists for the product, first create a product instance.
 - a. In the *Instances and Assigned Delivery Units* panel, choose ► *Instance* ► *New* ⌵.
 - b. Enter a number between **0** and **999** as *ID*.
 - c. Optional: Enter a description.
 6. If a product instance exists for the product, assign a DU to the selected product instance.
 - a. Choose *Assign Delivery Unit* to display a list of available DUs.
 - b. In the *Assign Delivery Units* dialog locate the DUs you want to assign to the selected product instance and choose *Assign*.
The assigned DUs are immediately removed from the *Assign Delivery Units* list and added to the *Assigned Delivery Units* list for the selected product instance.
 - c. Close the dialog after having assigned all relevant DUs.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Create a Product \[page 60\]](#)

[Maintaining Delivery Units \[page 49\]](#)

6 SAP HANA Change Recording

Change recording in SAP HANA is the infrastructure to keep track of changes during SAP HANA development.

It provides the following functions:

- **Automatic Recording and Grouping of Object Changes**

When change recording is enabled, you are prompted to assign your changes to a changelist when you activate a repository object in your development environment. You can group the objects that you want to transport together in one changelist.

We recommend that you assign objects of only one DU to the same changelist. If you use native SAP HANA transport, you should configure the transport route accordingly for the same DU, and if you use CTS transport, you should assign this DU to CTS and trigger the transport of changelists for this DU. This way, you make sure that all objects that belong to packages of this DU and that are part of a changelist are transported together.

- **Decoupling of Activation and Transport**

When change recording is not enabled, a transport of a delivery unit contains all active objects that are contained in the packages of the delivery unit. You can only transport the entire delivery unit, no matter whether the objects are ready for transport or not. Change recording allows you to make changes to individual objects of a delivery unit and transport only these in changelists whenever they are ready for transport. Only released changelists can be transported. The objects are transported in the state in which they were at the point in time when the changelist was released.

- **Different Transport Modes**

The changelist-based transport can be set up in two modes: Either all changelists that were ever released in one or more delivery units or a product, or selected changelists in one or more delivery units or a product can be transported.

Note

Transport of products is only supported using native SAP HANA transport.

If you use native SAP HANA transport, you configure the transport mode when you register the system for a transport route. If you use CTS transport, you select whether you want to attach [Released changelists](#) or [Delivery Units](#) to a transport request when you perform an export for CTS in SAP HANA Application Lifecycle Management.

- **Predecessor Calculation of Changes**

If a changelist contains objects from the same package that depend on objects in other changelists that are released but not yet transported, the system detects these and includes them in the transport as well.

- **Team Development**

Multiple team members can work on the same development objects (and use the same changelist). The changelist can only be released when all team members have indicated that the objects are ready for transport by approving their contribution to the changelist. The objects that are part of the changelist are locked for developers who do not contribute to the changelist.

Integration

The functions of change recording are integrated in the XS user interface SAP HANA Application Lifecycle Management. In your development environment (SAP HANA studio, or SAP HANA Web-based Development Workbench), you can start this UI using the context menus. For example, in the *Developer Perspective* of SAP HANA studio in the *Repositories* view, you choose *Change Management*, and in the *Project Explorer* view, you choose ► *Team* ► *HALM Change Management* . In the *Editor* of SAP HANA Web-based Development Workbench, choose ► *Navigation Links* ► *Lifecycle Management* .

There is an eclipse-based change view available as part of SAP HANA studio. However, to benefit from all functions, we recommend that you use the XS UI using the context menu.

Related Information

[Transport Modes in Change Recording \[page 70\]](#)

[Setup of the Transport Landscape \[page 68\]](#)

[Enable SAP HANA Change Recording \[page 77\]](#)

[Predecessor Changelists in SAP HANA Change Recording \[page 65\]](#)

6.1 Technical Details of Initial Change Recording Setup

When you enable change recording in SAP HANA Application Lifecycle Management, all active objects of the system are included in a base changelist.

You should enable change recording in development systems. These are the source systems for transports. To guarantee a consistent system state, SAP recommends that you enable change recording **before** you start your development activities.

Change recording is intended to be enabled one time when the system is set up. It is not intended to be enabled and disabled multiple times for a system. For compatibility reasons, change recording is disabled initially when a new system is installed. If you want to use it, enable it in SAP HANA Application Lifecycle Management and continue using it.

Note

You can also start using change recording at a later point in time during your development process, for example, after performing an upgrade.

When you select the checkbox to enable change recording, all active objects of the system that belong to packages assigned to a delivery unit (DU) are added to an initial changelist. This process is called creating a **base changelist**. Depending on the point in time when change recording is enabled, creating the base changelist can take some time.

This base changelist is not visible in the system. However, when the first changelist is transported for a DU that contains real changes, the base changelist with all active objects of packages that are assigned to the DU is

also transported to the target system. This ensures that the target system has exactly the same state as the source system.

Example

DU `ABC` contains one hundred objects at the time when change recording is enabled. All one hundred objects are included in the base changelist and will be released with it. After you enable change recording, you modify one object of DU `ABC` and you assign it to a changelist. Then you release the changelist and transport it. Even though the changelist that you release and transport only contains one modified object, all the one hundred objects of the released base changelist that are part of DU `ABC` are also included in this first transport.

If you use CTS transport, you can see the DU archive of the base changelist that contains the complete DU as an additional `.tgz` file next to the `.tgz` file created for the changelist in Transport Organizer Web UI. If you use native SAP HANA transport, the base changelist is not visible in the transport.

Since enabling change recording means that all active objects of the system that belong to packages assigned to a DU are initially transported to the target system, you should make sure beforehand that you actually want to have all objects that are part of the DU in the source system also available in the target system. For example, some test objects or local objects may exist that you do not require in the target system. Therefore, check in your development environment for each delivery unit whether there are packages or objects that you want to remove from the delivery unit before enabling change recording.

Note

The point in time is important when the transport (native SAP HANA transport) or export (CTS transport) of the first changelist is performed. If change recording was enabled when the DU was empty, but the transport (native SAP HANA transport) or export (CTS transport) is executed when packages are assigned to the DU, all the objects that belong to packages assigned to the DU are transported in the base changelist along with the first changed object.

Related Information

[Enable SAP HANA Change Recording \[page 77\]](#)

6.2 Predecessor Changelists in SAP HANA Change Recording

For SAP HANA application lifecycle management, predecessor changelists are those that were released earlier than the selected changelists, which were not transported yet and that contain objects from the same packages.

If you want to transport individual changelists, the system may calculate predecessor changelists that must be included in the transport. If all of the following prerequisites are met, a changelist is identified as a predecessor changelist:

- It was released earlier than the selected changelist.
- It was not transported to the target system yet.
- It contains objects from the same SAP HANA repository packages as objects in the selected changelist.

Example

The following released changelists are in your source system:

Example of predecessor changelists

Changelist	Release Time
Changelist 1 (demo .aaa, demo .bbb)	Released on 01.01.2015
Changelist 2 (demo .ccc)	Released on 02.01.2015
Changelist 3 (demo .bbb)	Released on 03.01.2015

If you select changelist 3 for transport, SAP HANA application lifecycle management detects changelist 1 as predecessor (because both changelists contain objects from the package demo .bbb) and includes it in the transport. You must transport it along with changelist 3.

If the changelists are imported successfully (without activation errors), SAP HANA application lifecycle management stores the information about which changelist was transported as the last one for each package. Changelists that were transported once are no longer offered for transport.

6.3 Consequences of Package - DU Reassignments

Reassigning packages from one delivery unit to another can cause inconsistencies.

Caution

SAP recommends that you do not reassign packages from one delivery unit to another, because inconsistencies can occur.

At some point during development, you might want to move packages that were already transported with a specific delivery unit (DU) to another one. If you need to do this, you must bear in mind that you need to keep the follow-on systems consistent. Inconsistencies that can occur in the process differ, depending on the transport scenario you are using.

Recommendation

If you can avoid it, do not reassign packages from one DU to another.

Transport of selected changelists

The transport of selected changelists is performed by selecting the following setting in the respective transport track:

- For native SAP HANA: transport mode [Selected changelists](#)
- For CTS transport: ► [CTS EXPORT](#) ► [Released Changelists](#) ►

If change recording is enabled in the source system and the transport is based on selected changelists, it is possible that after a DU reassignment, changelists that were transported earlier are retransported. For example, this can occur if some objects in a changelist were not transported, because these objects in the changelist belong to packages from a different DU. If you reassign the package to the DU later, the changelist will be offered for transport. When transported, the DU includes not only the objects from the reassigned package, but also all other DU objects in the changelist that were transported before. Further, other changelists that were transported previously might need to be transported again because they contain objects from the same DU that are newer than objects of the changelist that was retransported initially.

Example

DU A has the packages `demo.aaa` and `demo.bbb` assigned to it. DU C has the package `demo.ccc` assigned.

`demo.aaa` contains object `AAA.xsjs`.

`demo.bbb` contains object `BBB.xsjs`.

`demo.ccc` contains object `CCC.xsjs`.

Changelist 1 contains all three objects `AAA.xsjs`, `BBB.xsjs`, and `CCC.xsjs`.

If changelist 1 is transported for DU A, objects `AAA.xsjs` and `BBB.xsjs` are transported, because transports are always based on DUs and only objects are transported that are assigned to packages of the selected DU. Object `CCC.xsjs` is therefore not transported. If you reassign package `ccc` to DU A and you transport any changelist that contains objects from the package `demo.ccc` (DU A), changelist 1 will be transported again for DU A (and potentially many other previously transported changelists that contain objects from packages in the DU).

➔ Recommendation

Only reassign packages if the target system has exactly the same state as the source system.

Avoid including objects from more than one DU in one changelist.

Do not make any modifications in the target system. Always transport your changes from the development system to the target system. This also applies to corrections.

Transport of complete DUs

If change recording is enabled, the transport of complete DUs is performed by selecting the following setting in the respective transport track:

- For native SAP HANA: transport mode *All changelists*
- For CTS transport: ► *CTS EXPORT* ► *Delivery Units* ►

If change recording is enabled in the source system and the transport is based on complete DUs, it is possible that 'unassigned' objects or packages are deleted in the target system. The following example illustrates this.

Example

DU A has the packages aaa and bbb assigned to it. After you transported DU A to a target system, DU A has objects from both packages aaa and bbb in the target system.

If you reassign package bbb to a DU B in the source system and you transport the complete DU A to the target system, the package bbb and all its objects will be deleted in the target system.

Only after DU B is transported to the target system, will package bbb and its objects appear in the target system.

For more information on this subject, see the *Special Transport Situations* in *Remarks on the Transport*.

Related Information

[Remarks on the Transport \[page 45\]](#)

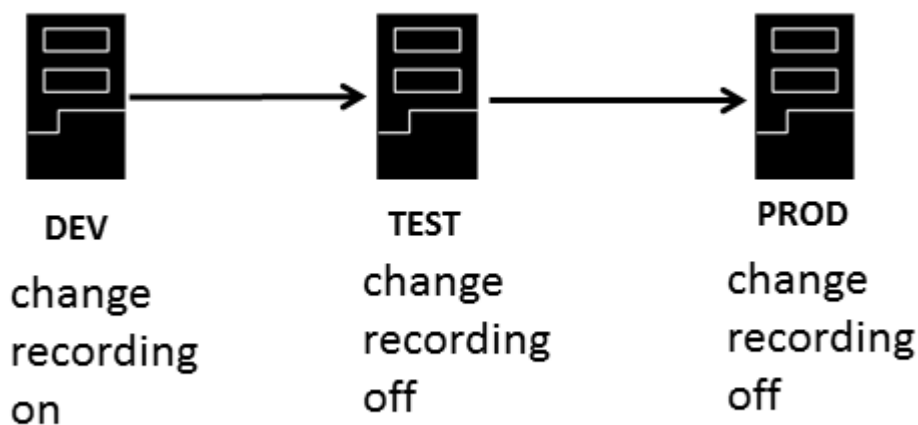
6.4 Setup of the Transport Landscape

Before you start working with change recording you must decide how you want to set up the systems in your transport landscape.

As a general rule, you enable change recording only in the development system and **not** in the follow-on systems (test and production systems).

➔ Recommendation

SAP recommends that you enable change recording in the development system only.



If change recording is enabled in the development system, all changes that you make in the development system are recorded in changelists.

After the changelists are transported to the test system, you can perform your tests there. If you need to correct an error, you make the corrections in the development system and subsequently transport the corrections in changelists to the test system.

Caution

Always change objects in the development system. If you do this, you can guarantee that all the follow-on systems in the transport track are consistent.

There are different scenarios available to set up your transport landscape when you use change recording. The decision regarding when to use what depends on whether you use CTS transport or native SAP HANA transport and whether you want to transport selected changelists or complete delivery units (all released changelists of delivery units (DUs)) from the test to the production systems.

Transport Scenarios for the Transport to Production Systems

When the tests are finished and all errors are corrected, you have the following options for the transport to the production systems:

- You can **continue to transport changelists** to the production systems.
This is required if you use CTS transport. For more information, see the *CTS and Changelists* topic. The link is included in the *Related Information* section.
If you use native SAP HANA transport, SAP recommends that you do **not** use this option. If you nevertheless want to transport changelists to the production systems, see the *Native SAP HANA and Selected Changelists* topic. The link is included in the *Related Information* section.
- You can **transport the complete DU** from the test system to the production systems.
If you use native SAP HANA transport, we recommend that you use this option. This guarantees maximum consistency in the production systems for this transport scenario. For more information, see *Native SAP HANA and Complete DU*.
If you use CTS transport, SAP recommends that you do **not** transport the complete DU from the test system to the production system, because this would involve a new export on the test system. In this case,

you cannot use the SAP tools for change control (Change Request Management and Quality Gate Management of SAP Solution Manager).

Transport Options

Transport Option	CTS Transport	Native SAP HANA Transport
Transport of Selected Changelists	<i>CTS and Changelists</i>	<i>Native SAP HANA and Selected Changelists</i>
Transport of Complete DU (All Changelists)	Not recommended	<i>Native SAP HANA and Complete DU</i>

i Note

Instead of transporting the software to the production systems, you can **use assembly and install** as mentioned in *Transport Scenarios in SAP HANA Application Lifecycle Management*.

Related Information

[Transport Modes in Change Recording \[page 70\]](#)

[Transport Scenarios in SAP HANA Application Lifecycle Management \[page 22\]](#)

[CTS and Changelists \[page 74\]](#)

[Native SAP HANA and Selected Changelists \[page 75\]](#)

[Native SAP HANA and Complete DU \[page 76\]](#)

6.4.1 Transport Modes in Change Recording

When change recording is enabled, you must choose a mode for the transport: You can either transport selected changelists in a delivery unit (DU), or all changelists in the DU.

Depending on whether you use native SAP HANA transport or CTS transport, you set the transport mode at different points in time of the transport process.

- **Native SAP HANA transport**

If you use native SAP HANA transport, you set the transport mode when you create the transport route in SAP HANA Application Lifecycle Management. You can choose to transport *Selected Changelists* or *All Changelists* for a DU or a product version.

i Note

Remember that native SAP HANA transports are “pull” transports. The transport configuration (connection to source system and creation of transport routes) is performed on the target system and the objects are pulled from the source system to the target system.

For more information, see *Create a Transport Route*.

- **CTS transport**

If you use CTS transport, you decide on the transport mode before you attach objects to transport requests in the development system. You do this on the *CTS EXPORT* tab in SAP HANA Application Lifecycle Management. If you want to transport selected changelists, you select the *Released changelists* tab, assign the DU to CTS, and attach selected changelists to a transport request. If you want to transport all changelists in a DU, you select the *Delivery Units* tab, assign the DU to CTS, and attach it to a transport request.

Characteristics of the Transport Modes

Transport of *Selected Changelists* or *Released Changelists*

If you choose this mode, every time you trigger a transport, you can select the released changelists that you want to transport. Only released changelists can be transported. If any of the selected changelists contains predecessor changelists, then they are included in the transported as well.

As a prerequisite for the transport, all objects in the changelist must be assigned to a package that is assigned to a DU. The transport route must also be configured for this DU. Additionally, for CTS transport, the DU must be assigned to CTS. For more information, see *Recommendations on the Transport of Changelists*.

Caution

Make sure that all objects in the changelist belong to packages that are assigned to a DU. Otherwise, these objects are **not transported**. The system displays a warning icon for changelists in which only parts can be transported. However, the transportable parts are still transported.

For each changelist, a `.tgz` file (DU archive) is imported into a workspace on the SAP HANA target system. This means that for multiple changelists, multiple DU archives (`.tgz` files) are imported. Activation of the objects occurs once – at the end of the import process.

After transporting selected changelists, these changelists are no longer available for transport.

This mode allows flexible transports, because individual changes can be transported. The performance is usually good, because usually only few objects are transported. Consistency is guaranteed because predecessor changelists are calculated.

Transport of *All Changelists* or *Delivery Unit*

If you choose this mode, all active objects in the DU that are part of released changelists are transported. This mode provides a fast and productive way of transporting the entire DU. You can use this mode if you want to make sure that no active objects in the DU that are part of released changelists are excluded from the transport.

The transport entity is one DU archive (`.tgz` file).

Related Information

[Create a Transport Route \[page 29\]](#)

[Export Delivery Units for CTS Transport \[page 42\]](#)

[Export Changelists for CTS Transport \[page 43\]](#)

[Predecessor Changelists in SAP HANA Change Recording \[page 65\]](#)

[Remarks on the Transport \[page 45\]](#)

[Understanding of native transport of changes in HALM !\[\]\(ee63952771dcddb5555d089c684847fd_img.jpg\)](#)

[Recommendations on the Transport of Changelists \[page 72\]](#)

6.4.2 Recommendations on the Transport of Changelists

In general, we recommend that you assign objects of packages assigned to only one delivery unit (DU) to a changelist to ensure consistent transport of all objects in the changelist.

Note

The following recommendations are valid for the native SAP HANA transport of changelists, both for the transport modes *Selected Changelists* and *All Changelists*.

For CTS transport, you must make sure that all objects that you want to transport belong to packages in DUs that are assigned to CTS transport. If a DU is not assigned to CTS transport, the related objects are not transported and you are not notified about this.

We recommend that you make sure that all objects of a changelist are transported on one transport route. To ensure this, the following prerequisites must be met:

- All objects in the changelist must be assigned to packages that are assigned to a delivery unit (DU).
- The transport route must be configured for this DU.

In this case, all objects of the changelist are transported when you start the transport on this transport route, and the changelist will no longer be offered for transport.



Example

DU A has the package `aaa` assigned to it and DU B has the package `B` assigned to it. Transport route `TAB` is configured for DUs A and B. Changelist `CL1` contains objects of packages `aaa` and `bbb`. When you transport changelist `CL1` on transport route `TAB`, all objects that are part of this changelist are transported.

Problems if the objects in one changelist are transported on different transport routes

If for some reason there are objects of packages of more than one DU in the same changelist and you use a transport route that is not configured for all DUs, only the objects in the packages of the assigned DU will be transported. The objects in the other DUs will not be transported on this transport route.

In this case, you must transport the changelist on both transport routes to make sure that all objects are transported. Depending on the transport route, only objects in packages of the assigned DU are transported.



Example

DU A has the package `aaa` assigned to it, DU B has the package `bbb` assigned to it, and DU C has the package `ccc` assigned to it. Transport route `TAB` is configured for DUs A and B. Transport route `C` is configured for DU C. Changelist `CL2` contains objects of packages `aaa`, `bbb`, and `ccc`. When you start the transport for changelist `CL2` on transport route `TAB`, all objects of the DUs A and B (packages `aaa` and `bbb`)

are transported. Objects of DU C (package ccc) are not transported on this transport route. However, when you start the transport of selected changelists on transport route C, the changelist is offered for transport. Only objects of DU C (package ccc) will be transported. When you start the transport of all changelists on transport route C, the complete DU C will be transported.

➔ Recommendation

We do not recommend that the transport is distributed on different transport routes.

If you are in the situation that you have configured different transport routes for different DUs and you cannot transport all objects of the changelist in one transport, we recommend that you proceed as follows: To make sure that all objects are transported on the same transport route, re-assign all packages with related developments to the same DU. Keep in mind the *Consequences of Package - DU Reassignments*. The link is in the *Related Information*.

Only if you cannot assign all related packages to the same DU, you can configure the transport route for multiple DUs to make sure that the objects of related developments are transported together.

i Note

If you encounter performance issues when working with change recording, see the blog about *Performance aspects by transporting of HANA change objects* (see link below).

Related Information

[Blog: Understanding of native transport of changes in HALM](#) 

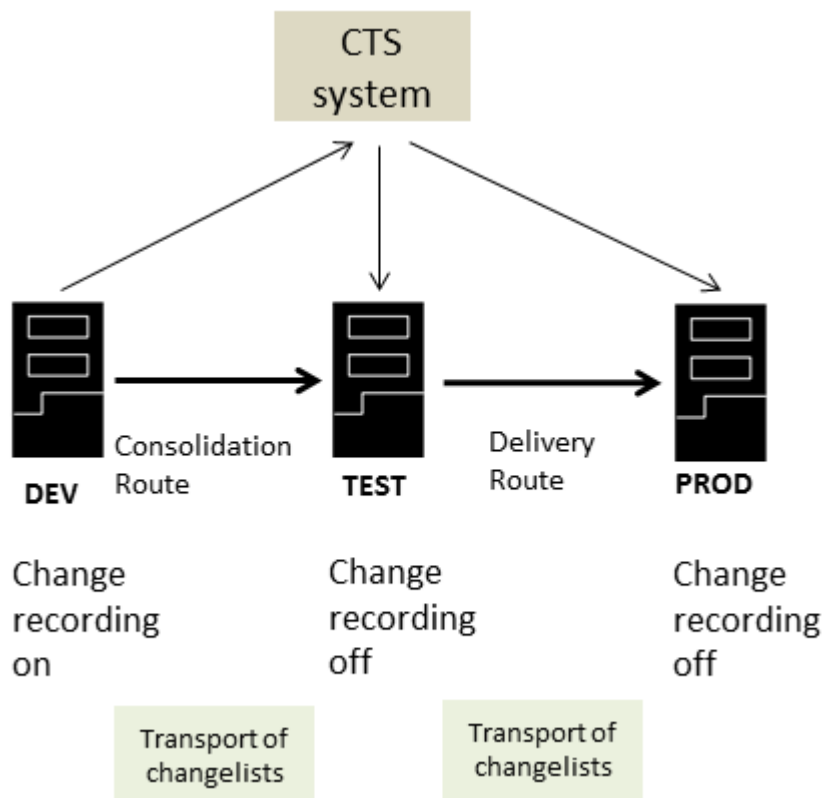
[Blog: Performance aspects by transporting of HANA change objects](#) 

[Consequences of Package - DU Reassignments \[page 66\]](#)

[Transport Modes in Change Recording \[page 70\]](#)

6.4.3 CTS and Changelists

If you use CTS transport, you can set up your transport track in such a way that object changes are made in the development system only and that changes are transported in changelists to the follow-on systems in your transport track.



When you configure your transport track, you must keep the following points in mind:

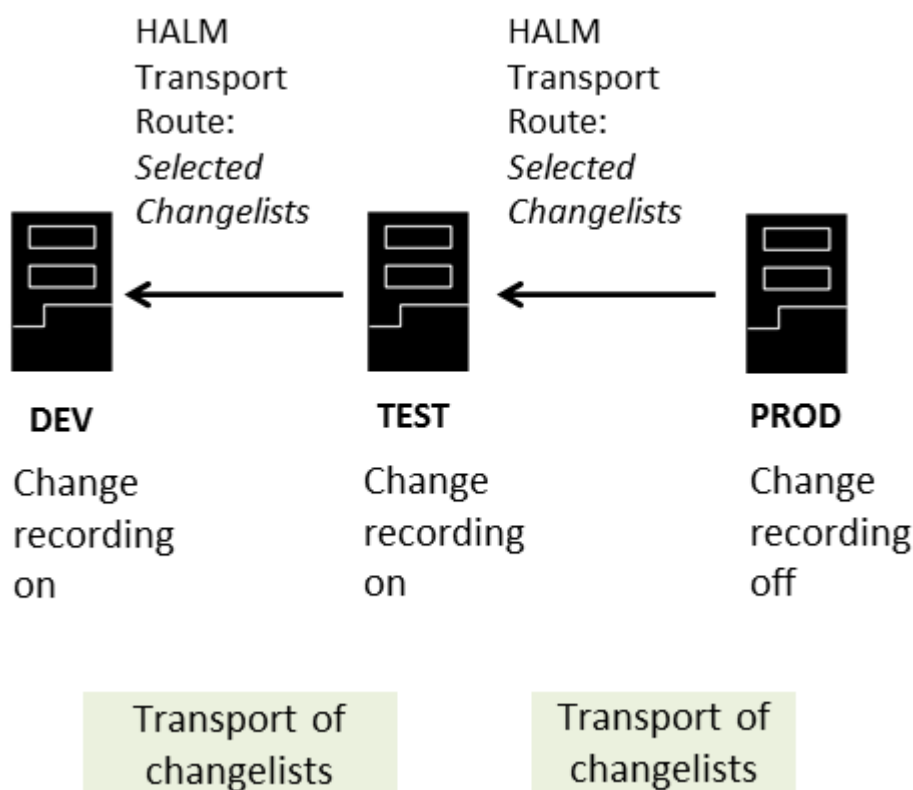
- Enabling of change recording
Change recording is enabled in the development system and it is disabled in all follow-on systems.
- Configuration of transport routes in CTS
You define a consolidation route between the development and the test system and a delivery route between the test and the production system.
- Transport Mode
To add the changelists to a transport request, you choose ► [CTS EXPORT](#) ► [Released changelists](#) ► on the development system in SAP HANA Application Lifecycle Management.
- Configuration of non-ABAP systems in CTS
The development system is configured as source system in CTS, and all other systems are configured as target systems.

6.4.4 Native SAP HANA and Selected Changelists

If you use native SAP HANA transport, you can set up your transport track in such a way that object changes are made in the development system only and that the changes are transported in changelists to the test system as well as to the production systems.

⚠ Caution

SAP does **not** recommend this landscape setup. If change recording is also enabled in the test system, a new changelist is created in the test system when changelists are imported, no matter how many changelists are imported. There is no relation between the imported changelists and the new changelist. The new changelist would be used for further transport. Therefore, the flow of software changes through the landscape cannot easily be tracked.



To obtain this transport behavior, you must make the following configuration settings:

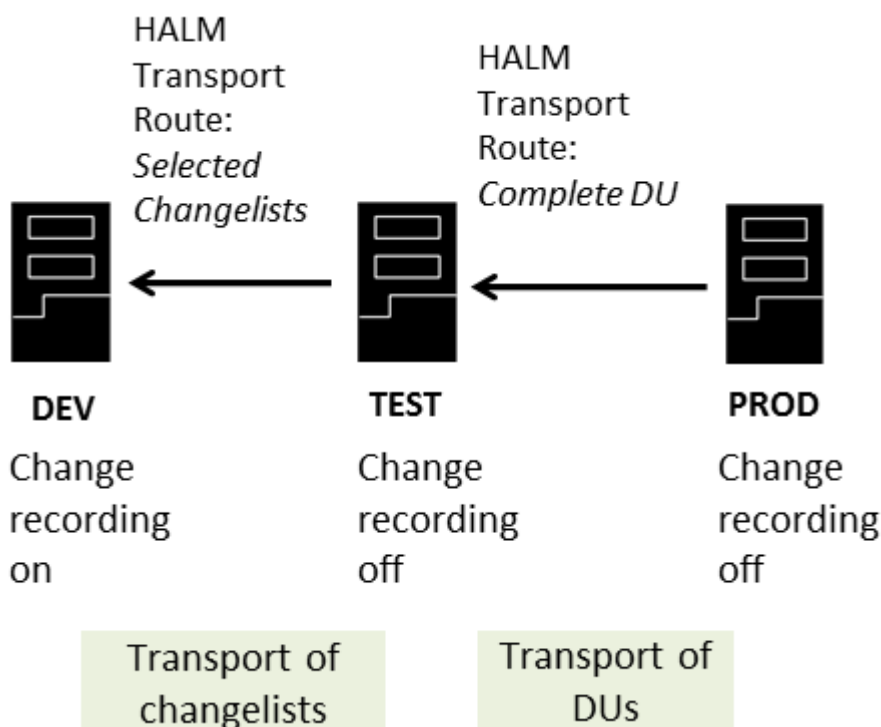
- Enabling of change recording
Change recording is enabled in the development system and in the test system. It is disabled in the production system.
- Transport routes in SAP HANA Application Lifecycle Management
You configure all transport routes with transport mode [Selected Changelists](#). For more information, see [Create a Transport Route](#).

Related Information

[Create a Transport Route \[page 29\]](#)

6.4.5 Native SAP HANA and Complete DU

If you use native SAP HANA transport, you can set up your transport track in such a way that object changes are made in the development system only and that changes are transported in changelists from the development system to the test system. From the test system, complete delivery units are transported to the production system. This requires exports both in the development and in the test systems.



To obtain this transport behavior, you must make the following configuration settings:

- Enabling of change recording
Change recording is enabled in the development system, and it is disabled in the test system and the production systems.
- Transport routes in SAP HANA Application Lifecycle Management
You must configure the transport route between the development and the test system with the transport mode *Selected Changelists*, and the subsequent transport routes with the transport mode *Complete DU*. For more information, see *Create a Transport Route*.

Related Information

[Create a Transport Route \[page 29\]](#)

6.5 Enable SAP HANA Change Recording

You enable change recording in your development system to manage changes to repository objects.

Prerequisites

- An SAP HANA system is available.
- You have the privileges granted by a role based on the `sap.hana.xs.lm.roles::Administrator` user role template.
- You have informed yourself about the implications of enabling change recording. For more information, see *Technical Details of Initial Change Recording Setup*. The link to this topic is in the *Related Information* section.

Context

The system administrator should enable change recording. If a system is configured to work with change recording, the activation of a repository object prompts developers to assign the object to a changelist. A changelist thus contains a list of one or more changed objects. This allows you to work on a development object or artifact and release the changelist only when the object is ready to be transported to the test system. This provides more precise control over which objects are transported from the development system. An object's changelist must be released in order to be included in the export in which the delivery unit containing the object is transported. Releasing a changelist does not trigger any automatic semantic checks but is a manual confirmation by you that the objects are consistent and ready for transport.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.

2. Choose the **SETTINGS** tab.
3. Select the **Enable Change Recording** checkbox.

i Note

When you enable change recording for your development system, the system initially records all active objects in the system. This process may take some time. During the process, the UI does not respond and there is no progress indicator telling you that the base changelist is being created. For more information, see *Technical Details of Initial Change Recording Setup*.

Results

Change recording is enabled.

Note

If you want to disable change recording, first verify that there are no open changelists in the system. You can disable change recording by repeating these steps and deselecting the [Enable Change Recording](#) checkbox.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Technical Details of Initial Change Recording Setup \[page 64\]](#)

6.6 Create Changelists

You can create a changelist in SAP HANA Application Lifecycle Management for your user, and add contributors to it.

Prerequisites

- Change recording is enabled in your development system.
- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::DevelopmentExpert` role template.

Context

You usually create a changelist while you are editing an object in the SAP HANA development environment. When you activate changes to an SAP HANA repository object, you are prompted to assign the changes to a changelist. If no changelist is available, you can create a new one.

Alternatively, you can create a changelist in SAP HANA Application Lifecycle Management. This is useful, for example, if you are a project manager and you want to create a changelist for multiple users who work closely together. You can add a description to the changelist, for example, to further specify for which developments the changelist is to be used.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.

2. Choose the **CHANGES** tab.
3. Choose **Create**.
4. Optional: Add a description for the changelist that you want to create and choose **Create**.

The changelist is created in the status **Open** with you as a contributor. A unique changelist ID is created. The changelist is added to the list of changelists on the left hand side of the screen. On the right hand side of the screen, the details of the changelist are displayed. This includes an **Contributions** area as well as an **Objects** area. Both areas are initially empty.

5. Optional: You can add more users as contributors to the changelist. To do this, choose **Add** in the **Contributions** area and select the users that are supposed to work on the changelist.

You can add a comment to the contribution. When prompted for a changelist in the development environment, this changelist is offered for selection to the contributors. The changelist can only be released if all contributors have approved their contributions.

6. Optional: You can change the changelist description. To save the changed description, choose **Save**.

Results

You have created a changelist. If required, you have added contributors to the changelist.

All contributors to the changelist can now assign their changes to the changelist.

You can also remove contributors from the changelist, as long as it is an empty contribution.

You can delete a changelist, as long as it is open and no objects are assigned to it.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Assign Objects to Changelists \[page 80\]](#)

6.7 Assign Objects to Changelists

You can assign an object to a changelist if change recording is configured and enabled in your development environment.

Prerequisites

- Change recording is enabled in your development system.
- You have the following system privileges on the SAP HANA system:
 - If you assign objects in your development environment: `REPO.MODIFY_CHANGE`, `REPO.MODIFY_OWN_CONTRIBUTION`.
 - If you want to move objects in SAP HANA Application Lifecycle Management from one changelist to another: privileges granted by a role based on the `sap.hana.xs.lm.roles::DevelopmentExpert` role template.

Context

You have the following options to assign objects to changelists:

- You can assign objects while you are performing software development in your SAP HANA development environment. Change recording is integrated in SAP HANA studio and Web-based Development Workbench. If change recording is enabled, the system prompts you to assign the object to a changelist when you activate the object.

Note

If the change recording is disabled from the SAP HANA Application Lifecycle Management system, the information will not be reflected in the activation process until the SAP HANA Studio is restarted.

- You can move objects from one open changelist to another open changelist in SAP HANA Application Lifecycle Management.

Note

The following is an example of assigning objects in SAP HANA Web-based Development Workbench.

Assign objects in SAP HANA Web-based Development Workbench

Procedure

1. Open the SAP HANA Web-based Development Workbench Editor tool.
The Editor tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/ide/editor`
2. When you create or activate changes to an existing object, a dialog window opens where you can select or create a changelist to which you can assign the object.
 - a. If you are already contributor to changelists, these changelists are displayed in the table. You can select the relevant changelist for the object and choose [Assign](#).
 - b. To create a new changelist, choose [New Changelist ...](#). Enter a description for the changelist, and choose [Create](#). You can assign the object to the new changelist.

Move objects from one changelist to another

Procedure

1. Open the SAP HANA Application Lifecycle Management.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.
From the SAP HANA Web-based Development Workbench Editor tool, you can choose ► [Navigation Links](#) ► [Lifecycle Management](#) ► to open the SAP HANA Application Lifecycle Management.
2. Choose the [CHANGES](#) tab.
3. Select a changelist that contains objects that you want to move to another changelist.
4. In the [Objects](#) area, select the objects that you want to assign to another changelist, and choose [Move](#).
5. Select the target changelist and choose [Move](#).

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

6.8 Approve Contributions to Changelists

By approving your contribution to the changelist, you mark that your work on the changelist is finished. The changelist can only be released after all contributors have approved their contribution.

Prerequisites

- Change recording is enabled in your development system.
- You have made a contribution to the changelist.
- You have the privileges granted by a role based on one of the SAP HANA Application Lifecycle Management role templates:
 - To approve your own contribution: `sap.hana.xs.lm.roles::Developer` role
 - To approve contributions of others: `sap.hana.xs.lm.roles::DevelopmentExpert` role

Procedure

1. Open the SAP HANA Application Lifecycle Management.
The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.
2. Choose the **CHANGES** tab.
 - a. To approve your own contribution: Choose **Approve My Contribution**.
 - b. To approve contributions of others: In the **Contributions** area, select the user whose contribution you want to approve and choose **Approve**.
3. You can enter a comment that is saved for your contribution. Choose OK.
The status of the contribution changes from *open* to *approved*.

Results

The contribution to the changelist was approved.

As long as the changelist is open, you can edit the comment of the contribution.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

6.9 Release Changelists

After all contributions of changelists are approved, you can release the changelist.

Prerequisites

- Change recording is enabled in your development system.
- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::DevelopmentExpert` role template.
- The contribution status must be *approved* for all contributors of a changelist before you can release the changelist, and the changelist status must be *Open*.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.

2. Choose the *CHANGES* tab.
3. From the list of open changelists, select the changelist that you want to release.
4. If all contributors have approved their contributions, you can choose *Release*.

If the *Release* button is greyed out make sure that all contributions are approved and that the changelist is in status *Open*. The status of the changelist changes from *Open* to *Released*.

Results

The changelist is released and it is ready to be transported.

If you have left the default filter settings unchanged, the released changelist disappears from the list of open changelists.

To display it, change the filter settings so that released changelists also appear in the list. If you select the released changelist, details are displayed, including the release date and time and the user who released it.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

[Filter and Search for Changelists \[page 84\]](#)

6.10 Filter and Search for Changelists

The filter function allows you to filter the changelists according to different filter criteria and display all changelists that meet the criteria. If the list of displayed changelists is very long, you can search the list for specific changelists.

Prerequisites

- Change recording is enabled in your development system.
- You have the privileges granted by a role based on one of the SAP HANA Application Lifecycle Management role templates:
 - `sap.hana.xs.lm.roles::Developer`
 - `sap.hana.xs.lm.roles::DevelopmentExpert`

Filter Changelists

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.

2. Choose the **CHANGES** tab.

By default, changelists are displayed that correspond to the following criteria:

- Status **Open**.
- Local changelists
- Log-on user is contributor of the changelist

The filter criteria are displayed next to the **Set Filter** link.

3. To change the filter criteria, choose **Set Filter**.
4. In the **Filter** dialog box, filter options are available, such as contributor, creator, release interval, contribution status, or changelist status. To remove the filter for a specific filter criterion, select the blank field from the dropdown list of the filter criterion. If you deselect the **Only Local Changelists** option, changelists that were imported into the current system are displayed in addition to changelists that were created locally in the system. You can only deselect this option if you also change the default value for **changelist status**.

If you remove the filter for the changelist status, it is possible that changelists in status **Suspended** are displayed. A changelist gets the status **Suspended** if change recording is disabled while the changelist is in status **Open**. If change recording is later re-enabled, previously open changelists can no longer be used. However, to provide a history of changelists, they are documented in the system with the status **Suspended**.

You have the following options:

- To display the changelists according to the specified filter criteria without closing the dialog box, choose [Apply](#).
- To close the dialog box and display the changelists according to the specified filter criteria, choose [Apply and Close](#).
- To reset the filter criteria to the default values, choose [Reset](#).
- To close the dialog box without applying the selected filter criteria, choose [Close](#).

The changelists that meet your filter criteria are displayed in the list.

When you restart SAP HANA Application Lifecycle Management, the default filter settings are re-set.

Search for Changelists

Procedure

1. Enter the character string for which you want to search in the search field.

Use a sequence of characters that really exists. You cannot use wildcards. All the changelists that correspond to your string and that match the currently selected filter are displayed in a dropdown list.

2. If the changelist you were searching is in the dropdown list, select it there.

The details of the changelist are displayed.

Related Information

[SAP HANA Application Lifecycle Management \[page 5\]](#)

7 Assembling Add-On Products and Software Components

To ship SAP HANA add-on product archives or software components archives, you must bring the developed software into a format that can later be installed in another system. To do this, you use the `hdbalm assemble` command.

Prerequisites

- You have modeled and developed your SAP HANA content along the guidelines laid out in the *SAP HANA Developer Guide*.
- You have access to `hdbalm` and you have the permissions required to use `hdbalm`. For more information, see the link to *Using hdbalm* in the *Related Information* section.

Context

Add-on product archives are `*.zip` files that contain one or more software component archives plus the metadata files `stack.xml` and `pd.xml` required for installation.

Software component archives are `*.zip` files (in previous versions, also `*.sar` files exist as software component archives) that contain one delivery unit archive file each plus (optionally) corresponding translation DUs and the metadata file `SL_MANIFEST.XML` required for installation. Support Packages or patches to add-on products are usually shipped as single software component archives.

For more information on the archive types, see *SAP HANA Content* in the *SAP HANA Administration Guide*. The link to the guide is in the *Related Information* section.

You can build `*.zip` archives for add-on products and software components using the `hdbalm assemble` command. These can later be installed using SAP HANA Application Lifecycle Management.

When assembling your developments into installable archives, the required metadata files are added to the archives which contain relevant installation information, such as required database versions, or other dependencies. In addition, you can add language DUs to the archives.

For more information on the `assemble` command, see the *Related Information* section.

Procedure

1. Start a command line client and navigate to the directory where `hdbalm` is located.
You can also add this directory to your path.

2. Start the assembly of your product or software components.

- a. Use the `assemble` command with the options that you require.

For information on the available options, you can use the `hdbalm help assemble` command. You can also read the `hdbalm assemble Command` documentation. The link is in the *Related Information* section.

To assemble an add-on product, you can specify one product name or multiple software component names.

Results

If no errors occur, the assembled product archive file or software component archive files can be found in the local directory or in the directory that you specified.

If errors occur during the assembly, an error message indicates the reason for the error and the system provides a log file with more detailed information.

Related Information

[hdbalm assemble Command \[page 127\]](#)

[SAP HANA Application Lifecycle Management \[page 5\]](#)

8 Installing and Updating SAP HANA Products and Software Components

SAP HANA application lifecycle management provides functions for installing and updating SAP HANA products that you have downloaded from the SAP Support Portal, or that you have assembled yourself.

Context

SAP HANA products consist of software components which are deployed to the SAP HANA repository. You have the following options to install and update SAP HANA products and software components:

- Using a **SAP Fiori application** integrated in the **SAP HANA Application Lifecycle Management XS application**. This application can be started in the following ways:
 - Start the SAP HANA Application Lifecycle Management on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`. Afterwards, choose the **INSTALLATION** tab or tile.
 - Start SAP HANA cockpit at the following URL: `http://<host_FQDN>:<port>/sap/hana/admin/cockpit`. Afterwards, choose the **Install Products and Software Components** tile in the **SAP HANA Application Installation and Update** group.
For more information about how to start SAP HANA cockpit, see *Open SAP HANA Cockpit* in this *SAP HANA Administration Guide*. The link is in the *Related Information* section.

For more information about using SAP HANA Application Lifecycle Management to install and update SAP HANA products and software components, see *Installing and Updating SAP HANA Products* and *Installing and Updating SAP HANA Software Components*.

- Using the **hdbalm commandline tool**.
To start **hdbalm**, start a command line client and navigate to the directory where **hdbalm** is located. You can also add this directory to your path.
For more information about using **hdbalm** to install and update SAP HANA products and software components, see *Using hdbalm* and *hdbalm install Command* in the *SAP HANA Application Lifecycle Management Guide*.

Related Information

[Installing and Updating SAP HANA Products \[page 89\]](#)

[Installing and Updating SAP HANA Software Components \[page 91\]](#)

[Installation and Update Options \[page 93\]](#)

[Using hdbalm \[page 114\]](#)

[hdbalm install Command \[page 123\]](#)

[SAP HANA Application Lifecycle Management \[page 5\]](#)

8.1 Installing and Updating SAP HANA Products

You can install and update SAP HANA products using SAP HANA application lifecycle management.

Prerequisites

- You have a product archive of an SAP HANA product that you want to install or update.

i Note

An SAP HANA product archive is a *.zip file that contains one or more software component archives as well as metadata files. For more information about the archive types that are used to deliver SAP HANA content, read the information about *SAP HANA content* in the *SAP HANA Administration Guide*.

- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role template.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.

2. Choose the **INSTALLATION** tab.
3. Click in the **Archives** selection field to select a product archive from your file directory that you want to install or update.

The product archive is uploaded. The header area contains information about the product version, including the action that is to be performed: Installation or Update.

4. The **Product Instances** tab lists all product instances that are part of the archive.

For each product instance, the result of a software component check is displayed.

The following results can occur:

- **OK**
The product instance can be installed.
- **Downgrade**
The product instance contains one or more software components that are already installed in newer versions than the ones to be installed. The installation of this product instance would lead to a downgrade of these software components. Downgrades are not allowed. To continue the installation of the product, you have to set the installation/update option **Keep newer version of software component**. In this case, the downgrading software components will be skipped during the installation of the product instance.

i Note

If it is required that you install the software component that causes the downgrade, for example, if the newer version has errors and you want to revert to the previous version, you can use the `install` command of `hdbadm` with the option `ALLOW_DU_DOWNGRADE` to enable the downgrade. However, use this option with care, since this may affect other installed products which require the newer version of this software component.

- *Some software components are installed already*

If software components are already installed in the same version, by default, the system skips their installation during the installation/update of the product instance. If you want to reinstall the same version, you can set the option *Overwrite the same version of software component* in the installation and update options.

Click in the line of the product instance to display more information about the software components that are part of the product instance. For each software component, a status is displayed, as well as the installed version and the new version. If you click on the status icon, you get more information about the status.

5. If required, set installation or update options.

The options allow you to override the default behavior of the installation or update for specific situations. Use them with care. For more information about the options, see *Installation and Update Options*.

6. Select product instances for installation.

You can individually select single product instances. To install all instances, select the *Instance* check box in the header row.

All instances of the product, that are already installed on your system will automatically be checked for updates. If the archive that you uploaded contains newer versions for one or several software components, they will automatically be updated. It doesn't matter whether you selected the respective instance for installation.

7. To start the installation, choose *Install*.

The system displays the progress of the individual installation steps. You can click on each step to expand the log of the step.

Results

If errors occur during the installation or update, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA software component or product instance that caused the error. The *Support Information* tab contains the relevant information. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

If the installation or update finished successfully, you can start another installation using *New Installation*.

Related Information

[Installation and Update Options \[page 93\]](#)

[Installing and Updating SAP HANA Products and Software Components \[page 88\]](#)

8.2 Installing and Updating SAP HANA Software Components

You can install and update SAP HANA software components using SAP HANA application lifecycle management.

Prerequisites

- You have one or multiple archives of SAP HANA software components that you want to install or update.

Note

An SAP HANA software component archive is a *.zip file that contains one delivery unit archive (*.tgz) as well as metadata files. For more information about the archive types that are used to deliver SAP HANA content, read the information about *SAP HANA content* in this *SAP HANA Administration Guide*.

Software components which need to be installed at the operating system level, such as Application Function Libraries (AFLs), are **not** installed using SAP HANA application lifecycle management.

- You have the privileges granted by the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.

2. Choose the *INSTALLATION* tab.
3. Click in the *Archives* selection field to select one or multiple software component archives from your file directory that you want to install or update.

The software component archives are uploaded. For each software component the following information is displayed:

- *Status*

The following status values exist:

- [New](#)
The software component is not yet installed and can be installed.
- [Update](#)
The software component is already installed and can be updated to a higher version.
- [Downgrade](#)
The software component is already installed in a newer version than the one that is to be installed. The installation of this software component would lead to a downgrade. Downgrades are not allowed. You cannot continue to install the software component.

i Note

If it is required that you install the software component that causes the downgrade, you can use the `install` command of `hdbalm` with the option `ALLOW_DU_DOWNGRADE` to enable the downgrade. However, use this option with care, since this may affect already installed products which require the newer version of this software component.

- [Already installed](#)
If software components are already installed in the same version, by default, the system skips their installation during the installation/update. If you want to reinstall the same version, you can set the option [Overwrite the same version of software component](#) in the installation and update options.
 - [Version](#) that is already [installed](#) in the system
 - [New version](#) that is to be installed
 - Whether the software component is part of a product that is already installed.
 - [Information](#) relevant for the [support](#)
4. If required, set installation or update options.

The options allow you to override the default behavior of the installation or update for specific situations. Use them with care. For more information about the options, see [Installation and Update Options](#).
 5. To start the installation, choose [Install](#).

The system displays the progress of the individual installation steps. You can click on each step to expand the log of the step.

Results

If errors occur during the installation or update, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA software component that caused the error. You can find this information in the support information of the component. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

If the installation or update finished successfully, you can start another installation using [New Installation](#).

Related Information

[Installation and Update Options \[page 93\]](#)

8.3 Installation and Update Options

Installation and update options are available that allow you to influence the installation and update behavior, if required.

Installation and Update Options

Option	Corresponding Installation Option in hdbalm	Description
Overwrite the same version of software component	ALLOW_DU_SAME_VERSION	By default, the system does not install a software component if the same version is already installed. It is possible to override this behavior in the following situations: <ul style="list-style-type: none"> • If a previous installation operation failed, for example, because of activation errors. • If you run continuous integration scenarios in which the same version of a software component is installed regularly.
Keep newer version of software component	ALLOW_KEEP_DU_NEWER_VERSION	If the product instance to be installed contains software components with lower versions than the installed ones, installing the software component in the lower version would lead to a downgrade of this software component. This is not allowed. You cannot install a downgrading software component. If you want to install the product instance without the downgrading software components, you can use this option. This option is useful if a software component is part of several products. If the product to be installed contains the software component in a version which is lower than the one already installed, you can choose to retain the newer version. In this case, the installation of the software component is skipped.
Allow version update	ALLOW_DU_VERSION_UPDATE	Allows version updates of software components. In some cases, for example, if a software component is part of several products, a version update of a software component could render one product inoperable. If the system detects an inconsistency, it aborts the operation. You can use this option to turn off this behavior.

Option	Corresponding Installation Option in hdbalm	Description
Roll back installation if activation errors occur (default)	This is the default behavior in hdbalm.	<p>By default, the installation is canceled if any activation errors occur and the complete installation is rolled back.</p> <p>Installation is also rolled back if you modified objects in your system and a modified object cannot be activated because it references an object that is part of the installation archive. This can occur, for example, if a procedure or view references a table in the archive.</p>
Ignore activation errors of referencing objects	USE_TWO_COMMIT_ACTIVATION	<p>If an installation fails since an object outside of the archive cannot be activated due to references to an object in the archive, you can repeat the installation with this activation option. In this case, the object remains broken in the system after the installation, but the installation itself finishes successfully. You must correct the errors manually after the installation.</p> <p>You can check the transport log after performing the installation without this option to find out whether the activation errors were caused by objects in the archive or outside of the archive. After repeating the installation with this option, check the transport log to find out which objects must be repaired afterwards.</p>

Related Information

[hdbalm install Command \[page 123\]](#)

[Installing and Updating SAP HANA Products and Software Components \[page 88\]](#)

9 Installing and Updating Products and Software Components in SAP HANA XS Advanced Model

Application lifecycle management for SAP HANA XS advanced model provides functions for installing and updating products as well as individual software components of SAP HANA XS advanced that you have downloaded from the SAP Support Portal.

Prerequisites

- The prerequisites described under *Prerequisites and Authorizations* are fulfilled. The link to the topic is in the *Related Information*.
- You have an SAP HANA XS advanced product or software component archive that you want to install or update.

Note

An SAP HANA XS advanced software component archive is a *.zip file that consists of a multi-target application archive (MTA archive = *.mtar file) and an `SL_MANIFEST.xml` file that contains metadata, such as version, vendor, support package, and patch level for the MTA archive.

An SAP HANA XS advanced product archive is a *.zip file that consists of one or multiple software component archives plus a `pd.xml` and a `stack.xml` file. Both files contain metadata for the product, such as version, support package level, and vendor.

Context

Note

From SPS 11, SAP HANA includes an additional run-time environment for application development: SAP HANA extended application services (XS), advanced model. SAP HANA XS advanced model represents an evolution of the application server architecture within SAP HANA by building upon the strengths (and expanding the scope) of SAP HANA extended application services (XS), classic model. SAP recommends that customers and partners who want to develop new applications use SAP HANA XS advanced model. If you want to migrate existing XS classic applications to run in the new XS advanced run-time environment, SAP recommends that you first check the features available with the installed version of XS advanced; if the XS advanced features match the requirements of the XS classic application you want to migrate, then you can start the migration process.

To install and update products and software components in SAP HANA XS advanced, the `xs install` command is available in the XS advanced command-line interface (CLI). Using this command you can install or update one product archive or one software component archive at a time.

Procedure

1. Start the XS advanced command-line interface (CLI).
2. Log on to the SAP HANA XS advanced runtime in the organization and space in which you want to install or update the product or software component.

To do this, use the `xs login` command with the following arguments and options:

Argument/Option	Description
<code>-u</code>	SAP HANA database user with the permissions as described in the <i>Prerequisites</i> section
<code>-p</code>	password
<code>-o</code>	organization in which the installation or update takes place
<code>-s</code>	space in which the installation or update takes place

Sample Code

```
xs login -u demo -p test -o demoorg -s demospace
```

For more information, see the *XS CLI: Logon and Setup* topic in *SAP HANA Developer Guide (for SAP HANA XS Advanced Model)*. The link is in the *Related Information* section.

3. Start the installation or update of the product or software component.

The `xs install` command is available in the XS advanced CLI both for installing product and software component archives in XSA and updating these. The `xs install` command detects whether the archive is a product archive or a software component archive. It also detects whether the product or software component is installed already and subsequently executes either an installation or update operation.

Enter the `xs install` command and specify the path to the archive. If required, enter any additional options. For example, to install a specific instance of a product, you can use the `-i` option and specify the product instance. Or to make sure that the entity you are about to install is a product, you can add the `-pv` option. In this case, the installation is only performed if you specify a product archive for the `xs install` command. If you specify a software component archive, the installation is not performed.

Sample Code

```
xs install ../sap_demo/target/XSASAMPLEPRODUCT1.0.zip
```

Instead of `xs install` you can also use the `xs ins` alias. For more information on the options, see *Installation and Update Options in XS Advanced Model*. For installation examples, see *Examples: Installing and Updating Products and Software Components in XS Advanced Model*. The links are in the *Related Information* section.

Results

Before installing or updating the product or software component, the system performs different checks. If no errors are found, the system performs the installation or update with the arguments and options that you specified. During the process, the product installer calls the deploy service that performs the actual deployment. Afterwards, the product installer registers the product or software component as installed.

If the installation or update cannot be performed, it is possible, in some situations, to use additional options to override the default behavior of the system. For more information, see *Checks Before Installing or Updating Products or Software Components in XS Advanced Model* and *Installation and Update Options in XS Advanced Model*.

If errors occur during the installation or update, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA product or software component that caused the error. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

To display the correct log file, use one of the following commands with the log ID that you find in the result of the installation or update process.

- To display the log of a product installation, use the `display-installation-logs` command with the `--pv` option.

```
xs display-installation-logs <log ID> --pv
```

- To display the log of a software component installation, use the `display-installation-logs` command with the `--scv` option.

```
xs display-installation-logs <log ID> --scv
```

Related Information

[Prerequisites and Authorizations \[page 98\]](#)

[Installation and Update Options in XS Advanced Model \[page 102\]](#)

[Examples: Installing and Updating Products and Software Components in XS Advanced Model \[page 104\]](#)

9.1 Prerequisites and Authorizations

The following prerequisites have to be fulfilled when you use functions required for installing and updating SAP HANA products and software components in SAP HANA XS advanced model.

- The XS advanced run time is installed and available on the SAP HANA server.
For more information, see *Installing XS Advanced Runtime* in the *SAP HANA Server Installation and Update Guide*.
- Optional when using the command line interface (CLI): The XS command line client is installed on your local machine.
The XS CLI client tools are installed by default on the SAP HANA server. You can log on to the server and execute the installation command there. However, if you want to connect to SAP HANA from your local machine, you must download and install the client tools locally. The XS CLI client tools (`xs.onpremise.runtime.client_<platform>-<version>.zip`) can be downloaded from the SAP HANA server, from the installation DVD, or from the SAP support portal.
- The SAP HANA database user that is used to perform the installation or update has one of the following permissions assigned:
 - The user has the `XS_CONTROLLER_USER` parameter assigned as well as the *SpaceDeveloper* role for each space in which the user wants to perform an installation or update.
 - The user has the `XS_CONTROLLER_ADMIN` parameter assigned.
This scope allows the installation in all spaces.

For more information on assigning roles in SAP HANA XS advanced, see *Setting Up Security Artifacts* in the *SAP HANA Administration Guide*.

9.2 Set Up a Virus Scan for Installation Archives

You can set an environment variable in your system to enable a default virus scan for all software component archives that you want to install or update.

Prerequisites

You have installed and configured the SAP virus scan interface as described in SAP Note [786179](#) .

Context

If the antivirus software that you are using does not check the software component archives that you want to install or update, you can use the SAP virus scan interface and set the environment variable `SCAN_UPLOADS` to the value `true`. This way, the system checks all archives that you want to install or update.

By default, no antivirus protection is set for the product installer.

Procedure

1. In the commandline tool, set the XSA environment variable `SCAN_UPLOADS` to **true**.

Sample Code

```
xs set-env product-installer SCAN_UPLOADS true
```

For more information about setting environment variables in XS advanced, see *12.2 XS CLI: Application Management* in the *SAP HANA Developer Guide For SAP HANA XS Advanced Model*. The link to the guide is in the *Related Information* section.

2. Restart the product installer.

The restart is required to ensure that the change to the environment variable takes effect.

Sample Code

```
xs restart product-installer
```

For more information about restarting applications in XS advanced, see *12.2 XS CLI: Application Management* in the *SAP HANA Developer Guide For SAP HANA XS Advanced Model*. The link to the guide is in the *Related Information* section.

Related Information

[Installing and Updating Products and Software Components in SAP HANA XS Advanced Model \[page 95\]](#)

9.3 Checks Before Installing or Updating Products or Software Components in SAP HANA XS Advanced Model

To ensure consistency of SAP HANA products, the system executes different checks before installing or updating a product or a software component in SAP HANA XS advanced.

Product installations only: Check whether the product to be installed is already installed and in which version

If the product to be installed is not yet installed, the installation will be performed. If it is already installed, the system checks the installed version. If it is already installed in the same version, or in a lower support package level, the installation or update will be performed.

- Product is already installed in higher version
If the version of the product to be installed is lower than the installed version, the system terminates the process because installing the lower version would lead to a downgrade of the product.
You can override this behavior and allow a downgrade of the product. To do this, you can use the `ALLOW_PV_DOWNGRADE` option with the `xs install` command.
- Product is already installed in lower version
If the version of the product to be installed is higher than the installed version, the system updates the installed version automatically.

Note

The version of a product usually consists of one or more numbers in an ascending order. In addition to the version number, a support package level is provided for the product.

Example

The version number is 1.0. In this case, the following versions are considered version updates: 1.1, 2.0, or 2.

Check whether the software component is already installed and in which version

If the software component to be installed was not yet installed, the installation will be performed. If it was already installed, the system checks the installed version. If it is installed in a lower support package or patch level, the update will be performed.

Note

The version of a software component has the following form: "`#.#.#`", for example 1.0.3, where

- 1 = the version
 - 0 = the support package level
 - 3 = the patch level
- Software component is already installed in higher version
If a version of a software component to be installed is lower than an installed version, the system terminates the installation.
You have the following options to override this behavior:
 - You can allow a downgrade of the software component. To do this, use the `ALLOW_SC_DOWNGRADE` option.
 - For product installation only: You can skip the installation of all software components that are part of the archive and that are already installed in higher versions. To do this, use the `ALLOW_KEEP_SC_NEWER_VERSION` option.
 - Software component is already installed in same version
If a version of a software component to be installed is the same as the installed version, the system proceeds as follows:

- Product installation: The system does not install this software component. The installation of this software component is skipped during the installation of the product.
- Software component installation: The system terminates the installation.

You can override this behavior and allow the reinstallation of the same version. To do this, use the `ALLOW_SC_SAME_VERSION` option for this software component.

i Note

If the software component is installed in the system in the same version with the status *BROKEN*, it is automatically reinstalled.

- Software component is already installed in lower version
If a version of a software component to be installed is higher than an installed version, the system updates the installed version automatically.

Check for dependencies on SAP HANA platform components or other XS advanced components

If the software component has dependencies on SAP HANA platform components or other XS advanced components that are not installed, the system terminates the process and displays the missing software components. You must install or update the missing software components before you can restart the current installation or update.

For more information on the options to override the default behavior, see *Installation and Update Options in XS Advanced Model*. The link is in the *Related Information* section.

Check whether extension descriptor is valid, if an extension descriptor is used

If an extension descriptor is used for the installation process, the system checks that the extension descriptor file does not exceed a specific file size and that the syntax of the extension descriptor file is correct. If the file is too big or if the syntax is incorrect, the system will not start the installation process.

For more information on extension descriptors, see the *The MTA Deployment Extension Description* topic in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*.

Related Information

[Installing and Updating Products and Software Components in SAP HANA XS Advanced Model \[page 95\]](#)

[Installation and Update Options in XS Advanced Model \[page 102\]](#)

[Display installed Products and Software Components in XS Advanced Model \[page 108\]](#)

9.4 Installation and Update Options in XS Advanced Model

Installation and update options are available in SAP HANA XS advanced that allow you to influence the installation and update behavior, if required.

The following is the default syntax for the `xs install` command in the XS advanced CLI:

```
xs install <ARCHIVE> [-p <TARGET_PLATFORM>] [-pv | --PRODUCT_VERSION] [-scv | --  
SOFTWARE_COMPONENT_VERSION] [-t <TIMEOUT>] [-e  
<EXT_DESCRIPTOR_1>[,<EXT_DESCRIPTOR_2>]] [-o  
<VERSION_OPTION_1>[,<VERSION_OPTION_2>]] [-i | --INSTANCES  
<INSTANCE_1>[,<INSTANCE_2>]] [--delete-services] [--delete-service-brokers] [--no-  
start] [--ignore-lock]
```

The following is an example of a product installation:

```
xs install /sap_demo/target/XSASAMPLEPRODUCT1.0.zip -pv -o ALLOW_SC_SAME_VERSION
```

For more installation examples, see *Examples: Installing and Updating Products and Software Components in XS Advanced Model*. The link is in the *Related Information* section.

Installation and Update Arguments

Argument	Description
<ARCHIVE>	The path to (and name of) the archive containing the product or software component (SCV) to install, update, or downgrade

Installation and Update Options

Option	Description
-p <TARGET_PLATFORM>	Specify the target platform where the product or software component will be installed. If not specified explicitly, a target platform is created implicitly as ' <code><ORG> <SPACE></code> '.
-pv --PRODUCT_VERSION	Install a product. The installation is performed only if the given archive is a product archive. Otherwise, the installation will fail.
-scv -- SOFTWARE_COMPONENT_VERSION	Install a software component. The installation is performed only if the given archive is a software component archive. Otherwise, the installation will fail.
-e <EXT_DESCRIPTOR_1>[,<EXT_DESCRIPTOR_2>]	Define one or more extensions to the installation/deployment descriptors; multiple extension descriptors must be separated by commas. For more information on extension descriptors, see the <i>The MTA Deployment Extension Description</i> topic in the <i>SAP HANA Developer Guide for SAP HANA XS Advanced Model</i> . The link is in the <i>Related Information</i> section.

Option	Description
<code>-t <TIMEOUT></code>	Specify the maximum amount of time (in seconds) that the installation service must wait for the installation operation to complete
<code>-o</code> <code><VERSION_OPTION_1>[, <VERSION_OPTION_2>]</code>	<p>Specify options which can be used to override the default behavior of the <code>install</code> command. The following options are available:</p> <ul style="list-style-type: none"> ALLOW_PV_DOWNGRADE Allows a downgrade of the product. By default, the system does not install a product if the product is already installed in a higher product version or support package stack since this would lead to a downgrade of the product. It is possible to override this behavior, for example, if the newer version has errors and you want to revert to the previous version. This option is available for product installations only. ALLOW_KEEP_SC_NEWER_VERSION Skips the installation of a software component if a newer version is already installed in the system. By default, the system does not install a product if a newer version of one of the software components contained in the product archive is already installed. It is possible to override this behavior. This option is useful, for example, if a software component is part of several products. If the product to be installed contains the software component in a lower version than the one already installed, you can choose to retain the newer version. If you use this option, the installation of this software component is skipped. This option is available for product installations only. ALLOW_SC_DOWNGRADE Allows a downgrade of the software component. By default, the system does not install a software component if this leads to a downgrade of the software component. It is possible to override this behavior, for example, if the newer version has errors and you want to revert to the previous version. <div style="background-color: #fff9c4; padding: 10px; margin: 10px 0;"> <p>⚠ Caution</p> <p>Use this option carefully.</p> </div> <ul style="list-style-type: none"> ALLOW_SC_SAME_VERSION Reinstalls the same version of the software component. By default, the system does not install a software component, if the same version is already installed. It is possible to override this behavior, for example, if you run continuous integration scenarios in which the same version of a software component is installed regularly.
<code>-i, --INSTANCES</code> <code><INSTANCE_1>[, <INSTANCE_2>]</code>	By default all instances are installed; a comma-separated list of instances can be specified to limit the number of instances installed
<code>--delete-services</code>	Recreate changed services and/or delete discontinued services

Option	Description
<code>--delete-service-brokers</code>	Delete discontinued service brokers
<code>--no-start</code>	Do not start applications that are updated during the installation
<code>--ignore-lock</code>	Force installation even if the space targeted for installation is locked
<code>--deploy-passthrough</code> <{"<key>" : "<value>", ...}>	Option for the deploy service

See also the `xs install` command reference in the *XS CLI: Plugin Management* topic in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*. The link is in the *Related Information* section.

Related Information

[Installing and Updating Products and Software Components in SAP HANA XS Advanced Model \[page 95\]](#)
[Examples: Installing and Updating Products and Software Components in XS Advanced Model \[page 104\]](#)

9.5 Examples: Installing and Updating Products and Software Components in XS Advanced Model

The examples show how you can use the `xs install` command.

In the following examples you must be logged on to the XS command line interface (XS CLI) with a user with the authorizations required for installation and in the organization and space in which you want to perform the installation.

- **Installation of new product**

The following example installs the product *XSA Sample Product* in version 1.0, SPS 0 (initial shipment stack) contained in the file `XSASAMPLEPRODUCT_1.0.zip`:

```
XSA Sample Product (sap.com) 1.0 SPS 0
Product Instance 1
SCV_A 1.0.0
SCV_B 1.0.0
```

No version of this product is currently installed. The following command is used:

```
xs install XSASAMPLEPRODUCT_1.0.zip
```

After the installation, the `xs list-products` command displays the product as follows:

```
name                vendor    version  SPS    instance ids
-----
XSA Sample Product  sap.com   1.0      0      1
```

The detail display for *XSA Sample Product* looks as follows:

```
xs list-products "XSA Sample Product"
-----
name                XSA Sample Product
vendor              sap.com
version             1.0
support package stack 0
-----
instance id         software component      version      state
-----
1                   -                      -            SUCCESS
                   SCV_A                  1.0.0        SUCCESS
                   SCV_B                  1.0.0        SUCCESS
-----
```

- **Update with support package stack**

The following example installs the product *XSA Sample Product* in version 1.0, SPS 5 contained in the file XSASAMPLEPRODUCT_1.0.5.zip in the system:

```
XSA Sample Product (sap.com) 1.0 SPS 5
Product Instance 1
SCV_A 1.5.0
SCV_B 1.5.0
```

Version 1.0, SPS 0 (initial shipment stack) of *XSA Sample Product* containing software components SCV_A in version 1.0.0 and SCV_B in version 1.0.0 is currently installed. To ensure that the archive to be installed is a product archive, the `-pv` option is used.

```
xs install XSASAMPLEPRODUCT_1.0.5.zip -pv
```

After the update, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
-----
name                XSA Sample Product
vendor              sap.com
version             1.0
support package stack 5
-----
instance id         software component      version      state
-----
1                   -                      -            SUCCESS
                   SCV_A                  1.5.0        SUCCESS
                   SCV_B                  1.5.0        SUCCESS
-----
```

- **Installation of lower support package version**

The following example installs the product *XSA Sample Product* in version 1.0, SPS 3 contained in the file XSASAMPLEPRODUCT_1.0.3.zip in the system:

```
XSA Sample Product (sap.com) 1.0 SPS 3
Product Instance 1
SCV_A 1.3.0
SCV_B 1.3.0
```

Version 1.0, SPS 5 of *XSA Sample Product* containing software components SCV_A in version 1.5.0 and SCV_B in version 1.5.0 is currently installed. If the installation was started without any options, it would fail. To allow the downgrade of the support package version, you must use the `ALLOW_PV_DOWNGRADE`

option. In addition, to allow a downgrade of the software components, you must use the `ALLOW_SC_DOWNGRADE` option.

```
xs install XSASAMPLEPRODUCT_1.0.3.zip -o ALLOW_PV_DOWNGRADE,
ALLOW_SC_DOWNGRADE
```

After the installation, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
-----
name                XSA Sample Product
vendor              sap.com
version             1.0
support package stack 3
-----
instance id  software component  version  state
-----
1            -                   -         SUCCESS
            SCV_A               1.3.0     SUCCESS
            SCV_B               1.3.0     SUCCESS
-----
```

- **Installation of higher product version**

The following example installs the product *XSA Sample Product* in version 2.0, SPS 1 contained in the file `XSASAMPLEPRODUCT_2.0.1.zip` in the system:

```
XSA Sample Product (sap.com) 2.0 SPS 1
Product Instance 1
SCV_A 2.1.0
SCV_B 2.1.0
```

Version 1.0, SPS 3 of *XSA Sample Product* containing software components `SCV_A` in version 1.3.0 and `SCV_B` in version 1.3.0 is currently installed. To allow the installation of a higher product version, you must use the `ALLOW_PV_VERSION_UPDATE` option. In addition, to allow the installation of higher software component versions, you must use the `ALLOW_SC_VERSION_UPDATE` option.

```
xs install XSASAMPLEPRODUCT_2.0.1.zip -o ALLOW_PV_VERSION_UPDATE,
ALLOW_SC_VERSION_UPDATE
```

After the installation, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
-----
name                XSA Sample Product
vendor              sap.com
version             2.0
support package stack 1
-----
instance id  software component  version  state
-----
1            -                   -         SUCCESS
            SCV_A               2.1.0     SUCCESS
            SCV_B               2.1.0     SUCCESS
-----
```

- **Installation of lower product version**

The following example installs the product *XSA Sample Product* in version 1.5, SPS 3 contained in the file `XSASAMPLEPRODUCT_1.5.3.zip` in the system:

```
XSA Sample Product (sap.com) 1.5 SPS 3
Product Instance 1
```

```
SCV_A 1.3.5
SCV_B 1.3.5
```

Version 2.0, SPS 1 of *XSA Sample Product* containing software components SCV_A in version 2.1.0 and SCV_B in version 2.1.0 is currently installed. To allow a downgrade of the product version, you must use the ALLOW_PV_DOWNGRADE option with the command. In addition, to allow a downgrade of the software components, you must use the ALLOW_SC_DOWNGRADE option.

```
xs install XSASAMPLEPRODUCT_1.5.3.zip -o ALLOW_PV_DOWNGRADE,
ALLOW_SC_DOWNGRADE
```

After the installation, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
-----
name                XSA Sample Product
vendor              sap.com
version             1.5
support package stack 3
-----
instance id         software component      version      state
-----
1                   -                     -            SUCCESS
                   SCV_A                 1.3.5        SUCCESS
                   SCV_B                 1.3.5        SUCCESS
-----
```

- **Installation of software component**

The following example installs the software component *SCV_A* in version 1.2.3 contained in the file `SCV_A_123.zip`. No version of this software component is currently installed. The `-scv` option is used to make sure that the archive to be installed is a software component archive.

```
xs install SCV_A_123.zip -scv
```

After the installation, the `xs list-components` command displays the software component as follows:

```
xs list-components
software component      version
-----
SCV_A (sap.com)         1.2.3
```

- **Installation of product with lower version of software component**

The following example installs the product *XSA Test Product* in version 1.0, SPS 3 contained in the file `XSATESTPRODUCT_1.0.3.zip`. No version of the product is currently installed. However, the product contains the software component *SCV_A* in version 1.0.3 which was already installed individually in version 1.2.3.

You have the following options to proceed with the installation:

- To allow a downgrade of the software component, you can use the ALLOW_SC_DOWNGRADE option with the command.

```
xs install XSATESTPRODUCT_1.0.3.zip -o ALLOW_SC_DOWNGRADE
```

After the installation, the `xs list-components` command displays the software component as follows:

```
xs list-components
software component      version
-----
```

```
SCV_A (sap.com) 1.0.3
```

- To keep the newer version of the software component, you can use the `ALLOW_KEEP_SC_NEWER_VERSION` option with the command.

```
xs install XSATESTPRODUCT_1.0.3.zip -o ALLOW_KEEP_SC_NEWER_VERSION
```

After the installation, the `xs list-components` command displays the software component as follows:

```
xs list-components
software component      version
-----
SCV_A (sap.com)        1.2.3
```

9.6 Display installed Products and Software Components in XS Advanced Model

To display products and software components of SAP HANA XS advanced that are already installed, the `xs list-products` and `xs list-components` commands are available.

Prerequisites

The prerequisites are fulfilled as described in the *Prerequisites and Authorizations* topic. The link is in the *Related Information* section.

Context

Instead of `xs list-products` you can also use the `xs lp` alias. Instead of `xs list-components` you can also use the `xs lc` alias.

For more information, see the *XS CLI: Plugin Management* topic in *SAP HANA Developer Guide (for SAP HANA XS Advanced Model)*. The link is in the *Related Information* section.

Procedure

1. Start the XS advanced command-line interface (CLI).
2. Log on to the SAP HANA XS advanced runtime in the organization and space where you want to display installed products or software components.
3. You have the following options:

- a. To display all products that are installed in the current organization and space, use the `xs list-products` command without any arguments.

```
xs list-products
```

The system lists all installed products with information about vendor, version, support package level and installed instances.

- b. To display all software components that are installed in the current organization and space, use the `xs list-components` command.

```
xs list-components
```

The system lists all installed software components with information about vendor and version. The version is displayed in the format `<software component version>.<support package level>.<patch level>`.

- c. To display detailed information for a specific installed product, use the `xs list-products` command and add the name of the product `<PRODUCT NAME>` as argument. Optionally, or if another product with the same name and different vendor exists, add the `<VENDOR>`. The following is an example:

```
xs list-products XSASAMPLEPRODUCT sap.com
```

Note

If the product name contains a space, enter the product name in quotation marks.

```
xs list-products "SAMPLE PRODUCT" sap.com
```

The system lists the specified product with information about vendor, version, and support package level. In addition, it lists all installed product instances and the software components that are assigned to the instances. For these, it lists the version and the state in which the software component exists in the system. They can have the following states:

- **SUCCESS**: The software component or product instance is successfully installed.
- **BROKEN**: The software component or product instance is installed in a broken state. This status can occur, for example, if there was an error in the deploy step during installation. Product instances can get this status, for example, after one software component of the product instance was uninstalled.
- **RUNNING**: The installation of this software component or product instance is currently running.

Example

Examples:

- The output for the `xs list-products` command can look as follows:

Sample Code

name	vendor	version	SPS	instance ids
XSA Sample Product	sap.com	1.0	0	1,3

- The output for the `xs list-products "XSA Sample Product"` command can look as follows:

Sample Code

```
-----
name      XSA Sample Product
vendor    sap.com
version   1.0
SP        0
-----
instance id  software component  version  state
-----
1            -                  1.0      SUCCESS
            JAVA_HELLO_XSA_B    1.0.0    SUCCESS
            JAVA_HELLO_XSA_A    1.0.0    SUCCESS
3            -                  1.0      SUCCESS
            JAVA_HELLO_XSA_C    1.1.0    SUCCESS
            JAVA_HELLO_XSA_D    1.1.0    SUCCESS
-----
```

Related Information

[Prerequisites and Authorizations \[page 98\]](#)

9.7 Uninstall Products and Software Components in SAP HANA XS Advanced Model

Application lifecycle management for SAP HANA XS advanced model provides functions for uninstalling products as well as individual software components of SAP HANA XS advanced.

Prerequisites

- The prerequisites described under *Prerequisites and Authorizations* are fulfilled. The link to the topic is in the *Related Information*.
- You have a product or software component of SAP HANA XS advanced that you want to remove.

Context

You can uninstall products and software components of SAP HANA XS advanced that were installed using the `xs install` command.

Procedure

1. Start the XS advanced command-line interface (CLI).
2. Log on to the SAP HANA XS advanced runtime in the organization and space where you want to uninstall an installed product or software component.
3. Optional: Display the product or software component using the `xs list-products` or `xs list-components` command.
4. Start the uninstallation of the product or software component.

Enter the `xs uninstall` command and specify the name of the product or software component to be uninstalled, as well as the vendor, if required. In addition, you can enter options as required. The following is the default syntax for the `xs uninstall` command in the XS advanced CLI:

```
xs uninstall <NAME> [<VENDOR>] [-pv | --PRODUCT_VERSION] [-scv | --  
SOFTWARE_COMPONENT_VERSION] [-f] [--ignore-scv-reuse] [--delete-services] [--  
delete-service-brokers] [--ignore-lock]
```

The following arguments are available:

Uninstallation Arguments

Uninstallation Argument	Description
<NAME>	The name of an installed product version (PV) or software component version (SCV)
[<VENDOR>]	The name of the vendor of the specified product or software component version; optional : only needed when the same product or software component name exists with different vendors

The following options are available:

Uninstallation Options

Uninstallation Option	Description
-pv --PRODUCT_VERSION	Remove the specified product. To make sure that the entity you are about to uninstall is a product, you can add the <code>-pv</code> option. In this case, the uninstallation is only performed if you specify a product name as <NAME>. If you specify a software component name, the uninstallation will fail.
-scv --SOFTWARE_COMPONENT_VERSION	Remove the specified software component. To make sure that the entity you are about to uninstall is a software component, you can add the <code>-scv</code> option. In this case, the uninstallation is only performed if you specify a software component name as <NAME>. If you specify a product name, the uninstallation will fail.

Uninstallation Option	Description
<code>--ignore-scv-reuse</code>	<p>Remove the specified software component even if it is used in other installed products.</p> <div> <p>i Note</p> <p>You can use this option for uninstalling software components only.</p> </div> <p>By default, a software component will not be uninstalled if it is also part of another installed product. You can override this behavior by using the <code>--ignore-scv-reuse</code> option.</p>
<code>-f</code>	Remove the specified product or software component without any system prompts or confirmation
<code>-i, --INSTANCES</code> <code><INSTANCE_1>[, <INSTANCE_2>]</code>	By default all instances are uninstalled; a comma-separated list of instances can be specified to limit the number of instances to be uninstalled
<code>--delete-services</code>	Recreate changed services and/or delete discontinued services
<code>--delete-service-brokers</code>	Delete discontinued service brokers
<code>--ignore-lock</code>	Force removal of the product or software component even if the target space is locked

Sample Code

```
xs uninstall 'XSA SAMPLE PRODUCT' -pv
```

Instead of `xs uninstall` you can also use the `xs unins` alias. For more information on the `xs uninstall` command, use the `xs help uninstall` command.

Results

The system undeploys and unregisters the specified product or software component from the SAP HANA server in the organization and space to which you are logged on.

If errors occur during the uninstallation, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA product or software component that caused the error. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

To display the correct log file, use the `xs display-installation-logs` command with the log ID that you find in the result of the uninstallation process and one of the `--unins_scv` or `--unins_pv` options.

```
xs display-installation-logs <log ID> --unins_scv
```

Related Information

[Prerequisites and Authorizations \[page 98\]](#)

10 Using hdbalm

SAP HANA provides the `hdbalm` command line tool to perform application lifecycle-management tasks.

Prerequisites

- You have performed the SAP HANA client installation.
If you have left the default installation options unchanged `hdbalm` is located in the `c:\Program Files\sap\hdbclient` directory on Microsoft Windows and `/usr/sap/hdbclient` directory on Linux. For more information, see the *SAP HANA Client Installation and Update Guide*.
- You have Python installed in your system with the latest 2.6 or 2.7 version.
`hdbalm` is written in the programming language Python. It requires a Python version 2.6 or 2.7. You have the following options:
 1. Use the Python version that comes with the `hdbclient` package and is located at `%dirname%Python`. Note that this Python version does not include SSL.
 2. On Unix, Python is usually installed as part of the operating system, or it can be installed using the package management provided by the system. As an alternative, and on Windows, you can download and install the latest 2.6 or 2.7 Python release from <https://www.python.org>.

To make sure that `hdbalm` uses the correct Python version, set the environment variable `PYTHON_HOME` to the location of the Python installation:

- On Windows, you can set the environment variable as shown in the following example:

```
set PYTHON_HOME="c:\Program Files\Python27"
```

Note

Make sure you set the value for this environment variable in quotation marks.

- On Unix (bourne shell), you can set the environment variable as shown in the following example:

```
export PYTHON_HOME=/usr/python27
```

- You have the permissions required to run `hdbalm`: You have an SAP HANA database user with the privileges granted by a role based on the SAP HANA `sap.hana.xs.lm.roles::Administrator` user role template assigned to it.
Note that you cannot use a newly created user that still has the initial password, since `hdbalm` will not ask you to change it. Change the password for this user in SAP HANA studio, for example, before using `hdbalm`.

Context

The command line tool `hdbalm` is part of the SAP HANA client installation (Microsoft Windows 64-bit and Linux 64-bit). You use `hdbalm` to **assemble** SAP HANA products and software components. You can also use `hdbalm`

to execute other application lifecycle management functions, if you prefer to use a command line tool over the SAP HANA XS user interface, or if you want to automate specific tasks.

i Note

You can also use all commands of `hdbalm` for tenant databases on multiple-container systems if you have specified the correct host and port of the tenant database. For more information, see *hdbalm Commands, Options, and Variables*.

Procedure

1. Start a command line client and navigate to the directory in which `hdbalm` is located (or add this directory to your path).
2. Optional: Set environment variables.
3. Enter the required `hdbalm` command with the required options, command options, or parameters.

Results

`hdbalm` executes the command. If errors occur, error messages indicate the reason for the errors. For the `install`, `import`, and `assemble` commands, the system provides log files with more detailed information.

Related Information

[Assembling Add-On Products and Software Components \[page 86\]](#)

[Installing and Updating SAP HANA Products and Software Components \[page 88\]](#)

[hdbalm Commands, Options, and Variables \[page 117\]](#)

[hdbalm install Command \[page 123\]](#)

[hdbalm assemble Command \[page 127\]](#)

[hdbalm import Command \[page 128\]](#)

[hdbalm transport Command \[page 129\]](#)

[hdbalm log Command \[page 130\]](#)

[hdbalm product Command \[page 131\]](#)

[hdbalm du Command \[page 132\]](#)

[hdbalm dependencies Command \[page 134\]](#)

[hdbalm package Command \[page 135\]](#)

[hdbalm admin Command \[page 136\]](#)

[Enable SSL for hdbalm \[page 121\]](#)

[Proxy Support for hdbalm \[page 122\]](#)

[SAP HANA Application Lifecycle Management \[page 5\]](#)

10.1 Using hdbalm with SAP HANA Cloud Platform

If you want to use `hdbalm` for SAP HANA systems in the SAP HANA Cloud Platform you must use secure communication (Secure Sockets Layer (SSL) protocol and certificates as described in the [Enable SSL for hdbalm](#) topic). This is required since systems in the SAP HANA Cloud Platform communicate using HTTPS.

If used in the SAP HANA Cloud Platform, `hdbalm` commands require the following options:

Option	Description	Sample Value
-h	host of the SAP HANA instance that is assigned to the account in the SAP HANA Cloud Platform	<code>demo123.hana.ondemand.com</code>
-p	port of the SAP HANA instance that is assigned to the account in the SAP HANA Cloud Platform	<code>443</code> This is the only port that allows HTTPS communication.
-s	indicates that HTTPS communication is used	no value
-c	valid certificate stored in <i>X.509 with chain (PEM)</i> format	<code>-.hana.ondemand.crt</code>

Example

Example of an installation of the SAP HANA product archive DEMOPRODARCH.zip on the SAP HANA Cloud Platform instance `demo123.hana.ondemand.com`:

Sample Code

```
hdbalm -u SYSTEM -h demo123.hana.ondemand.com -p 443 -s -c  
-.hana.ondemand.crt install DEMOPRODARCH.zip
```

Exporting a Certificate

To issue an `hdbalm` command in the SAP HANA Cloud Platform, a certificate is required to address the SAP HANA instance that is assigned to the SAP HANA Cloud Platform account for which you want to use `hdbalm` command. You can export this certificate using the Web browser's export function.

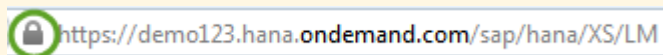
Example

The following is an example of how to export an *X.509 with chain (PEM)* certificate in Mozilla Firefox:

1. Start SAP HANA Application Lifecycle Management on the SAP HANA instance from which you want to export the certificate.

Example: `https://demo123.hana.ondemand.com/sap/hana/XS/LM`

2. Click on the Lock icon in the browser bar:



3. Choose *More Information...*
4. On the *Security* ribbon, choose *View Certificate*.
5. On the *Details* tab, choose *Export...*
6. Browse for a location where you want to store the file, and save the certificate as file of type *X.509 with chain (PEM)*.
The certificate is saved as *.crt file.

Related Information

[Enable SSL for hdbalm \[page 121\]](#)

[Documentation on SAP HANA Cloud Platform](#)

10.2 hdbalm Commands, Options, and Variables

With `hdbalm` you can use a selection of commands and their options to perform application lifecycle-management tasks in SAP HANA.

The following example depicts the syntax for `hdbalm`:

```
hdbalm [<general options>] <command> [<command-specific options>]
```

The general options are specified before the command and the command-specific options are specified after the command. Each command offers its own specific options.

Note

Enter `hdbalm` to display general information about the commands, options, and environment variables.

Enter `hdbalm help <command>` to display more information about a specific command and its options.

If you normally work in one environment, then you can set environment variables for your user, password, SAP HANA Extended Services (XS) engine host (including tenant database), and SAP HANA XS engine port. This way, you need not specify these details every time you use a command.

Example

Examples for Setting Environment Variables

On Microsoft Windows, you can set the environment variable for the user as in the following example:

```
set HDBALM_USER=<user name>
```

On Unix (bourne shell), you can set the environment variable for the user as in the following example:

```
export HDBALM_USER=<user name>
```

The following tables describe the various commands, options, and environment variables available for `hdbalm`.

hdbalm Commands





The following table contains the `hdbalm` commands and explains their functions.

Command	Description
help	Provides information about available commands, general options, and environment variables The <code>help</code> command also provides more information for every command using <code>hdbalm help <command></code> .
install	Provides functions for installing and updating product archives and software component archives (.zip files) that were downloaded from the SAP Support Portal
assemble	Provides functions for assembling SAP HANA add-on products and software components
import	Provides functions for importing delivery unit archives (.tgz files)
transport	Provides functions for managing transports, such as starting transports or displaying transport routes
log	Provides functions for displaying log files
product	Provides functions for managing SAP HANA products, such as creating a product or assigning delivery units to a product
du	Provides functions for managing delivery units, such as creating a delivery unit
dependencies	Provides functions for displaying delivery unit dependencies in the system
package	Provides functions for managing packages, such as creating packages and assigning packages to a delivery unit
admin	Provides administrative application lifecycle-management functions, such as enabling change recording

General hdbalm Options


The following options are supported by all `hdbalm` commands.

Option	Description
<code>-u <user></code> , <code>--user=<user></code>	User name

Option	Description
<code>-h <host>, --host=<host></code>	<p>SAP HANA Extended Services (XS) engine host</p> <div>  Example <code>101234.mycompany.com</code> </div> <p>If you want to use the command for a tenant database on a multiple-container system, then you specify the host of the tenant database. You do this by adding the system ID (SID) of the tenant database as an alias to the host name of the SAP HANA system. Use the following format: <code><Host name of SAP HANA system>-<SID of tenant database></code></p> <div>  Example <code>101234-DB1.mycompany.com</code> or <code>101234-DB1</code> </div> <p>For more information about where to find the SID of the tenant database, see <i>Configure HTTP Access to Multitenant Database Containers</i> in the <i>SAP HANA Administration Guide</i>. A link to this guide is provided in the <i>Related Information</i> section.</p>
<code>-p <port>, --port=<port></code>	<p>SAP HANA Extended Services (XS) engine port</p> <p>The default XS engine port is <code>80+<instance number></code>.</p> <p>If you set this option for a tenant database on a multiple-container system, make sure that you specify the correct port of the tenant database.</p>
<code>-v, --verbose</code>	Writes debug messages to standard error
<code>-s, --https</code>	Sends request using https
<code>-c <certificate>, --certs=<certificate></code>	Certificate file when using https
<code>-y, --yes</code>	<p>Runs command in non-interactive mode (does not prompt for confirmation)</p> <div>  Note This option is useful for automated mode. </div>
<code>-j, --json</code>	<p>Prints result in json notation if successful</p> <div>  Note This option is not available for all commands. </div>

Environment Variables

You can set the following environment variables:

Environment Variable	Description
HDBALM_USER	User name
HDBALM_PASSWD	Password
HDBALM_HOST	<p>XS engine host</p> <div> Example 101234.mycompany.com</div> <p>If you want to set the environment variable for a tenant database on a multiple-container system, specify the host as described above for option <code>-h <host></code>, <code>--host=<host></code>.</p>
HDBALM_PORT	<p>XS engine port</p> <p>The default XS engine port is <code>80+<instance number></code>.</p> <p>If you set the environment variable for a tenant database on a multiple-container system, make sure that you specify the correct port of the tenant database.</p>
http_proxy	<p>HTTP proxy</p> <p>For more information, see <i>Proxy Support for hdbalm</i>.</p>
https_proxy	<p>HTTPS proxy</p> <p>For more information, see <i>Proxy Support for hdbalm</i>.</p>
no_proxy	<p>Use this environment variable if you want to specify hosts and domains for which no proxy is to be used. You can enter a comma-separated list of hosts and domains.</p> <p>For more information, see <i>Proxy Support for hdbalm</i>.</p>

Note

The options `-u`, `-h`, and `-p` take precedence over environment variables. The program requests a password for the user if no password is set as an environment variable.

Related Information

[Proxy Support for hdbalm \[page 122\]](#)

10.3 Enable SSL for hdbalm

You can secure the communication between `hdbalm` and the SAP HANA system using the Secure Sockets Layer (SSL) protocol and certificates.

Prerequisites

You have Python installed in your system and set the environment variable `PYTHON_HOME` to the location of the Python installation as described in *Using hdbalm*.

Context

`hdbalm` is written in the programming language Python. To enable SSL for `hdbalm`, you must have a Python version installed in your system that includes SSL libraries and that supports secure HTTPS connections. SAP supports Python versions 2.6 and 2.7.

Note

For general information on how to use SSL and certificates to secure the SAP HANA server, see *Securing Data Communication* in the *SAP HANA Security Guide*.

Procedure

To enable secure communication in `hdbalm`, set the `hdbalm` option `-s` or `--https` and provide a valid certificate using the `-c <certificate>` or `--certs=<certificate>` options. The certificate is used to validate the identity of the SAP HANA server. The certificate needs to be stored in a file in X.509 PEM format.

Note

If you use these options with a Python version that does not support SSL, you receive an error message.

Results

You have enabled SSL for `hdbalm`.

Note

If you use proxies for communication, the proxy settings might also be relevant when you enable SSL. For more information, see *Proxy Support for hdbalm*.

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

[Proxy Support for hdbalm \[page 122\]](#)

[Using hdbalm \[page 114\]](#)

10.4 Proxy Support for hdbalm

hdbalm supports proxies both for HTTP and HTTPS communication.

If you can only access the SAP HANA system using a proxy, you can set the following environment variables so that hdbalm can connect to the SAP HANA system:

- `http_proxy`
You can set an HTTP proxy.
- `https_proxy`
You can set an HTTPS proxy.
- `no_proxy`
You can define that no proxy should be used for specific hosts and domains. You can specify a comma-separated list of hosts and domains for which no proxy is to be used.

Usually on Linux, the environment variables are already configured by your system administration. If not, you can set them as in the following example:

Example

```
export http_proxy=http://<host>:<port>/  
export https_proxy=http://<host>:<port>/  
export no_proxy=<.mycompany.com>
```

On Microsoft Windows, you can set them as described in the following example:

Example

```
set http_proxy=http://<host>:<port>/  
set https_proxy=http://<host>:<port>/  
set no_proxy=<.mycompany.com>
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.5 hdbalm install Command

Use this command and its corresponding options to install and update SAP HANA products (product archives and software component archives).

The `install` command is available both for installing product and software component archives (*.zip files) and for updating these. The `install` command detects whether the archive is an add-on product archive or a software component archive. It also detects whether the add-on product or software component is installed already and subsequently executes either an installation or update operation.

The following options are available to install or update **products**:

- You can specify an archive file.
- You can specify a directory location that contains unpacked archive files (usually shipped on DVDs).
- You can specify single instances by specifying a comma-separated list of instances.

The following options are available to install or update **software components**:

- You can specify one or more archive files.
- You can specify one or more directory locations that contain unpacked archive files.

The following code sample depicts the standard syntax in `hdbalm`.

```
hdbalm [<general options>] install [<command option>]* [<archive>|<directory>]*
```


Note

Command options are command-specific. For more information about the `install` command, enter `hdbalm help <command>` in `hdbalm`. Some command options depend on the archive type. For example, you can only use the `--instances` option for product archives.

The following table describes the command options available for the `install` command. For general options and environment variables that you can use with this command, see *hdbalm Commands, Options, and Environment Variables*. For examples of how to use the `install` command, see *Examples: hdbalm install Command* in the [SAP HANA Application Lifecycle Management Guide](#). Links to this guide and these topics are included in the *Related Information* section.

Command Options

Option	Description
<code>-d, --display</code>	Displays the contents of the archive No changes are applied to the system.
<code>-l <file name>, --log=<file name></code>	Sets an alternate location for the log file.

Option	Description
<pre>-o <installation option>, -- option=<installation option></pre>	<p>Provides installation options which can be used to override the default behavior if a version of the product or the software component is already installed</p> <p>Multiple options can be specified by repeating the <code>-o</code> option. The following installation options are available:</p> <ul style="list-style-type: none"> ALLOW_DU_DOWNGRADE Allows downgrades of software components. By default, the system does not install a software component if this leads to a downgrade of the software component. It is possible, however, to override this behavior, for example, if the newer version has errors and you want to revert to the previous version. This option may also be required if the versioning sequence was changed between Support Packages, for example, if SP09 has version 100.0.0 and SP10 has version 1.001.0. <div data-bbox="730 882 1396 1003">  Caution Use this option carefully. </div> <ul style="list-style-type: none"> ALLOW_DU_SAME_VERSION Reinstalls the same version of a software component By default, the system does not install a software component if the same version is already installed. It is possible to override this behavior in the following situations: <ul style="list-style-type: none"> If a previous installation operation failed, for example, because of activation errors If you run continuous integration scenarios in which the same version of a software component is installed regularly ALLOW_DU_VERSION_UPDATE Allows version updates of software components <code>hdbalm</code> attempts to keep the system in a consistent state. In some cases, for example, if a software component is part of several products, a version update of a software component could render one product inoperable. If <code>hdbalm</code> detects an inconsistency, it aborts the operation. You can use this option to turn off this behavior. ALLOW_KEEP_DU_NEWER_VERSION Allows to keep the version of the software component if it is installed already in a newer version This option is useful if a software component is part of several products. If the product to be installed contains the software component in a version which is lower than the one already installed, you can choose to retain the newer version. In this case, the installation of the software component is skipped. USE_TWO_COMMIT_ACTIVATION By default, the installation is canceled if any activation errors occur and the complete installation is rolled back.

Option	Description
	<p>Installation is also rolled back if you modified objects in your system and a modified object cannot be activated because it references an object that is part of the installation archive. This can occur, for example, if a procedure or view references a table in the archive.</p> <p>If an installation fails because an object outside of the archive cannot be activated due to references to an object in the archive, you can repeat the installation with this activation option. In this case, the object remains broken in the system after the installation, but the installation itself finishes successfully. You must correct the errors manually after the installation.</p> <p>You can check the transport log after performing the installation without this option to find out whether the activation errors were caused by objects in the archive or outside of the archive. After repeating the installation with this option, check the transport log to find out which objects must be repaired afterwards.</p> <div> <p>➔ Recommendation</p> <p>Do not use this option for installations into production systems.</p> </div>
--instances	<div> <p>i Note</p> <p>This option is available only for the installation and update of product archives.</p> </div> <p>By default all relevant instances are installed. A comma-separated list of instances can be specified here to install only particular product instances.</p>

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

[Installing and Updating SAP HANA Products and Software Components \[page 88\]](#)

[Examples: hdbalm install Command \[page 125\]](#)

[SAP HANA Application Lifecycle Management \[page 5\]](#)

10.5.1 Examples: hdbalm install Command

The examples show how you can use the `hdbalm install` command.

In the following examples, no environment variables are set. The general options `-u <user>`, `-h <host>`, and `-p <port>` are included in the command. When the user is included as a general `hdbalm` option in the command, you are prompted for a password after starting the command.

- The following example installs or updates the product contained in the file `SAP_APO_ANALYTICS_1.0.zip` on the XS engine host `101234.mycompany.com` and port `8000`.

i Note

The default XS engine port is `80+<instance number>`.

The installation or update is performed using the `SYSTEM` user. You are prompted for a password after starting the command.

```
hdbalm -u SYSTEM -h 101234.mycompany.com -p 8000 install
SAP_APO_ANALYTICS_1.0.zip
```

- The following example installs or updates the product contained in the file `SAP_APO_ANALYTICS_1.0.zip` on the tenant database with the SID `DB1` of the XS engine host `101234.mycompany.com` and port `8000`. The installation or update is performed using the `SYSTEM` user. You are prompted for a password after starting the command.

```
hdbalm -u SYSTEM -h 101234-DB1.mycompany.com -p 8000 install
SAP_APO_ANALYTICS_1.0.zip
```

In the following examples, the environment variables `HDBALM_USER`, `HDBALM_PASSWD`, `HDBALM_HOST`, and `HDBALM_PORT` are set. This way, you do not have to include this information as general options in the command.

- The following example installs or updates the product located in the directory `c:\products` `\SAP_APO_ANALYTICS` and writes the log file to the file `%TEMP%\install.log`:

```
hdbalm install -l %TEMP%\install.log c:\products\SAP_APO_ANALYTICS
```

- The following example installs or updates instances 1 and 2 of the product contained in the file `my_product.zip`. Any additional instances that might be part of this product archive are not installed:

```
hdbalm install --instances 1,2 my_product.zip
```

- The following command installs or updates the software components `SCV1` and `SCV2`:

```
hdbalm install scv1.zip scv2.zip
```

- The following command installs the software components `SCV1` and `SCV2`. The new version is installed even if either of the two components are installed already and the new version has a higher version or higher SP version than the installed software component:

```
hdbalm install --option=ALLOW_DU_VERSION_UPDATE scv1.zip scv2.zip
```

- The following command looks for software component files in the `c:\patches` directory and installs or updates the software components in the SAP HANA system:

```
hdbalm install c:\patches
```

Related Information

[hdbalm install Command \[page 123\]](#)

10.6 hdbalm assemble Command

Use this command to assemble SAP HANA add-on products and software components.

Use the following syntax for the `assemble` command:

```
hdbalm [<general options>] assemble [<command options>] [<name>,<vendor>]+
```

Note

If the vendor is unique in the system, you can omit it.

Command Options

The following command options exist:

Command Option	Description
<code>-d <directory>,--directory=<directory></code>	Specifies an alternate location for the assembled add-on product archives or software component archives
<code>-l <file name>,--log=<file name></code>	Sets an alternate location for the log file
<code>--languages <languages></code>	Comma-separated list of language codes that are exported for the software components
<code>--ignore_language_errors</code>	Ignores errors if languages are inconsistently configured for a delivery unit No language delivery unit is exported.
<code>--overwrite</code>	Overwrites archives if they exist in the file system
<code>--timestamp</code>	Adds a timestamp to the archive file name to distinguish between different assembly builds
<code>--products_only</code>	Assembles only product archives This can be required if a product and a delivery unit have the same name in the system.
<code>--scvs_only</code>	Assembles only software components This can be required if a product and a software component have the same name in the system.

Command Option	Description
<code>--export_version</code>	An export format can be specified to make the format compatible with older SAP HANA versions. By default, the current file format is used. This can be required if you want to install the assembled product or software component in an older SAP HANA version. For a mapping between file format and SAP HANA version, see <i>SAP Note 1984354</i> .

Example

The following example assembles the product `SAP APO ANALYTICS` of the vendor `sap.com` and writes the product archive to the local directory:

```
hdbalm assemble "SAP APO ANALYTICS",sap.com
```

Related Information

[SAP Note 1984354](#)

[Assembling Add-On Products and Software Components \[page 86\]](#)

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.7 hdbalm import Command

Use this command to import SAP HANA delivery units (`.tgz` files).

Use the following syntax for the `import` command:

```
hdbalm [<general options>] import [<command option>]* [<du tgz>|<directory>]*
```

Command Options

The following command options exist:

Command Option	Description
<code>-d, --display</code>	Displays the archive contents No changes are applied to the system.
<code>-l <file name>, --log=<file name></code>	Sets an alternate location for the log file

Example

The following example imports delivery units `mydu1` and `mydu2`:

```
hdbalm import mydu1.tgz mydu2.tgz
```

This command looks for delivery units in the `c:\delivery_units` directory and imports them:

```
hdbalm import c:\delivery_units
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.8 hdbalm transport Command

Use this command to execute transport-related SAP HANA application lifecycle-management activities, such as displaying transport routes and starting the transport for a specific transport route.

Use the following syntax for the `transport` command:

```
hdbalm [<general options>] transport <transport command>
```

Transport commands

The following `transport` commands exist:

Transport Command	Description
<code>list</code>	Lists available transport routes.
<code>start</code>	Starts a transport operation.

Example

The following example shows how to start a transport operation on the specified transport route:

```
hdbalm transport start <route id>
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.9 hdbalm log Command

Use this command to display logs for other commands.

Use the following syntax for the `log` command:

```
hdbalm [<general options>] log <log command> [<parameter>] *
```

Log Commands

The following `log` commands exist:

Log Command	Description
<code>list</code>	Lists available log entries.
<code>get</code>	Displays the log for another command.

Example

The following example shows how to display the log for a particular process ID.

```
hdbalm log get <ID>
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.10 hdbalm product Command

Use this command to manage SAP HANA add-on products.

Use the following syntax for the `product` command:

```
hdbalm [<general options>] product <product commands> [<command option>]*  
[<parameter>]*
```

Product Commands

The following product commands exist:

Product Command	Description
<code>list</code>	Lists all products installed in the system.
<code>get</code>	Displays metadata for the product.
<code>create</code>	<p>Creates a product in the system (metadata only). The vendor is set to the vendor name configured in the system.</p> <p>The supported command options are:</p> <ul style="list-style-type: none">• <code>-v <version>, --version=<version></code>• <code>-d <description>, --description=<description></code>
<code>delete</code>	Deletes the product (metadata only). No delivery units are removed from the system.
<code>createInstance</code>	<p>Creates a product instance for the specified product.</p> <p>There is one supported command option:</p> <p><code>d <description>, --description=<description></code></p>
<code>deleteInstance</code>	Deletes a product instance for the specified product. All assigned delivery units are unassigned.
<code>assign</code>	Assigns a delivery unit to a product instance.
<code>unassign</code>	Unassigns a delivery unit from a product instance.

Examples

The following example shows how to display metadata for the product.

```
hdbalm product get <product name> <vendor name>
```

The following example shows how to create a product in the system (metadata only). The vendor is set to the vendor name configured in the system.

```
hdbalm product create [<command option>]* <product name>
```

The following example shows how to delete the product (metadata only). It does not remove deliver units.

```
hdbalm product delete <product name> <vendor name>
```

The following example shows how to create a product instance for the specified product.

```
hdbalm product createInstance [<command option>] <product name> <vendor name>  
<instance id>
```

The following example shows how to delete a product instance for the specified product and ensures that all its assigned delivery units are unassigned.

```
hdbalm product deleteInstance <product name> <vendor name> <instance id>
```

The following example shows how to assign a delivery unit to a product instance.

```
hdbalm product assign <du name> <du vendor> <product name> <product vendor>  
<instance id>
```

The following example shows how to unassign a delivery unit from a product instance.

```
unassign product <du name> <du vendor> <product name> <product vendor> <instance  
id>
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.11 hdbalm du Command


Use this command to manage SAP HANA delivery units.

Use the following syntax for the `du` command:

```
hdbalm [<general options>] du <du command> [<command option>]* [<parameter>]*
```

DU Commands

The following `du` commands exist:

DU Command	Description
list	Lists all delivery units deployed in the system.
get	Displays metadata for the delivery unit.
create	<p>Creates a new delivery unit (metadata only).</p> <p>The supported command options are:</p> <ul style="list-style-type: none"> • <code>-v <version>, --version=<version></code> • <code>-r <responsible>, --responsible=<responsible></code> • <code>-d <description>, --description=<description></code> <p>The version syntax must use this format: a, a.b, or a.b.c, where a is the version number, b the version SP, and c the patch number.</p>
delete	Deletes a delivery unit (metadata only). No objects are removed from the system.
undeploy	<p>Undeploys a delivery unit. The delivery unit metadata and all objects are removed from the system.</p> <div>  Caution Use this command option with caution. </div>
make_local	This is a developer feature that sets the source system of a delivery unit to the local system. This is not supported for delivery units shipped by SAP.
languages	Retrieves the original language for a delivery unit and all translations available in the system.
set_original_language	<p>Sets the original language attribute for all packages that belong to the specified delivery unit.</p> <p>The language is either a two-character ISO 639-1 language code or a two-character ISO 639-1 language code followed by an underscore followed by a two-character ISO 3166-1 country code.</p>

Examples

The following example shows how to display metadata for the delivery unit.

```
hdbalm du get <du name> <du vendor>
```

The following example shows how to create a new delivery unit (metadata only).

```
hdbalm du create [<command option>]* <du name>
```

The following example shows how to delete a delivery unit (metadata only). This command does not remove objects from the system.

```
hdbalm du delete <du name> <du vendor>
```

The following example shows how to undeploy a delivery unit. This command removes delivery unit metadata and all objects from the system.

Caution

Use this command option with caution.

```
hdbalm du undeploy <du name> <du vendor>
```

The following example shows how to retrieve the original language for a delivery unit and all translations available in the system.

```
hdbalm du languages <du name> <du vendor>
```

The following example shows how to set the original language attribute for all packages that belong to the specified delivery unit.

```
hdbalm du set_original_language <du name> <du vendor> <language>
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.12 hdbalm dependencies Command

Use this command to display and analyze dependencies of SAP HANA delivery units.

Use the following syntax for the `dependencies` command:

```
hdbalm [<general options>] dependencies [<command option>]* [<source du>]  
[<source du vendor>] [<target du>] [<target du vendor>]
```

Command options

The following command options exist:

Command Option	Description
-f, --full	Shows the full dependency view and analysis.
-r, --references	Shows object references between delivery units. This command requires the name and vendor of the source and the target delivery units.
-n, --nirvana	Shows nirvana references for a delivery unit. This option displays object references for objects that are part of a delivery unit to objects which are not part of a delivery unit. If a delivery unit contains objects with these references, it cannot be imported into another system.

Note

If you do not specify any command options, a list of delivery unit dependencies is displayed. Each line of the output lists a delivery unit followed by a colon and a comma-separated list of referenced delivery units.

Example

To display the dependencies of the delivery unit `HANA_XS_LM`, you use the following command:

```
hdbalm dependencies HANA_XS_LM sap.com
```

The delivery unit `HANA_XS_LM` has references to the delivery units `SAPUI5_1` and `HANA_XS_BASE`. The output appears as follows:

```
HANA_XS_LM(sap.com) : SAPUI5_1(sap.com), HANA_XS_BASE(sap.com)
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.13 hdbalm package Command

Use this command to manage SAP HANA packages.

Use the following syntax for the `package` command:

```
hdbalm [<general options>] package <package command> [<parameter>]*
```

Package Commands

The following package commands exist:

Command Option	Description
create	Creates a new package.
delete	Deletes a package. The package must not contain any sub-packages or objects.
assign	Assigns a package to a delivery unit.

Example

The following example shows the syntax for assigning a package to a delivery unit.

```
package assign <du name> <du vendor> <package name>
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

10.14 hdbalm admin Command

Use this command to execute administrative commands in SAP HANA application lifecycle management.



Use the following syntax for the `admin` command:

```
hdbalm [<general options>] admin <admin command> [<parameter>]*
```

Admin Commands

The following `admin` commands exist:

Admin Command	Description
getvendor	Returns the vendor name of the system.
setvendor	Sets the vendor to the new vendor name.

Admin Command	Description
enablechangemanagement	<p>Enables change recording.</p> <div>  Caution <p>Enabling change recording makes all existing active objects part of a released changelist. In addition, all object in subsequent activations are assigned to changelists that are released afterwards. Only these changelists can be transported from the system.</p> </div>
disablechangemanagement	<p>Disables change recording.</p> <div>  Caution <p>Disabling change recording switches off change tracking in the system. This makes all existing objects transportable, including those active objects that were already part of open changelists.</p> </div>

Example

The following example shows the syntax for setting a new vendor.

```
hdbalm admin setvendor <new vendor>
```

Related Information

[hdbalm Commands, Options, and Variables \[page 117\]](#)

11 Configuring SAP HANA Applications with the Process Engine

The Process Engine (PE) is a framework available with SAP HANA application lifecycle management to enable automated technical configuration.

After the installation of a product or a delivery unit, an application typically must be configured before it can be used. The configuration tasks are described in the installation guides that are provided on the SAP Help Portal (help.sap.com). Instead of performing cumbersome and error-prone manual activities, you can use the Process Engine to automate application configuration completely or partially. As a prerequisite, your application must provide content for the automated technical configuration.

The Process Engine (PE) framework is installed with SAP HANA application lifecycle management as automated content. It is available from the following locations:

- On the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm/pe/ui/`
- Using the *Configuration Services (Process Engine)* tile in SAP HANA Application Lifecycle Management XS user interface
- Using the *Manage Configuration Services* tile in SAP HANA cockpit

Related Information

[Tutorial: Execute a Configuration Service with Process Engine \[page 139\]](#)

[Process Engine Roles \[page 138\]](#)

[Troubleshooting \[page 142\]](#)

11.1 Process Engine Roles

To grant users the privileges they require to perform tasks with the Process Engine, you must assign them the relevant Process Engine roles.

The following table lists the roles that are available for tasks related to the Process Engine. The roles are hierarchical and interlinked. The `sap.hana.xs.lm.roles::Administrator` role is the *Administrator* role of SAP HANA application lifecycle management and grants the privileges of all other Process Engine-related roles as well as application lifecycle management roles. For more information, see *SAP HANA Application Lifecycle Management Roles* in the *SAP HANA Application Lifecycle Management Guide*.

➔ Recommendation

Do not use the repository roles delivered with SAP HANA directly, but instead use them as templates for creating your own roles. Furthermore, if repository package privileges are granted by a role, we recommend

that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

Roles available for the Process Engine

Role	Description
<code>sap.hana.xs.lm.pe.roles::PE_Display</code>	The user can monitor processes and display services.
<code>sap.hana.xs.lm.pe.roles::PE_Execute</code>	In addition to the previous role, the user can start, stop, skip, and resume processes.
<code>sap.hana.xs.lm.pe.roles::PE_Activate</code>	In addition to the previous roles, the user can activate services from repository files.
<code>sap.hana.xs.lm.roles::Administrator</code>	The user can install products. This role includes all previous roles.

Related Information

[SAP HANA Application Lifecycle Management Roles \[page 14\]](#)

11.2 Tutorial: Execute a Configuration Service with Process Engine

In this tutorial, you use the demo content delivered with the Process Engine to execute a configuration service.

Prerequisites

- An SAP HANA system is available.
- SAP HANA XS is up and running on the SAP HANA system.
- Depending on the task you want to perform with the Process Engine, you must have the privileges based on a role granted by one of the Process Engine role templates described in *Process Engine Roles*. The link to the topic is in the *Related Information* section. The privileges of the `sap.hana.xs.lm.pe.roles::PE_Activation` role allows you to perform all Process Engine tasks.

Context

The Process Engine uses different terms for identifying design time or runtime artifacts. The *service* is the core entity at design time. It has multiple attributes describing its purpose and steps representing the executable entities. They perform the actual work during execution. An executable can be a JavaScript function in an XS JavaScript library or an SQL stored procedure. When starting a service, the Process Engine creates a *process* based on a service. It copies all steps associated with the service as tasks, and it copies the parameters of the selected variant to the parameters of the process. Furthermore, the Process Engine associates a *status* with the process.

You execute the following steps to configure the demo service:

- Activate the demo service.
Services are delivered as repository objects. The services required by the administrator need to be enabled once before use. This activity is called *activation*.
- Prepare the demo service parameters.
The demo service needs parameters during execution. The set of required parameters is stored under a common key, the *variant*. Before you can start a service you need to prepare variants. Since you are about to start the service for the first time, you do not have any variants prepared. If you repeat an execution, you can use an existing variant. For the demo service, you enter *user* and *password*. Since this is a demo example, the user does not need to exist and the password can be any set of characters.
- Start the demo service.
The demo service consists of the following steps:
 - JS_APPVAR by JavaScript
This step executes a JavaScript function that shows how to consume and return parameters in JavaScript.
 - SQL_APPVAR by SQL Script
This step executes a SQL script function that shows how to consume and return parameters in SQL script.

Note

The demo content does not perform any configuration of the system. It only writes messages into the log of the Process Engine. It provides you with a hands-on experience for using the Process Engine.

Procedure

1. Open SAP HANA Application Lifecycle Management.

SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`

2. Choose *Configuration Services (Process Engine)*.

The process engine opens in a new browser window or a tab.

3. Select *Services* on the left-hand side of the screen.

A list of services available for configuration appears. Active services are indicated by a green status icon, inactive services have a grey status icon. Inactive services must be activated before they can be started.

You find the demo service *DEMO_VarCont* as inactive in the list.

4. Select *DEMO_VarCont* and choose *Activate*.

Note that the activation process can take some time. After the service was activated successfully, you can start it.

If the activation were not successful, you can find the error messages in a detailed log.

5. To display the service after the activation, choose *Go To Service*.

The details of the active service *Demo Service with Process Engine Variant Container* appear.

6. To prepare the parameters for the demo service, choose *Maintain Variants*.

A new screen for variant maintenance appears.

7. Enter a *user* name and a *password* as parameters and choose *Save As*.
8. Enter a *variant ID* and, optionally, a *description*, and then choose *Create*.
9. Return to the previous screen to view the variant that you just created.

If you open the *Steps* tab on screen, two executable steps are displayed.

10. To start the service, select your variant and choose *Start Variant*.

The *Process* tab opens and a new process appears at the top of the list.

11. Select the process to navigate to the process details.

A list of tasks appears.

12. Choose *Refresh* to observe the progress of the process.

The overall process status is displayed on top of the progress bar. It is a cumulation of the statuses of the individual tasks.

The status icons allow you to intervene in the process execution if errors occur. You can click on an icon to display the task log.

13. When the process completion reaches 100%, choose the *Parameters* tab.

You see an overview on the scalar parameters and their changes during execution.

Entries for the step *NA* show the parameter values after the variant container is copied and before the process execution starts. The other entries show the values after the step was executed.

14. To view the log of the *Consuming and returning parameters via SQL* task, select this task at the bottom of the screen.
 - a. Search for a message with a green status that starts with *JavaScript function sends*. At the end of the message, you see the parameter value of your **user**.
 - b. Search for a message with a green or orange status that contains the text ... *password received*.

The step compares the received value of the parameter you entered as **password** with a value set by the demo service. If you entered the password as set by the demo service, the Process Engine issues the message *Correct password received*. If you entered a different password, the Process Engine writes *Incorrect password received* in the log.

Results

You have used the demo configuration service of the Process Engine. You have activated the demo service, prepared the parameters, and executed the service. Afterward, you have checked the logs of the Process Engine.

Related Information

[Process Engine Roles \[page 138\]](#)

[Troubleshooting \[page 142\]](#)

11.3 Troubleshooting

If a process stops with errors, you should first analyze the logs to find out why an error occurred. Afterward, you have various options to respond to the error situation.

The Process Engine provides a process log and a task log. If a single task has an error you can start with the task log to analyze if an error message is due to a specific step. If this does not help, you can open the process log and search or filter for error messages.

- **Process Log**

This is a collection of all task logs and additional entries related to the process. This log contains all messages with technical details, including the log of the internal activities of the Process Engine. You can find this log when you open the [Log](#) tab in the single process view.

- **Task Log**

This is the log of the execution of a single task. Messages with technical details about the Process Engine usually are not displayed here. You get this log when you navigate to a task view by clicking on a task in the single process view.

You have the following options to respond to an error:

- If the error is only temporary or you solved the error already, you can execute the step again by choosing [Resume](#).
- You can decide to perform the task manually and skip the execution of the task by choosing [Skip](#).
- You can cancel the current process and start a new one. To do this, choose [Cancel](#).
- If you cannot resolve the error, and you need to contact SAP, open an incident and assign it to the support component of the application that provides the configuration content or, alternatively, to component HAN-LM-APP. Make sure that you attach the diagnosis information that you can download for each process using the link on the [Diagnosis Information](#) tab.

12 FAQs on SAP HANA Application Lifecycle Management

Here you can find answers to frequently asked questions in SAP HANA application lifecycle management.

General Questions

Can I merge the content of the same DU from two source systems?

SAP does **not** recommend that you merge the content of the same DU from two systems. SAP HANA does not provide any functions to determine if conflicts exist and to resolve them automatically.

When should I use SAP HANA native transport and when should I use CTS transport?

In short, you use native SAP HANA transport to transport only SAP HANA objects, especially if you want to use SAP HANA transports out-of-the-box with little configuration effort. You use CTS transport to transport SAP HANA content if you already use CTS for other transports.

For more information on when to use which mode, see [Transport Scenarios in SAP HANA Application Lifecycle Management \[page 22\]](#).

Transport of SAP HANA Content: When should I transport DUs, when should I transport products?

If you only want to transport the content of one or multiple independent DUs, you can use DU transport. Doing this means that you save organizational overhead because you do not have to enter all the metadata that is required on product level.

If there are complex dependencies between DUs, or if multiple DUs are to be distributed as a whole, you can model and transport products. You can maintain metadata for products and thus transport a well-defined state. Bear in mind that only native SAP HANA transport supports the transport of products currently. If you use CTS transport, you **cannot** transport products. You can assemble products and install them using the `hdbalm` commandline tool, however.

If I find an error in the test system, can I correct the error there?

You should always correct errors in the development system and transport the corrections through your development landscape. Doing this ensures that all systems in your landscape are consistent.

What do I have to consider when I need to change the CTS upload system?

If change recording is enabled in your system, changing the CTS upload system may cause inconsistencies in your system landscape. For more information, see [Change CTS Configuration \[page 40\]](#).

How can I import language DUs?

You can import language DUs using the `hdbalm` command line tool. For more information, see [hdbalm import Command \[page 128\]](#).

What can I do when I encounter authentication issues when registering a remote SAP HANA source system in SAP HANA Application Lifecycle Management?

When you encounter an authentication issue when registering a remote SAP HANA system in SAP HANA Application Lifecycle Management, you can use the `/sap/hana/xs/lm/core/TestHttpDestination.xsjs` service to check that the HTTP destination was configured correctly. For more information, see SAP Note [2561270](#).

Questions on Change Recording

How can I perform a top-down approach in development when empty packages cannot be transported when change recording is enabled? What needs to be considered in the bottom-up approach?

In a top-down development approach, the project responsible provides and models the development infrastructure before development starts their work. That is, the responsible creates containers for development, such as products, product instances, or delivery units, and assigns packages, provides users, and so on. This preparatory work also includes configuring the transport infrastructure, and in this process, it might involve simulating a transport. However, when change recording is enabled, you cannot transport empty packages. Transport is only possible if there is at least one package that contains at least one object.

If you want to realize the top-down approach, you can create a test application with initial content. For example, SAP HANA Web-based Development Workbench includes a wizard for creating applications and automatically generates the application-descriptors that are required for the SAP HANA application (`.xsapp` and `.xsaccess` files). You can then test that the application can be transported through the landscape.

In a bottom-up approach, development starts developing in packages without creating any metadata. These development artifacts can be activated and deployed. However, they cannot be transported since the transport tools do not support the transport of packages without metadata. As a prerequisite for the transport, a container must exist, such as a delivery unit or a product, to which the packages that you want to transport are assigned.

I forgot to assign a package to my DU and I changed objects that are recorded and released in a changelist. They were not transported as they are not assigned to any DU. How can I fix this?

You can assign the package to your DU. The changelist will be offered for transport again.

Caution

If you transport this changelist again, all the other objects in the DU that are part of this changelist will be imported again. In addition, possibly other changelists have to be retransported because they also contain objects in the same DU.

See also the remarks on the effects of reassigning packages to DUs: [Consequences of Package - DU Reassignments \[page 66\]](#).

How can I transport deleted SAP HANA repository packages when change recording is enabled?

When change recording is enabled in a system, the deletion of a package is not recorded as a change, since change recording only records changes to objects. You cannot transport a deleted package in a changelist

when you use transport mode [Selected Changelists](#) (native SAP HANA transport) or [Released Changelists](#) (CTS transport). To bring the deletion of a package to the target system, you must transport [All Changelists](#) in the DU (native SAP HANA transport) or [Delivery Units](#) (CTS transport) from which the package was deleted.

See also the remarks on the effects of reassigning packages to DUs: [Consequences of Package - DU Reassignments \[page 66\]](#).

When am I supposed to turn change recording on or off, only once or multiple times? What will be the impact when I temporarily turn it off, for example, if there are deletion issues?

You should **switch on change recording only once**, ideally before you start your development. If you switch change recording off, all open changelists will be closed and not released. If you do this, you must make sure that you release and transport all open changelists beforehand. If you enable change recording again, the base changelist is created, which includes all active objects of DUs. This invisible changelist will be transported together with the first released changelist. You must therefore make sure that all active objects of the DU that exist in the source system are supposed to reach the target system. For more information, see [Technical Details of Initial Change Recording Setup \[page 64\]](#).

I have transported a changelist with only a few objects, but now there are lots of objects in my target system. What happened?

An important concept of SAP HANA application lifecycle management is that it tries to **keep the target system consistent** with respect to the changes that were performed and released in the **source system**. Therefore, a transport of individual changelists does not always include only the released changelists of a DU. It is possible that other objects are also part of the transport, of which you might not be aware. For example, the following objects can be included in a transport:

- Objects contained in predecessor changelists of the selected changelist. If the system finds predecessor changelists, it does not allow you to transport the selected changelist without transporting the predecessors as well.
For more information, see [Predecessor Changelists in SAP HANA Change Recording \[page 65\]](#).
- Objects contained in the base changelist if this is the first transport of a changelist after change recording was enabled.
If you enabled change recording after you started development, the base changelist contains all the active objects that exist for the DU at the point in time when change recording is enabled. It is created and transported together with the first transport of changelists for the DU.
For more information, see [Technical Details of Initial Change Recording Setup \[page 64\]](#).
- SAP HANA packages
Packages are not normal transportable objects. They cannot be activated and they are always transported as metadata together with the DU. Whenever a DU archive is transported, the information about its packages is also transported.
Therefore, if you delete a package and transport the changelist with the deletion individually to the target system, the empty package will still be visible the target system. The deletion will only be effective when you transport the complete delivery unit.
For more information, see *How can I transport deleted SAP HANA repository packages when change recording is activated?*

When can I use changelists spanning multiple DUs? What is the impact on transport and what do I need to keep in mind ? What is recommended?

In general, SAP recommends that you assign objects of only one DU to a changelist. If for some reason this is not possible and there are objects of more than one DU in the same changelist, then SAP recommends that you configure the transport route for the same DUs. If you transport the changelist using a transport route that is

not configured for all DUs, only the objects in the assigned DUs will be transported. The objects in the other DU will not be transported.


For more information, see also [Recommendations on the Transport of Changelists \[page 72\]](#).

When I use change recording, DUs delivered by SAP are also recorded during import in customer systems. What does this mean for upgrades, updates, and patches of SAP software?

If upgrades, updates, or patches of SAP HANA content are imported in a system in which change recording is enabled, this is also recorded in changelists. If the import finishes without errors, the changelists are immediately released. In order not to transport these changelists to the follow-on systems, you must ensure that the SAP DU is not included in any transport route (if you use native SAP HANA transport) or that the SAP DU is not assigned to CTS (if you use CTS transport). This way, you ensure that changelists with objects of DUs delivered by SAP are not offered for transport.

If the SAP DUs were imported with other tools, SAP HANA Studio, for example, and the import failed, then it is possible that an open changelist exists in the system. If the import of the same DU is retried, the open changelist locks the DU, and the import cannot proceed. In this case, you must first release the changelist and then you can retry the import.

After renaming the system ID (SID) of my source system, changelists are displayed as available for transport even though they were already transported. What can I do?

If you have renamed the SID of a source system, for example, by a system copy, or migration of a single-container system to a multiple-container system, and you have changelists available for transport that were already transported, you can fix the issue by updating the history information about transported changelists using a script. For more information, see SAP Note [2560918](#) .

13 SAP HANA Repository Translation Tool

The Repository Translation Tool (RTT) is a Java-based command line tool shipped with the SAP HANA client that enables you to transport language files in a standard format between the SAP HANA repository and a file system or between the SAP HANA repository and a dedicated SAP translation system.

During the translation process, the inactive (design-time) content of tables in the SAP HANA repository must be uploaded to the translation system using the repository translation tool (RTT). After translation is completed, you use the repository translation tool to re-import the translated texts into the SAP HANA repository.

Use the following RTT commands to implement the translation process:

Command	Abbreviation	Definition
download (SAP Internal Only)	d	Downloads the translated texts from the SAP translation system to a file system.
export		Exports the texts in the original language (written by the developer) from the SAP HANA repository text tables to a file system.
export/upload (SAP Internal Only)	e	Exports and uploads the texts in the original language (written by the developer) from the SAP HANA repository text tables to a file system.
import		Imports the translated texts from a file system to the SAP HANA repository text tables.
download/import (SAP Internal Only)	i	Downloads and imports the translated texts from a file system to the SAP HANA repository text tables.
upload (SAP Internal Only)	u	Uploads the texts from the file system to the SAP translation system where the translators can translate the texts from the original language into the required target languages.

If you are using the SAP translation system, you can combine operations in one command, as follows:

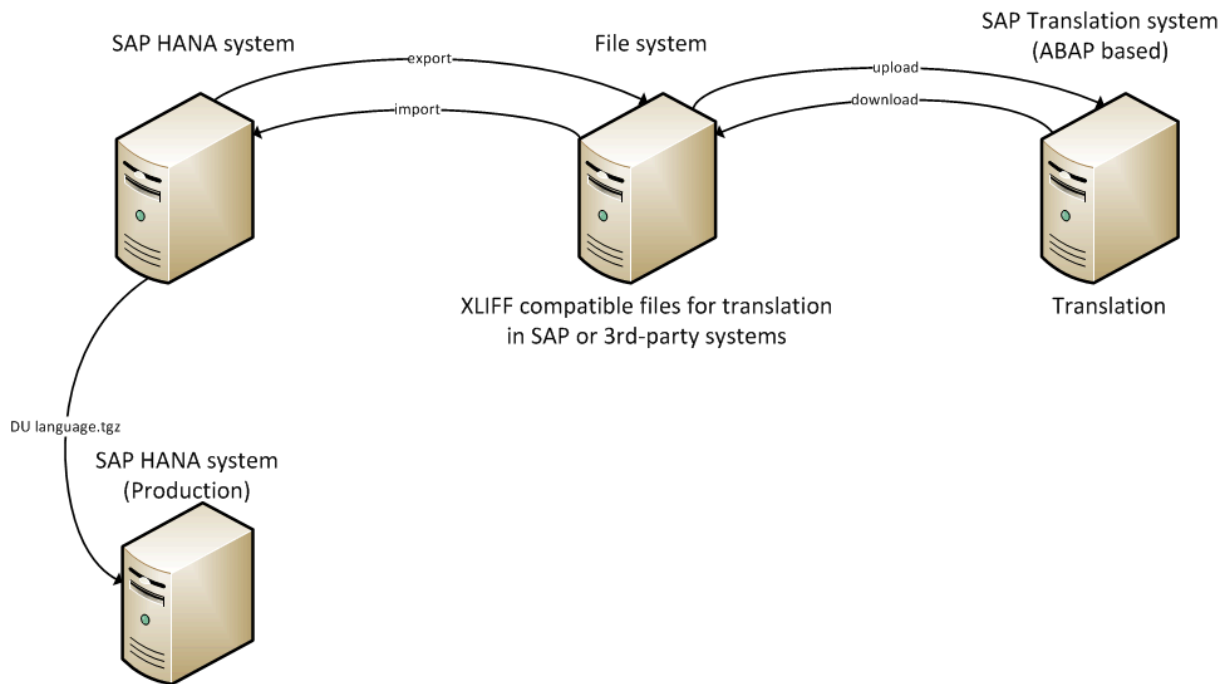
- **export/upload**
(SAP Internal Only): Exports the texts in the original language and uploads the texts to the SAP translation system.
- **download/import**
(SAP Internal Only): Downloads the translated texts from the SAP translation system and imports the translated texts to the SAP HANA repository text tables.

The following graphic depicts the translation tool process using the commands provided by the RTT.

i Note

This tool is designed for use with one SAP HANA system and one translation system.

Repository Translation Tool Process



Related Information

[SAP HANA Repository Translation Tool \(RTT\) Parameters \[page 148\]](#)

[Create Text-Strings Packages for Translation \[page 151\]](#)

[Export Text-Strings Files for Translation \[page 152\]](#)

[Import Translated Text-Strings Files \[page 152\]](#)

13.1 SAP HANA Repository Translation Tool (RTT) Parameters

The SAP HANA Repository Translation Tool (RTT) is a Java-based command line tool that exports language files in a standard format for translation for customer or partner use.

The following parameters can be used with the commands.

Parameter	Abbreviation	Definition
config	c	Configuration file (default: C:\Users\<your_user_ID>\workspace-java-indigo\rtt.properties)
dbPasswd		Database password (overrides config file) (default: no password)
dbReadFromStdin		Read database password from stdin (overrides config file and --dbpasswd) (default: false)
deliveryUnit	d du	Delivery units (format: <vendor>.<deliveryunit>) (default: no delivery units)
force		Force import of translated texts (skip source text matching) (default: false)
languageChange		Allow original language change (default: false)
locale	l	Locales to download/import (default: all locales)
noDelete		Do not delete XLIFF files before export/download (default: false)
noExcludePrivate		Do not exclude objects marked as private (= "fncViewLayer" tag set to "Private") (default: false)
package	p	Packages to export/upload/download/import (default: no packages)
r3ReadFromStdin		Read R/3 translation system password from stdin (overrides config file and --r3passwd) (default: false)
r3Passwd		Database password (overrides config file) (default: no password)
severity		Log severity level (all, debug, error, fatal, info, none, path or warning) (default: WARNING)
skipReview		Specifies that the review step is to be skipped. (default: false) <div> i Note Setting the parameter to true will overwrite already reviewed texts. </div>
verbose	v	Verbose mode (show messages with severity INFO) (default: false)
veryVerbose	vv	Very verbose mode (show all messages with severity DEBUG) (default: false)
xliffDir	x	XLIFF file directory (default: "rtt_exports" or "rtt_imports" in "C:\Users\<your_user_ID>\AppData\Local\Temp")

Here are some examples of RTT syntax:

- Export the texts from those packages matching "pack*" from the database using the default configuration file ("rtt.properties"):


```
rtt --export -p pack*
```
- Import the translated texts into the database using the default configuration file ("rtt.properties"):


```
rtt --import -p pack*
```
- Export the texts from the database into the directory "exports":


```
rtt --export -p pack* -x exports
```
- Import the translated texts from the directory "imports":

```
rtt --import -p pack* -x imports
```

13.2 Configure the Repository Translation Tool

The repository translation tool (RTT) reads a configuration file (`rtt.properties`) to determine the settings for file transfer.

Context

You need to maintain your information in the `rtt.properties` file in order to use the RTT. The `rtt.properties` file enables you to specify the settings required to transfer text-resource files (in the required XLIFF format) between the system hosting the SAP HANA repository and the system hosting the translation database and tools. You can use `rtt.properties` file to set system-related access details, for example, system user-logon IDs, and the translation-area number (TAN) for the translation system.

Procedure

1. Locate the `hdbclient` directory on your server's hard drive.
2. Open the `rtt.properties` file and add all required information by replacing the placeholders.

```
# db settings (SAP HANA repository)
db.hostname=<db hostname, e.g. myhost.name.com>
db.instance=<db instance, e.g. 00>
db.user=<db username>
db.passwd=<db password>
# translation system settings
jco.client.tan=<translation area number e.g. 027001>
jco.client.client=<translation system client, e.g. 000>
jco.client.user=<translation system user>
jco.client.passwd=<translation system password>
jco.client.mshost=<translation system host>
jco.client.r3name=<translation system SID>
```

i Note

The RTT properties file is intended to help automate the connection between SAP HANA and the translation system, which are installed and licensed separately. The connection details for a specific translation system will need to be obtained from the system's administrator.

The translation area number required in `jco.client.tan` represents a technical area in a specific language and is typically of the form "027001"; the "translation-system user" specified in `jco.client.user` is used to establish a connection to the translation system and must have the permissions required to log on (and upload content) to the translation system. The name of the R3 system defined in `jco.client.r3name` is the typical three-character-long system ID, for example, "B1Y".

3. Save your work.

i Note

The `rtt.properties` file is not encrypted, which means that information you include in the configuration potentially is exposed.

13.3 Create Text-Strings Packages for Translation

You must create packages to transport text strings for translation.

Context

All text strings must be stored in a `.hdbtextbundle` file. The `.hdbtextbundle` file can be retrieved using a JavaScript API. The following example demonstrates how you can externalize the texts in a `.hdbtextbundle` file.

```
# TRANSLATE
# XBUT,20
BUTTON_SAVE=Save
# XBUT,20
BUTTON_CANCEL=Don't save
# XMSG,40
MSG_SUCCESS=File has been saved.
```

In this example, `# TRANSLATE` defines that the texts can be translated (exported), and `# XBUT,20` defines the text type with a 20-character length maximum.

Procedure

1. In the SAP HANA studio, select the system from which you want to transport text files for translation.
2. Click the *Content* file with the alternate mouse button and choose **New** **Package**.
3. In the *New Package* dialog box, enter all relevant information and then choose *Translation*.
4. Select a *Terminology Domain*, enter a *Text Collection* name, and choose *OK*.
5. Choose *Save and Activate*.

13.4 Export Text-Strings Files for Translation

You want to export text-strings files to a file system for translation.

Prerequisites

You have updated the `rtt.properties` file with all appropriate information.

Context

When you have created your text-strings files and assigned them to packages, you can begin exporting them to a file system for translation.

Procedure

1. Start a command line application and navigate to the directory in which the `<rtt.properties>` file is stored.
2. Type `rtt -e -p <name of the package> -v <other parameters>` and press .

Results

The `Upload finished` message appears. The file has been exported to the file server and is ready to be sent for translation.

13.5 Import Translated Text-Strings Files

You want to import translated text-strings files from a file system.

Context

When your text strings are translated, you can import them back into your system.

Procedure

1. Start a command line application and navigate to the directory in which the `<rtt.properties>` file is stored.
2. Type `rtt -i -p <name of the package> -v <other parameters>` and press .

Results

The `Download finished` message appears. The file has been imported from the file server.

14 Maintaining Translation Text Strings

Maintain the translated text strings used in an application's user interface, error messages, and documentation.

For the purposes of localisation (L10N), you can provide the text strings displayed in an application's user interface in multiple languages, for example, English, French, or Chinese. You can also provide notifications and error messages in the same, local languages. To manage and maintain these translated text strings, SAP HANA provides an online translation tool (OTT). The translation of the text strings themselves can be performed manually or with suggestions provided by an external service, for example, SAP Translation Hub. Access to external translation services is not covered by the SAP HANA license and usually requires a user account.

Setting up and maintaining the online translation tools for SAP HANA includes the following high-level tasks:

- Enabling the translation tool
- Accessing packages in the SAP HANA repository
- Maintaining text strings in the source and target languages
This task involves maintaining the contents of the following SAP HANA tables:
 - ACTIVE_CONTENT_TEXT
 - ACTIVE_CONTEXT_TEXT_CONTENT
 - ACTIVE_OBJECT_TEXT
 - ACTIVE_OBJECT_TEXT_CONTENT
- Enabling access to a remote text-translation service (**optional**)

Restriction

Access to external translation services is not granted in the SAP HANA license. To use external translation services such as the *SAP Translation Hub*, an additional license is required. In addition, the *SAP Translation Hub* is currently available only for Beta testing.

- Maintaining HTTP destinations for any remote systems that provide services used by the *Online Translation Tool* (**optional**)
Remote translation services such as *SAP Translation Hub* can provide access to a database of translated text strings, which are used to provide suggestions in the target language. To access such a remote service, you must maintain an HTTP destination (or extend an existing destination) that provides details of the host system where the translation service is running as well as a valid user account and logon authentication. You must also ensure that a trust relationship exists between the translation server and SAP HANA, for example, by importing the translation server's client certificate into the SAP HANA trust store.

The SAP HANA *Online Translation Tool* is available on the SAP HANA XS Web server at the following URL:

`http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/translationTool/`

Tip

The privileges required to use the SAP HANA *Online Translation Tool* (OTT) are granted by the role `templatesap.hana.xs.translationTool.roles::translator`.

Related Information

[Create and Edit Text Translations \[page 155\]](#)

[Export and Import Translated Text \[page 159\]](#)

[SAP Translation Hub Cloud Service \(beta\)](#)

14.1 Create and Edit Text Translations

Maintain translations for text strings displayed in an SAP HANA application's user interface.

Prerequisites

To maintain translated text for an application in SAP HANA XS, the following prerequisites apply:

- You have access to an SAP HANA system.
- You have the privileges required to access the repository packages containing the text strings to be localized/translated.
- You have a role based on the role template `sap.hana.xs.translationTool.roles::translator`.
- If you want to make use of optional external translation services, you must maintain access to the translation server system.

Restriction

Access to external translation services is not granted in the SAP HANA license. To use external translation services such as the *SAP Translation Hub*, an additional license is required. The *SAP Translation Hub* is currently available only for BETA testing.

Details of the remote systems where the translation service is running (for example, SAP Translation Hub) are defined in HTTP destination configuration files along with details of any corresponding user account and authentication certificates.

Context

An application's user interface and notifications can be translated from the original source language (for example, English) into one or more local (target) languages, for example, French, Spanish, or Japanese. You

can either translate the texts manually or with the help of an (optional) external translation service. To provide translations of the UI text strings for your SAP HANA application, perform the following steps:

Procedure

1. Start the *SAP HANA Online Translation Tool*.

The *SAP HANA Online Translation Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/translationTool`.

Note

In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must also have the privileges required to perform the tasks associated with the maintenance of translation texts.

2. Select the delivery unit that contains the application with the text strings you want to translate.

Use the *Delivery Unit* drop-down list to select a delivery unit.

Tip

The name of the vendor associated with the selected delivery unit is displayed automatically in the *Vendor* field, for example, *acme.com*; the vendor name cannot be changed here.

3. Select the package that contains the text strings you want to translate.

Use the *Package* drop-down list to select a package. If the selected package contains text elements, they are displayed alphabetically in a list.

Tip

The original source language associated with the contents of the selected package is displayed automatically.

4. Enable access to a text-translation service, for example, *SAP Translation Hub*. (**optional**).

Restriction

Access to external translation services is not granted in the SAP HANA license. To use external translation services, an additional license is required.

If you want to make use of the services provided by a translation server, you need to maintain an HTTP destination **extension** that provide details of the host system where the translation service is running; access to the translation service usually requires a user account and logon authentication. You must also ensure that a trust relationship exists between the translation server and SAP HANA, for example, by importing the translation server's client certificate into the SAP HANA trust store that you are using to handle authentication for this HTTP destination.

The HTTP destination configuration

`sap.hana.xs.translationTool.server:translationService.xshttpdest` defines details of the

server hosting the SAP Translation Hub service. Although you cannot edit this destination configuration, note that you can use an HTTP destination **extension** to change the details, for example, to point to an alternative host name.

5. Add a translation for a text element.

For a given text element in the *Text ID* list, you can provide a suitable translation in one or more languages, for example: French (*fr*), Spanish (*es*), and Japanese (*ja*).

a. Expand the desired UI text element.

In the *Text ID* list, locate and expand the element for which you want to provide a translation.

b. Add a translation.

Choose *Add Translation*.

c. Select the desired language for the translation from the *Target Language* drop-down list.

d. In the *Target Language Text* box, type the translation for the selected text element.

➔ Tip

If the *SAP Translation Hub* option is enabled, language-specific suggestions for possible translation matches are provided as you type. If you see a suggestion that is suitable, use the mouse to select the suggested text.

e. Add another translation.

Choose *Add Translation*

f. Edit an existing translation

Choose the *Edit* icon next to the translation you want to modify and make the required changes.

6. Save your additions and changes.

Choose *Save* to store the added translations or any modifications in the appropriate tables in the SAP HANA database.

Related Information

[Online Translation Tool Details \[page 157\]](#)

[Export and Import Translated Text \[page 159\]](#)

14.1.1 Online Translation Tool Details


Display details of the source text for an application's user interface elements and, if available, any available translations.

The *Online Translation Tool* tool enables you to view details of the text elements contained in the individual packages of an SAP HANA application. The following table indicates which information can be viewed.

Note

The privileges required to use the SAP HANA [Online Translation Tool](#) (OTT) are granted by the role template `sap.hana.xs.ott.roles::translator`.

Translation Text Details

UI Element	Description	Example
Delivery Unit	Name of the SAP HANA delivery unit (DU) that contains the default text strings for which a translation is required along with the name of the vendor associated with the selected delivery unit	ACME_XS_BASE - acme.com
Package	The name of (and path to) the package containing the text strings for which a translation is required	acme.com.app.ui.login
Source language	Short name of the source language for the text strings contained in the selected package, for example: en (English), fr (French), ja, (Japanese)	en
Target Language	Long or short name of the target language for the text strings contained in the selected package, for example: Bulgarian (bg), French (fr), Japanese (ja)	Chinese (zh)
Domains	The SAP product-specific translation domain to which the selected DU/package belongs, for example, <i>Financial Accounting</i> or <i>Customer Relationship Managment</i> . Domains are used in the translation process to determine the correct terminology for a text string that has to be translated; the same text might require a different translation depending on the domain (or application) in which it is used. Suggestions from a remote translation service such as the SAP Translation Hub are restricted to the currently selected domain.	"Basis", or "Accounting - General"
Enable Translation Hub	<p>Enable automatic suggestions (in the Target language text box) for translation texts using a remote service such as SAP Translation Hub; the suggestions are provided by a remote translation database.</p> <div> Restriction Access to external translation services is not granted in the SAP HANA license. To use external translation services such as the <i>SAP Translation Hub</i>, an additional license is required. The <i>SAP Translation Hub</i> is currently available only for BETA testing.</div> <p>Access to the remote translation service usually requires a user account and logon authentication. You also need to maintain an HTTP destination (or extend an existing one) for the translation server system and ensure the server system is trusted by SAP HANA, for example, by importing the translation server's client certificate into the SAP HANA trust store.</p>	Yes/No

UI Element	Description	Example
<i>Text ID</i>	The name/ID of the UI element for which a text string is required. This could be a tab title, a box name, a notification, or an error message.	LOGON_LABEL
<i>Default Text</i>	The text string associated with the text ID	HANA Logon
<i>Target Language Text</i>	Proposed/accepted translation (in the target language) of the text string displayed (in the source language) in the <i>Default Text</i> field. Activate the <i>Enable Translation Hub</i> option to enable auto-suggestions in the target language.	-
<i>Source Object</i>	The name of the design-time artifact that contains the UI text strings.	logonForm.hdbtextbundle

Related Information

[Create and Edit Text Translations \[page 155\]](#)

[Export and Import Translated Text \[page 159\]](#)

14.2 Export and Import Translated Text

Transport text translations between systems using the industry-standard, XML-based `xliff` format.

Prerequisites

To export and import translated text for an application in SAP HANA XS, the following prerequisites apply:

- You have access to an SAP HANA system.
- You have access to the repository packages containing the text strings to be localized/translated.
- You have been granted a role based on the role template
`sap.hana.xs.translationTool.roles::translator`.

Context

An application's user interface and notifications can be translated from the original source language (for example, English) into one or more target local languages, for example, French, Spanish, or Japanese. To provide translations of the UI text strings for your SAP HANA application, perform the following steps:

Procedure

1. Start the *SAP HANA Online Translation Tool*.

The *SAP HANA Online Translation Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/translationTool`.

Note

In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must also have the privileges required to perform the tasks associated with the maintenance of translation texts.

2. Select the delivery unit that contains the application with the text strings you want to translate.

Use the *Delivery Unit* drop-down list to select a delivery unit.

Tip

The name of the vendor associated with the selected delivery unit is displayed automatically in the *Vendor* field, for example, *acme.com*. You cannot change this here.

3. Select the package that contains the text strings you want to translate.

Use the *Package* drop-down list to select a package. If the selected package contains text elements, they are displayed automatically in an alphabetically ordered list.

Tip

The original source language associated with the contents of the selected package is displayed automatically.

4. Export the UI text elements from the local source system.

You can export the translation texts to an archive on a local file system using the industry-standard, XML-based `xliff` format.

5. Import the UI text elements to the remote target system.

You can import the translation texts into SAP HANA from an archive whose content are stored using the industry-standard, XML-based `xliff` format.

6. Confirm that the import operation was successful.

Check the status of the following tables in the SAP HANA database:

- `ACTIVE_CONTENT_TEXT`

-
- ACTIVE_CONTEXT_TEXT_CONTENT
 - ACTIVE_OBJECT_TEXT
 - ACTIVE_OBJECT_TEXT_CONTENT

Related Information

[Online Translation Tool Details \[page 157\]](#)

[Create and Edit Text Translations \[page 155\]](#)

Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities

SAP HANA server software and tools can be used for several SAP HANA platform and options scenarios as well as the respective capabilities used in these scenarios. The availability of these is based on the available SAP HANA licenses and the SAP HANA landscape, including the type and version of the back-end systems the SAP HANA administration and development tools are connected to. There are several types of licenses available for SAP HANA. Depending on your SAP HANA installation license type, some of the features and tools described in the SAP HANA platform documentation may only be available in the SAP HANA options and capabilities, which may be released independently of an SAP HANA Platform Support Package Stack (SPS). Although various features included in SAP HANA options and capabilities are cited in the SAP HANA platform documentation, each SAP HANA edition governs the options and capabilities available. Based on this, customers do not necessarily have the right to use features included in SAP HANA options and capabilities. For customers to whom these license restrictions apply, the use of features included in SAP HANA options and capabilities in a production system requires purchasing the corresponding software license(s) from SAP. The documentation for the SAP HANA options is available in SAP Help Portal. If you have additional questions about what your particular license provides, or wish to discuss licensing features available in SAP HANA options, please contact your SAP account team representative.

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: <https://help.sap.com/viewer/disclaimer>).



**go.sap.com/registration/
contact.html**

© 2018 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.
Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.
Please see <https://www.sap.com/corporate/en/legal/copyright.html> for additional trademark information and notices.