



PUBLIC

SAP HANA Platform 2.0 SPS 03

Document Version: 1.1 – 2018-10-31

SAP HANA Security Guide

Content

- 1 **SAP HANA Security Guide.** **7****
- 2 **Security Information by Guide.** **9****
- 3 **SAP HANA Security Patches.** **13****
- 4 **SAP HANA Overview.** **15****
- 4.1 The SAP HANA Database. 15
- 4.2 SAP HANA XS and Development Infrastructure. 16
 - SAP HANA XS, Advanced Model. 16
 - SAP HANA XS, Classic Model. 17
- 4.3 Technical System Landscape. 18
 - Overview of SAP HANA Security Functions. 20
 - Database Isolation. 24
 - Security Considerations After Updating a Single-Container System. 26
- 4.4 SAP HANA Implementation Scenarios. 27
 - SAP HANA as a Data Mart. 27
 - SAP HANA in a Classic 3-tier Architecture. 29
 - SAP HANA as Technical Infrastructure for Native Application Development. 31
- 5 **SAP HANA Network and Communication Security.** **35****
- 5.1 Communication Channels. 35
- 5.2 Network Security. 37
- 5.3 Securing Data Communication. 40
 - Secure Communication Between SAP HANA and JDBC/ODBC Clients. 43
 - Secure Communication Between SAP HANA and an LDAP Directory Server. 57
 - Secure Communication Between SAP HANA XS Classic and HTTP Clients. 59
 - Secure Internal Communication. 60
- 6 **SAP HANA User Management.** **70****
- 6.1 User Types. 71
- 6.2 User Groups. 73
 - SQL Statements and Authorization for User Group Administration (Reference). 77
- 6.3 User Administration Tools. 80
- 6.4 Predefined Users. 83
- 6.5 Deactivate the SYSTEM User. 90
- 7 **SAP HANA Authentication and Single Sign-On.** **92****
- 7.1 User Authentication Mechanisms. 93

| | | |
|----------|--|------------|
| 7.2 | SAP HANA Logon Checks. | 96 |
| 7.3 | Password Policy. | 96 |
| | Password Policy Configuration Options. | 98 |
| | Password Blacklist. | 105 |
| | _SYS_PASSWORD_BLACKLIST. | 105 |
| 7.4 | Single Sign-On Integration. | 106 |
| | Single Sign-On Using Kerberos. | 107 |
| | Single Sign-On Using SAML 2.0. | 108 |
| | Single Sign-On Using SAP Logon and Assertion Tickets. | 111 |
| | Single Sign-On Using JSON Web Tokens. | 112 |
| 7.5 | LDAP User Authentication. | 114 |
| 8 | SAP HANA Authorization. | 119 |
| 8.1 | Privileges. | 120 |
| | System Privileges. | 123 |
| | Object Privileges. | 129 |
| | Analytic Privileges. | 135 |
| | Package Privileges. | 155 |
| | Application Privileges. | 156 |
| 8.2 | Database Roles. | 158 |
| | Predefined Database (Catalog) Roles. | 159 |
| | Catalog Roles and Design-Time Roles Compared. | 163 |
| | SAP HANA DI Roles. | 166 |
| | Repository Roles. | 169 |
| 8.3 | Authorization in the Repository of the SAP HANA Database. | 174 |
| | Developer Authorization in the Repository. | 175 |
| | _SYS_REPO Authorization in the Repository. | 176 |
| | Granting and Revoking Privileges on Activated Repository Objects. | 177 |
| 8.4 | Cross-Database Authorization in Tenant Databases. | 179 |
| 8.5 | LDAP Group Authorization. | 180 |
| | LDAP Group Authorization for Existing Users. | 181 |
| 8.6 | Shared Business Authorizations in SAP HANA. | 184 |
| | GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION (SYS). | 187 |
| 8.7 | Data Masking. | 188 |
| | Masks and Authorization in Masked Tables and Views. | 190 |
| | Example: Masking Data Using a Built-In Procedure. | 193 |
| | Example: Masking Data in a View with Structured Privilege Check. | 194 |
| | Example: Masking Data in View Hierarchy with Structured Privilege Check. | 195 |
| 9 | SAP HANA Data Anonymization. | 197 |
| 9.1 | k-anonymity. | 198 |
| 9.2 | Differential Privacy. | 200 |

| | | |
|-----------|--|------------|
| 10 | Data Storage Security in SAP HANA. | 204 |
| 10.1 | Data Security in the File System. | 204 |
| 10.2 | Server-Side Data Encryption Services. | 205 |
| | Encryption Key Management. | 207 |
| | Data and Log Volume Encryption. | 209 |
| | Backup Encryption. | 212 |
| | Internal Application Encryption Service. | 216 |
| | Root Key Backup. | 219 |
| | Secure Stores in the File System (SSFS). | 220 |
| 10.3 | Client-Side Data Security. | 222 |
| | Secure User Store (hdbuserstore). | 223 |
| | Client-Side Data Encryption. | 230 |
| | Protection of Data in SAP HANA Studio Workspaces. | 237 |
| 10.4 | Cryptographic Service Provider. | 238 |
| 11 | Auditing Activity in SAP HANA Systems. | 240 |
| 11.1 | Audit Policies. | 241 |
| | Actions Audited by Default Audit Policy. | 244 |
| 11.2 | Audit Trails. | 246 |
| | Audit Trail Layout for Trail Target CSV and SYSLOG. | 249 |
| | Audit Trail Layout for Trail Target Database Table. | 252 |
| 11.3 | Auditing Configuration and Audit Policy Management. | 254 |
| | System Properties for Configuring Auditing. | 255 |
| 11.4 | Best Practices and Recommendations for Creating Audit Policies. | 258 |
| 12 | Certificate Management in SAP HANA. | 261 |
| 12.1 | In-Database Certificate Management Workflow. | 264 |
| 12.2 | Client Certificates. | 265 |
| 12.3 | Certificate Collections. | 265 |
| 12.4 | SQL Statements and Authorization for In-Database Certificate Management (Reference). | 267 |
| 13 | Data Protection and Privacy in SAP HANA. | 270 |
| 13.1 | Deletion of Personal Data. | 273 |
| 14 | Security Risks of Trace, Dump, and Captured Workload Files. | 274 |
| 15 | Security of Further SAP HANA Components and Capabilities. | 276 |
| 15.1 | Security Aspects of SAP HANA Platform Lifecycle Management. | 277 |
| 15.2 | Security of SAP HANA Content. | 278 |
| 15.3 | Security Aspects of SAP HANA Smart Data Access. | 280 |
| 15.4 | Security Aspects of SAP HANA R Integration. | 281 |
| 15.5 | Security Aspects of SAP HANA Cockpit. | 282 |
| | Data Protection and Privacy in SAP HANA Cockpit. | 284 |

| | | |
|-----------|--|------------|
| 15.6 | Securing the SAP HANA Database Explorer. | 286 |
| | Secure the SAP HANA Database Explorer from Web Socket Attacks. | 287 |
| | Data Protection in SAP HANA Database Explorer. | 288 |
| 15.7 | Security for SAP HANA Replication Technologies. | 290 |
| 16 | Security for SAP HANA Extended Application Services, Advanced Model. | 293 |
| 16.1 | Technical System Landscape of SAP HANA XS Advanced. | 296 |
| | Application Server Components. | 299 |
| | Users and Clients. | 301 |
| 16.2 | User Administration and Authentication in SAP HANA XS Advanced. | 302 |
| | User Management. | 302 |
| | Predefined XS Advanced Users. | 305 |
| | Predefined Database Roles for XS Advanced. | 310 |
| | User Authentication. | 312 |
| | User Administration Tools. | 313 |
| 16.3 | Authorization in SAP HANA XS Advanced. | 313 |
| | Organizations and Spaces. | 314 |
| | Scopes, Attributes, and Role Collections. | 320 |
| | Controller Role Model. | 322 |
| | Authorization Management Tools | 326 |
| 16.4 | Network and Communication Security with SAP HANA XS Advanced. | 328 |
| | Security Areas. | 329 |
| | Public Endpoints. | 330 |
| | Single-Host Scenario. | 331 |
| | Multiple-Host Scenario. | 332 |
| | XS Advanced Certificate Management. | 334 |
| 16.5 | Data Storage Security. | 336 |
| | System Component Storage. | 337 |
| | Application Storage. | 338 |
| 16.6 | Security-Relevant Logging and Tracing. | 338 |
| | Audited Operations. | 339 |
| | Audit Trails. | 339 |
| | Application Auditing. | 340 |
| 16.7 | Data Protection and Privacy in SAP HANA XS Advanced. | 341 |
| | Processing of Personal Data in Platform-Controlled Artifacts | 344 |
| | Processing of Personal Data in Standard XS Advanced Applications and Services. | 347 |
| 16.8 | Security Aspects of SAP Web IDE for SAP HANA. | 348 |
| | User Authorization and Authentication. | 350 |
| | Known Security-Related Issues. | 351 |
| 17 | SAP HANA Security Reference Information. | 353 |
| 17.1 | SQLScript Security Procedures. | 353 |

| | | |
|------|--|-----|
| | IS_VALID_USER_NAME (SYS) | 354 |
| | IS_VALID_PASSWORD (SYS) | 355 |
| | GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION (SYS) | 356 |
| 17.2 | Security Reference for Tenant Databases. | 358 |
| | Restricted Features in Tenant Databases. | 358 |
| | Default Blacklisted System Properties in Tenant Databases. | 361 |
| 17.3 | Components Delivered as SAP HANA Content | 363 |
| | Administration. | 363 |
| | Application Lifecycle Management. | 369 |
| | Runtime Libraries. | 371 |
| | Configuration. | 372 |
| | Supportability and Development. | 373 |
| | User Interface. | 377 |
| | Documentation. | 380 |

1 SAP HANA Security Guide

The SAP HANA Security Guide is the entry point for all information relating to the secure operation and configuration of the on-premise deployment SAP HANA.

i Note

This guide does not cover security-relevant information of some additional capabilities that may be installed in the SAP HANA system, such as SAP HANA accelerator for SAP ASE, SAP HANA remote data sync, or SAP HANA streaming analytics. For more information, see [Security of Further SAP HANA Components and Capabilities \[page 276\]](#).

Why is Security Necessary?

Protecting corporate information is one of the most important topics for you as an SAP HANA customer. You need to meet ever increasing cyber-security challenges, keep your systems secure, and stay on top of the compliance and regulatory requirements of today's digital world. SAP HANA allows you to securely run and operate SAP HANA in a variety of environments and to implement your specific compliance, security, and regulatory requirements.

1.1 Important Critical Configurations

⚠ Caution

SAP HANA has many configuration settings that allow you to customize your system specifically for your implementation scenario and system environment. Some of these settings are specifically important for the security of your system, and misconfiguration could leave your system vulnerable. For this reason, a security checklist of critical configuration settings is available. See *SAP HANA Security Checklists and Recommendations* on SAP Help Portal.

We recommend that you verify your system for critical configurations and latest security patches. Specifically, we recommend verifying that:

- The master keys of the following stores have been changed:
 - The secure store in the file system (SSFS) of the instance
 - The SSFS used by the system public key infrastructure (PKI)
 - The SAP HANA secure user store (`hdbuserstore`) of the SAP HANA client
- Critical privileges are only assigned to trusted users and critical privilege combinations are avoided if possible.
- The network configuration of your SAP HANA system is set up to protect internal SAP HANA communication channels.

- Latest security patches are applied for the SAP HANA system as well as the underlying operating system.

For more information about how to check critical settings and how to find information on recommended settings, see *SAP HANA Security Checklists and Recommendations* on SAP Help Portal.

For more information about keeping your system up to date by installing the latest security patches, see the section on security patches.

Related Information

[SAP HANA Security Patches \[page 13\]](#)

2 Security Information by Guide

Find security-related information by guide

In addition to this *SAP HANA Security Guide*, several other documents in the SAP HANA documentation set provide task- and tool-oriented security information for specific roles and lifecycle phases. Security-related reference documentation is also available.

SAP HANA Security Guide

This guide is the entry point for all information relating to the secure operation and configuration of an on-premise deployment of SAP HANA. Use it to understand security concepts and features of the SAP HANA database and the SAP HANA extended application services, advanced model.

For an overview of the security features of the SAP HANA database, see [Overview of SAP HANA Security Functions \[page 20\]](#).

i Note

Security information for some SAP HANA components, as well as additional capabilities that may be installed in your SAP HANA system is available in other documents. See [Security of Further SAP HANA Components and Capabilities \[page 276\]](#).

SAP HANA Security Checklists and Recommendations

This document contains information and recommendations on critical settings with a security impact. It covers:

- SAP HANA database
- SAP HANA XS advanced

Open [SAP HANA Security Checklists and Recommendations](#)

SAP HANA Administration Guide

This guide is the entry point for all information related to the ongoing operation and maintenance of the SAP HANA platform. It contains information about performing the following security-related administration tasks using the SAP HANA cockpit:

- Monitoring critical security settings
- Creating and provisioning database users
- Configuring auditing and creating audit policies
- Configuring data-at-rest encryption and managing encryption keys
- Managing in-database certificates and certificate collections

To help you integrate SAP HANA securely into your network environment, refer to the section on network administration with detailed information on ports and connections, and host name resolution.

Open the [SAP HANA Administration Guide](#)

SAP HANA Developer Guide

This guide describes the complete application-development process for SAP HANA XS advanced, including aspects of application security, for example:

- Understanding user identity, authentication, and authorization
- Defining the authentication and authorization models
- Protecting applications from Web-based attacks

Open the [SAP HANA Developer Guide](#)

SAP HANA References

The following SAP HANA references contain essential information for administrators and developers with a security focus:

- [SAP HANA SQL and System Views Reference](#)
- [SAP HANA SQL Command Network Protocol](#)
- [SAP HANA Client Interface Programming](#)

i Note

The topics listed above for each guide are not intended to be exhaustive but representative.

→ Tip

For a high-level overview of all security capabilities in the SAP HANA platform, as well as links to security-related blog posts, videos, and white papers, visit <http://sap.com/hanasecurity>.

Target Audiences

| Document | | Content Type |
|--|--|-------------------------|
| SAP HANA Security Guide | Technology consultants, security consultants, system administrators | Concept and overview |
| SAP HANA Security Checklists and Recommendations | System administrators | Reference |
| SAP HANA Administration Guide | System administrators | Task- and role-oriented |
| SAP HANA Developer Guide for XS Advanced Model | Database developers, application programmers and client UI developers working in the SAP HANA XS advanced model using the SAP Web IDE for SAP HANA | Task- and role-oriented |
| SAP HANA SQL and System Views Reference | Technology consultants, security consultants, system administrators | Reference |
| SAP HANA SQL Command Network Protocol Reference | Developers | Reference |
| SAP HANA Client Interface Programming Reference | Developers | Reference |

Additional Documentation Resources

Further SAP HANA Guides

For more information about the SAP HANA landscape, including installation and administration, see [SAP HANA Platform on SAP Help Portal](#).

Important SAP Notes

Important SAP Notes that apply to SAP HANA security are listed in the table below. In addition, SAP publishes information related to security corrections and improvements through SAP security notes. For more information about security notes, see the section on security patches.

i Note

SAP supports that customers install additional tools on the SAP HANA appliance within defined boundaries. It is the responsibility of the customer to ensure that the network channels used by those tools are appropriately protected. For detailed information, see the SAP Notes listed below. For SAP HANA deployments that use the SAP HANA tailored data center integration model, the regulations are less restrictive compared to the appliance delivery model. The listed SAP notes can give guidance of the options available for securing SAP HANA.

| SAP Note | Title |
|-------------------------|---|
| 2380229 | SAP HANA 2.0: Central Note |
| 1730928 | Using external software in an SAP HANA appliance For more information about specific topics, see the quick links in the table below. |
| 1730929 | Using external tools in an SAP HANA appliance |
| 1730930 | Using anti-virus software in an SAP HANA appliance |
| 1730996 | Non-recommended external software and software versions |
| 1730997 | Non-recommended versions of anti-virus software |
| 1730998 | Non-recommended versions of backup tools |
| 1730999 | Configuration changes in SAP HANA appliance |
| 1731000 | Non-recommended configuration changes |

Other Information

| Content | SAP Service Marketplace or SDN Quick Link |
|--------------------------------|--|
| SAP Notes | https://support.sap.com/notes http://support.sap.com/securitynotes |
| Released platforms | https://apps.support.sap.com/sap/support/pam |
| SAP Solution Manager community | https://go.sap.com/community/topic/solution-manager.html |

| Content | SAP Service Marketplace or SDN Quick Link |
|--|---|
| SAP NetWeaver community | https://go.sap.com/community/topic/netweaver.html |
| SAP HANA in-memory computing community | https://go.sap.com/community/topic/hana.html |

Related Information

[SAP HANA Security Patches \[page 13\]](#)

3 SAP HANA Security Patches

To ensure the security of SAP HANA, it's important that you keep your systems up to date by installing the latest SAP HANA revision and monitoring SAP security notes.

SAP HANA Revisions

Security-related code improvements and corrections for SAP HANA are shipped with SAP HANA revisions. SAP publishes information related to security corrections and improvements through SAP security notes. In general, security notes contain information about both the affected SAP HANA application areas and specific measures that protect against the exploitation of potential weaknesses. Additional security measures are also documented here. SAP security notes are released as part of the monthly SAP Security Patch Day.

We recommend that you regularly review new security notes for SAP HANA application areas and decide whether they are relevant in the context of your systems and environment.

For more information about SAP security notes and the SAP Security Patch Day, see SAP Support Portal at <http://support.sap.com/securitynotes>.

i Note

To get full access to SAP Support Portal, you need an authorized user ID.

For a list of all SAP HANA application areas, see the *SAP HANA Master Guide*.

For more information about updating SAP HANA to a new revision, see the *SAP HANA Server Installation and Update Guide*.

Operating System Patches

Install security patches for your operating (OS) system as soon as they become available. If a security patch impacts SAP HANA operation, SAP will publish an SAP Note where this fact is stated. It is up to you to decide whether to install such patches.

If your SAP HANA system runs on SUSE Linux Enterprise Server 11.x for SAP Applications, see SAP Note 1944799.

If your SAP HANA system runs on Red Hat Enterprise Linux (RHEL) 6.x, see SAP Note 2009879.

Related Information

[SAP Note 1944799](#)

SAP Note 2009879 

4 SAP HANA Overview

SAP HANA is an in-memory platform for doing real-time analytics and for developing and deploying real-time applications. For on-premise deployment, SAP HANA comes either pre-installed on certified hardware provided by an SAP hardware partner (appliance delivery model) or must be installed on certified hardware by a certified administrator (tailored data center integration model).

However, SAP HANA is more than a database management system. It is also a comprehensive platform for the development and execution of native data-intensive applications that run efficiently in SAP HANA, taking advantage of its in-memory architecture and parallel execution capabilities.

[The SAP HANA Database \[page 15\]](#)

At the core of SAP HANA is the high-performance, in-memory SAP HANA database.

[SAP HANA XS and Development Infrastructure \[page 16\]](#)

SAP HANA includes the SAP HANA extended application services (SAP HANA XS), a layer on top of SAP HANA that provides the platform for running SAP HANA-based Web applications.

[Technical System Landscape \[page 18\]](#)

An SAP HANA system comprises multiple isolated databases and may consist of one host or a cluster of several hosts (scale-out system).

[SAP HANA Implementation Scenarios \[page 27\]](#)

How you implement SAP HANA determines what you need to consider from a security perspective.

4.1 The SAP HANA Database

At the core of SAP HANA is the high-performance, in-memory SAP HANA database.

SAP HANA is an in-memory platform that combines an ACID-compliant database with advanced data processing, application services, and flexible data integration services. The SAP HANA database can act as a standard SQL-based relational database. In this role, it can serve as either the data provider for classical transactional applications (OLTP) and/or as the data source for analytical requests (OLAP). Database functionality is accessed through an SQL interface.

Standard Database Interfaces

SAP HANA provides standard database interfaces such as JDBC and ODBC and supports standard SQL with SAP HANA-specific extensions.

Data Provisioning

Several data provisioning mechanisms are available for getting data from different sources into SAP HANA. For example, in a data mart or analytics scenario, data is replicated into SAP HANA from source systems using one of the supported replication technologies). For applications that use SAP HANA as their primary database (such as SAP S/4HANA), data is created directly in SAP HANA.

Data Recovery

Although the SAP HANA database holds the bulk of its data in memory for maximum performance, it still uses persistent storage to support system restart and recovery. There's minimal delay and no loss of data in the event of failure. For example, after a power failure, the database can be restarted like any disk-based database and returned to its most recent consistent state. In addition, SAP HANA provides functions for backup and recovery, as well as high availability (disaster recovery and fault recovery).

Related Information

[Security for SAP HANA Replication Technologies \[page 290\]](#)

4.2 SAP HANA XS and Development Infrastructure

SAP HANA includes the SAP HANA extended application services (SAP HANA XS), a layer on top of SAP HANA that provides the platform for running SAP HANA-based Web applications.

SAP HANA XS, Advanced Model

Available since SAP HANA 1.0 SPS 11, the 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.

The SAP HANA XS advanced platform supports several programming languages and execution environments, such as Java, and Node.js. The SAP HANA XS advanced application runtimes are invoked over HTTP and communicate with the SAP HANA database via SQL.

The database part of an SAP HANA XS advanced application (for example the definitions of tables, views, and procedures) is deployed using the SAP HANA deployment infrastructure (SAP HANA DI, or HDI). HDI is a service layer of the SAP HANA database that simplifies the consistent deployment of SAP HANA database objects. It supports isolated deployment containers, which can be used, for example, to deploy several instances of the same application on the same SAP HANA database.

SAP Web IDE for SAP HANA is the browser-based development environment for SAP HANA-based applications. It can be used to develop all layers of an application, including UI, XS advanced server applications, and SAP HANA database content. It is based on SAP HANA XS advanced and HDI, and uses Git for source code management.

→ Recommendation

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 runtime 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. For more information, see the *SAP HANA XS Advanced Migration Guide*.

Downloading XS Advanced from SAP Marketplace

SAP HANA Extended Application Services, advanced model, is available not only on the SAP HANA media but also as a separate component on SAP Marketplace. Users with the required S-User ID can download the latest version of XS advanced component in the package `SAP_EXTENDED_APP_SERVICES_1` from the following location:

▶ [Service Marketplace](#) ▶ [Software Downloads \[Downloads\]](#) ▶ [SUPPORT PACKAGES & PATCHES](#) ▶ [By Alphabetical Index \(A-Z\)](#) ▶ [H](#) ▶ [SAP HANA PLATFORM EDITION](#) ▶

- ▶ [SAP HANA PLATFORM EDITION 1.0](#) ▶ [XS ADVANCED RUNTIME](#) ▶ [SAP EXTENDED APP SERVICES 1](#) ▶
- ▶ [SAP HANA PLATFORM EDITION 2.0](#) ▶ [SAP EXTENDED APP SERVICES 1](#) ▶

→ Tip

SAP HANA Extended Application Services, advanced model, is backwards compatible; you can provide access to new features by installing the latest version of the XS advanced component even on older versions of SAP HANA. To download the package `SAP_EXTENDED_APP_SERVICES_1`, see *SAP Software Download Center* in *Related Information* below.

SAP HANA XS, Classic Model

SAP HANA XS classic is the original implementation of SAP HANA XS. The classic XS server is fully integrated into the SAP HANA database and provides application server functions. Accessible through HTTP, the XS server can deliver data through Open Data Protocol (OData) calls and HTML user interfaces. For creating new structures and programs, for example modeling database structures, analytical queries, reports and procedures, as well as developing applications, SAP HANA provides a development environment. This development environment is integrated into the SAP HANA studio and the SAP HANA Web-based Development Workbench. Design-time artifacts, such as custom applications, roles, and application content, are managed in SAP HANA's built-in repository. Design-time objects can be transported from development systems to test and production systems.

i Note

SAP HANA XS, classic and the SAP HANA repository are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.

Related Information

[SAP HANA as Technical Infrastructure for Native Application Development \[page 31\]](#)

[Security for SAP HANA Extended Application Services, Advanced Model \[page 293\]](#)

[Security Aspects of SAP Web IDE for SAP HANA \[page 348\]](#)

[SAP Note 2465027](#)

[SAP Software Download Center \(Logon required\)](#)

4.3 Technical System Landscape

An SAP HANA system comprises multiple isolated databases and may consist of one host or a cluster of several hosts (scale-out system).

An SAP HANA system, identified by a single system ID (SID), contains one or more tenant databases and one system database. Databases are identified by a SID and a database name. From the administration perspective, there is a distinction between tasks performed at system level and those performed at database level. Database clients, such as the SAP HANA cockpit, connect to specific databases.

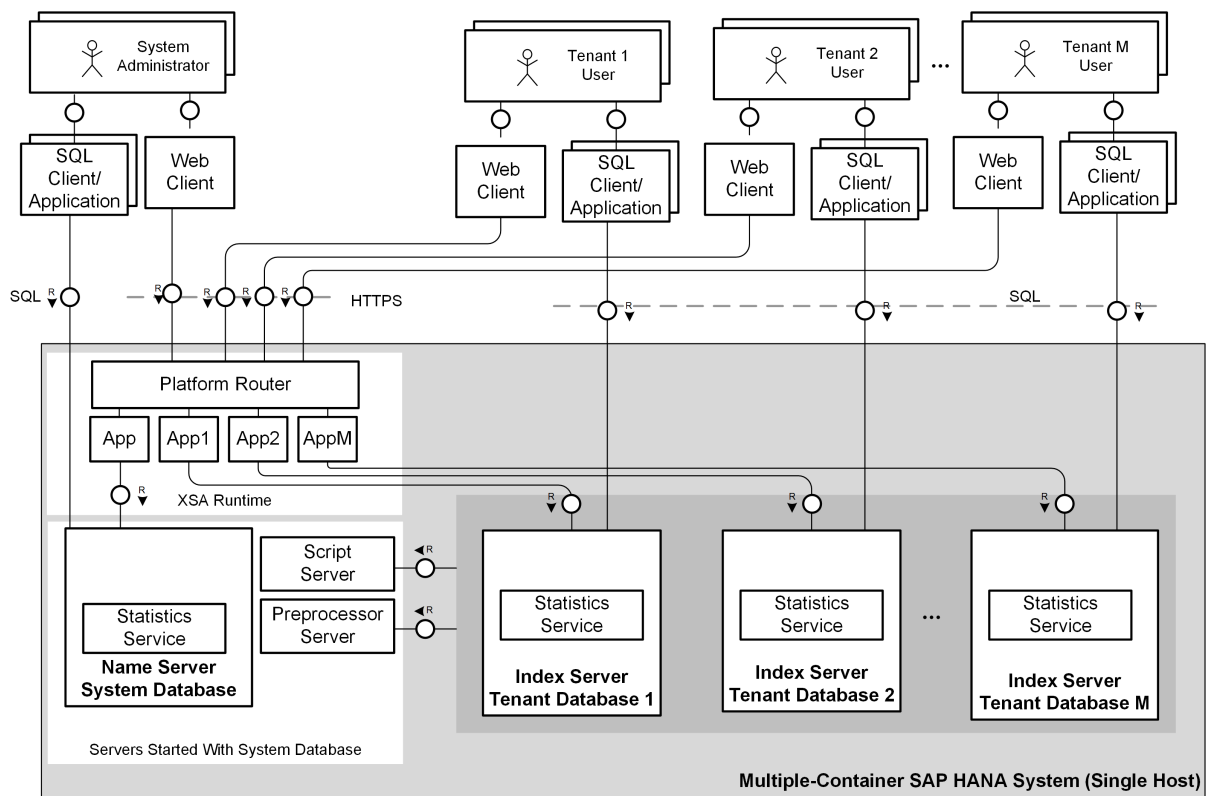
All the databases in a system share the same installation of database system software, the same computing resources, and the same system administration. However, each database is self-contained and fully isolated with its own set of database users, database catalog, persistence, and so on.

The System Database

The system database, which is created during installation, is used for central system administration, for example the creation of tenant databases and global system configuration. The system database stores overall system landscape information, including knowledge of the tenant databases that exist in the system. However, it doesn't own database-related topology information, that is, information about the location of tables and table partitions in databases. Database-related topology information is stored in the relevant tenant database catalog.

Server Architecture

An example of the basic architecture of a single-host SAP HANA system with three tenant databases is shown below. For more information about system architecture, see the *SAP HANA Administration Guide*.



SAP HANA System with Tenant Databases

[Overview of SAP HANA Security Functions \[page 20\]](#)

SAP HANA provides a range of security features and functions at the database and system level to ensure secure access control and secure system setup and configuration.

[Database Isolation \[page 24\]](#)

Every tenant database is self-contained and isolated in terms of users, database catalog, repository, logs, and so on. However, to protect against unauthorized access at the operating system (OS) level, it's possible to increase isolation further through OS user separation and authenticated communication within databases.

[Security Considerations After Updating a Single-Container System \[page 26\]](#)

As of SAP HANA 2.0 SPS 01, all SAP HANA systems support tenant databases. If you updated a single-container system, this means that your system now has a system database and one tenant database. You therefore need to review aspects of your operations concept, including security.

4.3.1 Overview of SAP HANA Security Functions

SAP HANA provides a range of security features and functions at the database and system level to ensure secure access control and secure system setup and configuration.

Security Features of the SAP HANA Database

The following table provides an overview of standard security features in the SAP HANA database. For more detailed information, see the relevant section in this guide.

| Security Feature | Description |
|--------------------------|--|
| User and role management | <p>Every tenant database has its own database users and roles, including a tenant database-specific superuser SYSTEM.</p> <p>Depending on the isolation level of the system, there may be only one operating system (OS) user (the default <sid>adm user), or one OS user for each tenant database, which must be created.</p> |

| Security Feature | Description |
|------------------------|---|
| Authentication and SSO | <p>The SAP HANA database supports a number of authentication mechanisms, including database user name/password, SAML bearer tokens, JSON Web tokens, Kerberos, and LDAP directory server name and password. Whether a per-database configuration is possible depends on the authentication mechanism and the user client:</p> <ul style="list-style-type: none"> • Authentication by database user name and password is database specific. • For Kerberos-based authentication, a per-database configuration is not possible. Databases users in all databases must be mapped to users in the same Key Distribution Center. • For SAML and JWT-based authentication, a per-database configuration is possible for JDBC/ODBC client access. Different trust stores (containing different certificates) can be configured for individual databases. For this purpose, we recommend using certificates and certificate collections (also referred to as personal security environments or PSEs) stored in the database as opposed to the file system. • For LDAP-based authentication, a per-database configuration is possible. Connections to different LDAP directory servers can be set up by creating separate LDAP providers in each database. To secure communication between the SAP HANA database and the LDAP server (including the transmission of passwords), different trust stores (containing different certificates) can be configured for individual databases using in-memory certificates and certificate collections. <div data-bbox="651 1081 1396 1216" style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;"> <p>i Note</p> <p>LDAP-based authentication is only possible for users if authentication using their local SAP HANA password is disabled.</p> </div> <ul style="list-style-type: none"> • Database-specific trust stores cannot be configured for HTTP client access through SAP HANA Extended Services, classic model (SAP HANA XS classic). Therefore, user authentication based on SAML assertions and X.509 certificates cannot be database specific. |
| Authorization | <p>SAP HANA's standard authorization mechanisms are applied to users at the database level with the following additions:</p> <ul style="list-style-type: none"> • In the system database, the system privilege DATABASE ADMIN exists to allow system administrators to perform certain tasks on tenant databases (for example, stop a tenant database or back up a tenant database). • A cross-database authorization mechanism exists to support read-only queries between tenant databases. This is made possible through the association of a user in one tenant database with a user in another database. Cross-database access is disabled by default and must be enabled and configured by a system administrator before such user mappings can be set up. |

| Security Feature | Description |
|--|---|
| Encryption of data communication in the network | <p>Secure communication based on the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol can be configured separately for external communication between individual databases and JDBC/ODBC clients. Separate key and trust stores must be available and configured for each database. For this purpose, we recommend using certificates and certificate collections stored in the database as opposed to the file system.</p> <p>For communication with HTTP clients, a per-database configuration of TLS/SSL keys and certificates is also possible.</p> |
| Data-at-rest encryption in the persistence layer | <p>Data and log volume encryption, as well as data and log backup encryption can be enabled for the system database and tenant databases individually. Data and log volume encryption ensures that anyone who can access the data and log volumes on disk using operating system commands cannot see the actual data and redo log entries. Backup encryption prevents unauthorized parties from reading the content of backups.</p> |
| Masking and anonymization | <p>Data masking is an additional layer of access control that can be applied to tables and views. A column mask protects sensitive or confidential data in a particular column of a table or view by transforming the data in such a way that it is only visible partially or rendered completely meaningless for an unprivileged user, while still appearing real and consistent.</p> <p>Anonymization allow you to gain statistically valid insights from your data while protecting the privacy of individuals. Unlike masking and pseudonymization, anonymization methods (also called privacy-enhancing methods) provide a more structured approach to modifying data for privacy protection. The quality of such anonymized or privacy-enhanced data is still sufficient for meaningful analysis. Data anonymization capabilities are integrated into calculation views.</p> <p>Masking and anonymization are both applied at the database level.</p> |
| Auditing | <p>Actions performed in every tenant database and the system database can be audited individually.</p> <p>To ensure the privacy of tenant database audit trails, they are by default written to a database table that is local to the database being audited. By default, tenant database administrators cannot change the audit trail targets for their database. The system administrator can , but this is not recommended.</p> <p>If the audit trail target for tenant databases is changed to the syslog, audit entries contain a field <i>Database Name</i> so that it is possible to differentiate entries from different tenant databases.</p> |

Additional System-Level Security Features

The following table provides an overview of additional feature to ensure the secure setup and configuration of the SAP HANA system .

| Security Feature | Description |
|--------------------------------|---|
| Database isolation | To maximize system security by preventing cross-tenant attacks through operating system mechanisms, it is possible to configure the system for high isolation. In high isolation mode, the processes of individual tenant databases must run under dedicated OS users belonging to dedicated OS groups, instead of all processes running under the single default OS user <code><sid>adm</code> (low isolation). Data in the file system is protected using file and directory permissions. |
| Configuration change blacklist | To ensure the stability and performance of the overall system or for security reasons, it may be necessary to prevent certain system properties from being changed by tenant database administrators, for example, properties related to resource management. A configuration change blacklist (<code>multidb.ini</code>) is available for this purpose. This blacklist contains several critical properties by default. You can customize the default configuration as well as add further properties by editing the file, for example, in the SAP HANA cockpit. |
| Restricted features | To safeguard and/or customize your system, certain features of the SAP HANA database can be disabled in tenant databases. |

Related Information

- [SAP HANA User Management \[page 70\]](#)
- [SAP HANA Authentication and Single Sign-On \[page 92\]](#)
- [Certificate Management in SAP HANA \[page 261\]](#)
- [SAP HANA Authorization \[page 119\]](#)
- [Securing Data Communication \[page 40\]](#)
- [Data and Log Volume Encryption \[page 209\]](#)
- [Backup Encryption \[page 212\]](#)
- [Auditing Activity in SAP HANA Systems \[page 240\]](#)
- [Data Masking \[page 188\]](#)
- [Database Isolation \[page 24\]](#)
- [Restricted Features in Tenant Databases \[page 358\]](#)

4.3.2 Database Isolation

Every tenant database is self-contained and isolated in terms of users, database catalog, repository, logs, and so on. However, to protect against unauthorized access at the operating system (OS) level, it's possible to increase isolation further through OS user separation and authenticated communication within databases.

OS User Separation

By default, all database processes run under the default OS user `<sid>adm`. If it's important to mitigate against cross-database attacks through OS mechanisms, you can configure the system for high isolation. In this way, the processes of individual tenant databases must run under dedicated OS users belonging to dedicated OS groups, instead of all database processes running under `<sid>adm`. Database-specific data on the file system is subsequently protected using standard OS file and directory permissions.

i Note

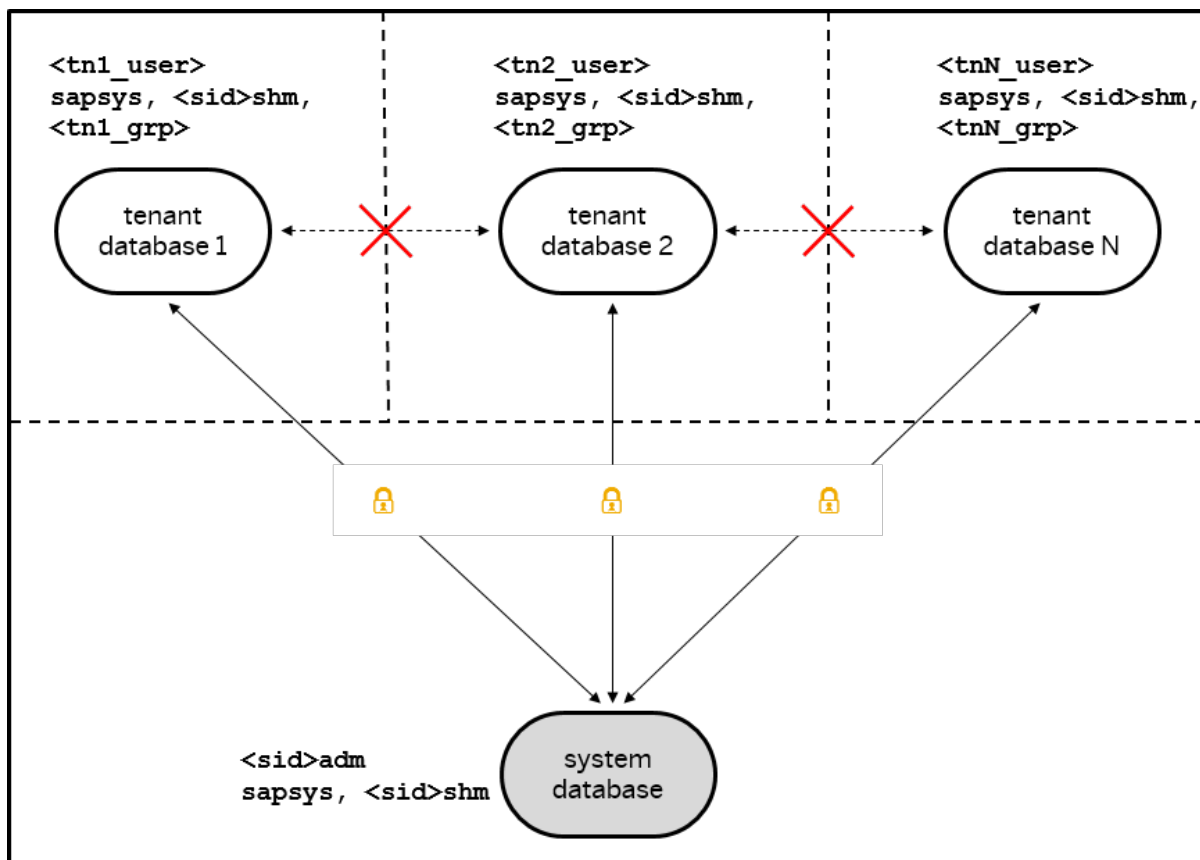
`<sid>adm` is the OS user for the system database.

Authenticated Communication

In addition, once high isolation has been configured, internal database communication is secured using the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol. Certificate-based authentication is used to ensure that only the processes belonging to the same database can communicate with each other. It is also possible to configure internal communication so that all data communication within databases is encrypted.

i Note

If cross-database access is enabled, communication between configured tenant databases is allowed.



High Database Isolation

Configuration

You can specify the isolation level of the system during installation. The default isolation level is low. It is also possible to change the isolation level of an existing system (from low to high or from high to low) at any time. For more information about how to do this, see *Increase the System Isolation Level* in the *SAP HANA Administration Guide*. Once high isolation has been configured, a dedicated OS user and group must exist for every tenant database. Otherwise, it's not possible to create or start a tenant database.

Internal database communication is secured with the same mechanism used for securing other internal SAP HANA communication channels. Once high isolation has been configured, authenticated communication within databases is enabled without any change required to the default TLS/SSL configuration for internal communication. However, encryption of data communication may need to be configured explicitly.

Related Information

[Secure Internal Communication \[page 60\]](#)

4.3.3 Security Considerations After Updating a Single-Container System

As of SAP HANA 2.0 SPS 01, all SAP HANA systems support tenant databases. If you updated a single-container system, this means that your system now has a system database and one tenant database. You therefore need to review aspects of your operations concept, including security.

| Configuration Area | After Update |
|--|--|
| User administration | You need to set up new users for administration (at least for recovery) in the system database. |
| Network security | <p>The system database uses additional ports. These ports that might be firewalled for security reasons in the system. If this the case, you need to open these ports so that the system database can be accessed from the SAP HANA cockpit on other hosts.</p> <p>The system database is accessible via SQL port 3<instance_no>13.</p> |
| TLS/SSL configuration for external communication | <p>If TLS/SSL is enabled for both the system database and tenant database, the in-database certificate collection containing the certificates used for trust validation is available only in the tenant database. If you want to use the same certificates, you will need import them into the certificate store of the system database and add them to a certificate collection there.</p> <div data-bbox="603 1048 1394 1249"><p>⚠ Caution</p><p>If TLS/SSL is being enforced for client connections (that is, parameter [communication] <code>sslEnforce</code> in <code>global.ini</code> set to <code>true</code>), it will not be possible to establish a connection to the system database. You have to set <code>sslEnforce</code> to <code>false</code> first.</p></div> <p>If the certificates used for trust validation are stored in a PSE in the file system, both the tenant database and the system database will have access, so no reconfiguration is required.</p> <p>However, you should validate that sharing the certificate stores for system database and tenant is actually intended.</p> |
| Database isolation | The default system isolation level is low. It is possible to change the isolation level (from low to high or from high to low) at any time after the update. Once high isolation has been configured, a dedicated OS user and group must exist for every tenant database. Otherwise, it's not possible to create or start a tenant database. |
| Auditing | Existing audit policies are available in the tenant database database only. You need to create new audit policies for administration tasks in the system database. |

For more information about non-security-related aspects, see the update section of the *SAP HANA Server Installation and Update Guide*.

4.4 SAP HANA Implementation Scenarios

How you implement SAP HANA determines what you need to consider from a security perspective.

For more detailed information, see the section on SAP HANA use cases in the *SAP HANA Master Guide*.

[SAP HANA as a Data Mart \[page 27\]](#)

In a data mart scenario, data is replicated from a source system such as SAP Business Suite into the SAP HANA database. Reporting is then carried out on the data in SAP HANA (for example, using read-only views, dashboards, and so on). Different architectures can be used in this scenario.

[SAP HANA in a Classic 3-tier Architecture \[page 29\]](#)

SAP HANA can be used as a relational database in a classic 3-tier architecture (client, application server, and database).

[SAP HANA as Technical Infrastructure for Native Application Development \[page 31\]](#)

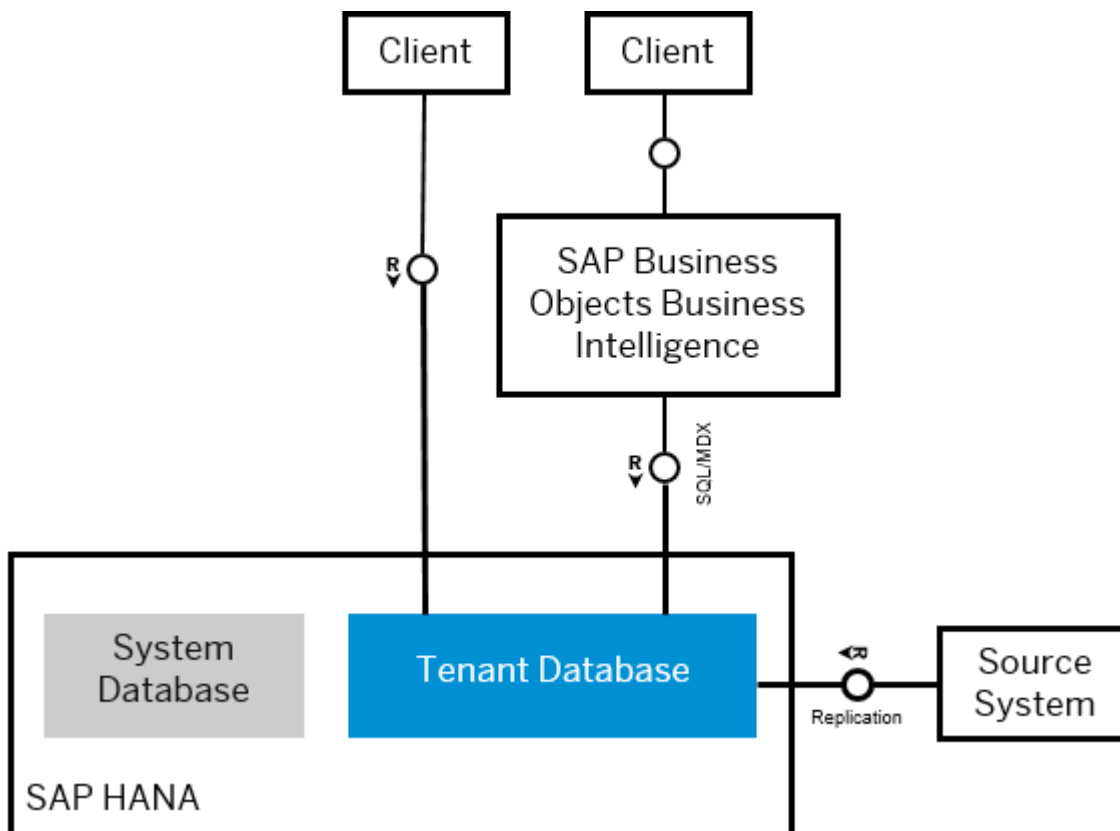
SAP HANA extended application services (SAP HANA XS), advanced model is the default framework for native application development on SAP HANA. The SAP HANA XS, classic model is the deprecated application development framework of SAP HANA.

4.4.1 SAP HANA as a Data Mart

In a data mart scenario, data is replicated from a source system such as SAP Business Suite into the SAP HANA database. Reporting is then carried out on the data in SAP HANA (for example, using read-only views, dashboards, and so on). Different architectures can be used in this scenario.

For example, SAP HANA can be integrated into the SAP BusinessObjects Business Intelligence (BI) platform as a relational database. The source data can then be analyzed and reported on by SAP BusinessObjects Business Intelligence Suite products. Alternatively, SAP HANA can be accessed directly by BI clients such as Microsoft Excel. In this case, end-user clients connect directly to the database. These architectures are depicted in the following figure:

SAP HANA as a Data Mart



The implemented architecture determines the extent to which security-related aspects are handled in SAP HANA. However, user and role management in the database layer of SAP HANA is required, at least for technical users and administrators.

The following table outlines the relevance of SAP HANA security-related features in this implementation scenario.

| SAP HANA Feature | Relevance in Scenario |
|--------------------------|---|
| User and role management | <p>The extent to which SAP HANA user and role management is required in this scenario depends on your system architecture as follows.</p> <ul style="list-style-type: none"> • If SAP HANA is integrated into a business intelligence solution (for example, SAP BusinessObjects Business Intelligence platform) only as the reporting database, end users and roles are managed in the relevant application server. User and role management in the database layer of SAP HANA is required only for technical database users and administrators. • If end users connect to the SAP HANA database directly through a SQL client (for example, SAP BusinessObjects Explorer or Microsoft Excel), user and role management in the database layer of SAP HANA is required for both end users and administrators. |

| SAP HANA Feature | Relevance in Scenario |
|---|---|
| Authentication and SSO | <p>The extent to which authentication and SSO is handled in SAP HANA depends on your system architecture in the same way as described above.</p> <ul style="list-style-type: none"> • If SAP HANA is used only as the data store, end-user authentication is handled in the application server. The relevant technical database user accounts are used to authenticate connections to the database. • If end users connect to the SAP HANA database directly through a SQL client, the database user is authenticated. End-user clients support several authentication mechanisms for integration into SSO environments (SAML, Kerberos, SAP logon /assertion tickets). |
| Authorization | SAP HANA authorization applies to users managed directly in the database. |
| Encryption of data communication in the network | Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are supported and recommended for network communication where possible. |
| Encryption in the persistence layer | Data and log volume encryption ensures that anyone who can access the data and log volumes on disk using operating system commands cannot see the actual data and redo log entries. |
| Auditing | Actions performed in the SAP HANA database can be audited. |

Related Information

[SAP HANA User Management \[page 70\]](#)

[SAP HANA Authentication and Single Sign-On \[page 92\]](#)

[SAP HANA Authorization \[page 119\]](#)

[Securing Data Communication \[page 40\]](#)

[Data and Log Volume Encryption \[page 209\]](#)

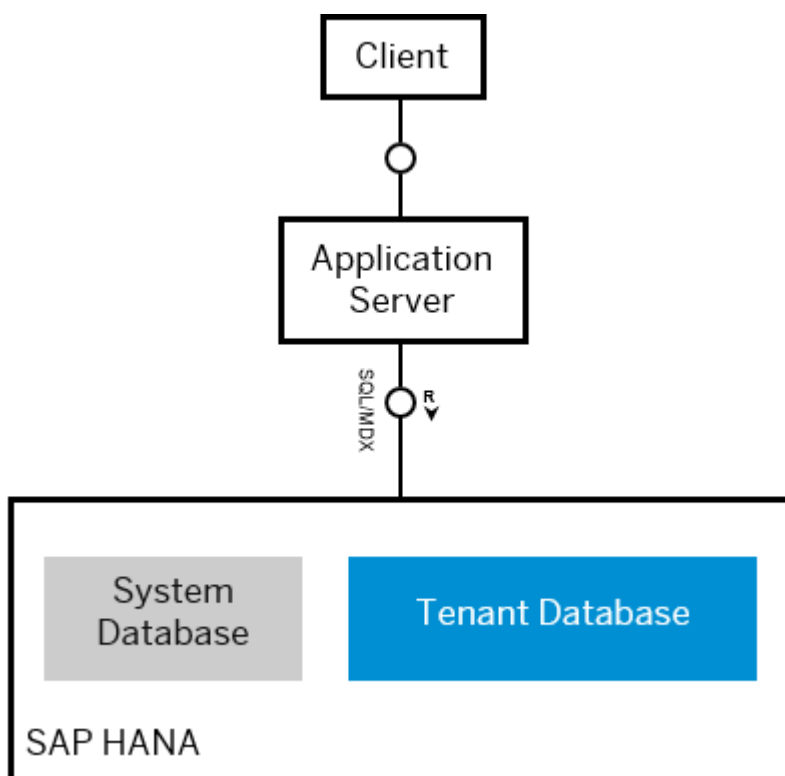
[Auditing Activity in SAP HANA Systems \[page 240\]](#)

4.4.2 SAP HANA in a Classic 3-tier Architecture

SAP HANA can be used as a relational database in a classic 3-tier architecture (client, application server, and database).

This architecture is depicted in the following figure:

SAP HANA in 3-tier Architecture



In this architecture, security-related features, such as authentication, authorization, encryption, and auditing, are located and enforced primarily in the application server layer. The database is used as a data store only. Applications connect to the database using a technical user, and direct access to the database is only possible for database administrators. End users do not have direct access to either the database itself or the database server on which it's running.

As a consequence, security in the database layer is mainly focused on securing administrative access to the database. Typical examples of this architecture are the SAP S/4HANA and SAP BW. When SAP HANA is used as a database in these scenarios, the same security approach applies, and specific SAP HANA security features are mainly needed to control access of administrators to the database.

The following table outlines the relevance of SAP HANA security-related features in this implementation scenario.

| SAP HANA Feature | Relevance in Scenario |
|--------------------------|---|
| User and role management | <p>End users and roles are managed in the application server layer. For example, SAP S/4HANA applications use the user management and authentication mechanisms of the SAP NetWeaver platform, and in particular, SAP NetWeaver Application Server.</p> <p>User and role management in the database layer of SAP HANA is required only for technical database users and administrators.</p> |

| SAP HANA Feature | Relevance in Scenario |
|---|---|
| Authentication and SSO | <p>End-user authentication is handled in the application server layer.</p> <p>The relevant technical database users are used to authenticate connections to the database.</p> <p>Administrators with direct access to the database must be authenticated in the database. Administration clients that access the database through SQL (for example, the SAP HANA studio and the <code>hdsql</code> command line tool) support the authentication mechanisms Kerberos and SAP logon/assertion tickets for integration into SSO environments.</p> |
| Authorization | SAP HANA authorization applies only to technical and administrative database users managed in the database. |
| Encryption of data communication in the network | Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are supported and recommended for network communication where possible. |
| Encryption in the persistence layer | Data and log volume encryption ensures that anyone who can access the data and log volumes on disk using operating system commands cannot see the actual data and redo log entries. |
| Auditing | Actions performed in the SAP HANA database can be audited. |

Related Information

[SAP HANA User Management \[page 70\]](#)

[SAP HANA Authentication and Single Sign-On \[page 92\]](#)

[SAP HANA Authorization \[page 119\]](#)

[Securing Data Communication \[page 40\]](#)

[Data and Log Volume Encryption \[page 209\]](#)

[Auditing Activity in SAP HANA Systems \[page 240\]](#)

4.4.3 SAP HANA as Technical Infrastructure for Native Application Development

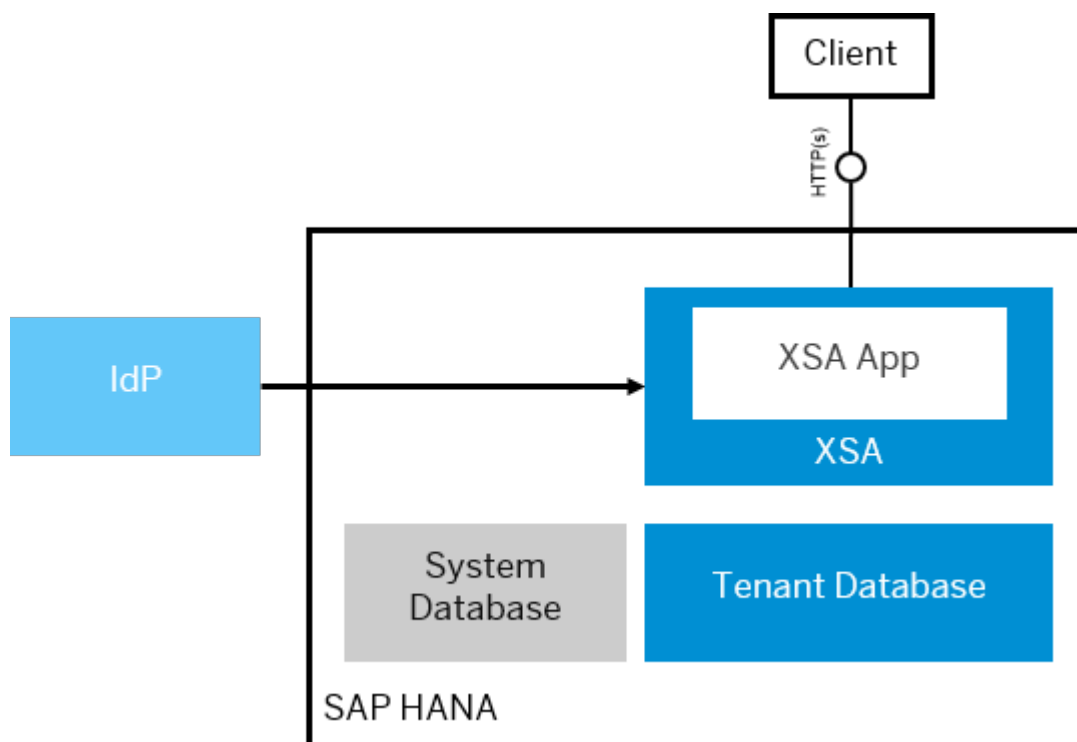
SAP HANA extended application services (SAP HANA XS), advanced model is the default framework for native application development on SAP HANA. The SAP HANA XS, classic model is the deprecated application development framework of SAP HANA.

SAP HANA XS Advanced Model

The SAP XS advanced model provides a comprehensive platform for the development and execution of native data-intensive applications. XS advanced supports a variety of programming languages, such as SQLScript for

execution on the data layer, or Java and node.js for execution in the application server runtime. End-user clients access applications developed on XS advanced via HTTP(S), while the application run-times communicate with the SAP HANA database via SQL.

The architecture of SAP HANA XS advanced (simplified) is depicted in the following figure:



SAP HANA as Technical Infrastructure for Native Applications, XS Advanced

Security and XS Advanced

XS advanced provides deployment flexibility and specific security options.

With XS advanced, the application and database layer can be decoupled. You can either install XS advanced directly on the SAP HANA server, or install it on a separate host to the SAP HANA database. This enables you to scale XS advanced independently of the database, as you can have many more XS advanced hosts than SAP HANA database hosts. You can also install XS advanced in a separate network from SAP HANA itself, which makes it possible to put XS advanced applications into a different network zone and have a firewall between the application and database layers.

XS advanced applications are strictly isolated from each other. They are deployed in dedicated containers via the SAP HANA deployment infrastructure (HDI) at the database layer. At the application layer, you can use dedicated operation system users per application.

For detailed information about the security architecture of SAP HANA XS, advanced model, see the corresponding section in this guide.

For more information about secure application development, see the section *Maintaining Application Security in XS Advanced* in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*.

→ Recommendation

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. For more information, see the *SAP HANA XS Advanced Migration Guide*.

SAP HANA XS Classic Model

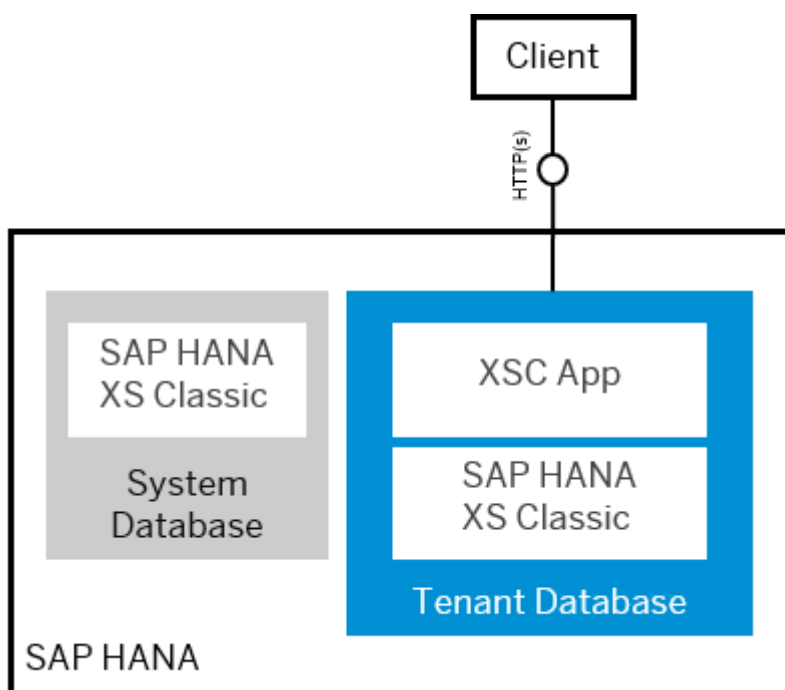
The SAP HANA XS classic model embeds a full-featured application server, Web server, and development environment within SAP HANA. Applications can be developed and deployed directly on SAP HANA XS, which exposes them to end users through a web interface.

Note

SAP HANA XS classic and the SAP HANA repository are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.

The architecture of SAP HANA XS classic is depicted in the following figure:

SAP HANA as Technical Infrastructure for Native Applications, XS Classic



Classic native SAP HANA applications rely on the security-related features of SAP HANA. In particular, users of native SAP HANA applications must always have a corresponding user in the SAP HANA database.

The following table outlines the relevance of SAP HANA security-related features in this implementation scenario.

| SAP HANA Feature | Relevance in Scenario |
|---|---|
| User and role management | User and roles are managed fully in SAP HANA. |
| Authentication and SSO | <p>The database user is used to authenticate not only users connecting to the database through the SQL interface, but also to HTTP clients that connect to SAP HANA XS.</p> <p>Several mechanisms are supported for the integration of HTTP access through SAP HANA XS into SSO environments, including SAML, X.509 client certificates, Kerberos with Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO), and SAP logon/assertion tickets.</p> |
| Authorization | User access to native SAP HANA applications and applications functions is determined by the privileges granted to the database user. |
| Encryption of data communication in the network | <p>SSL/TLS are supported and recommended for network communication where possible.</p> <p>The SAP Web Dispatcher can be configured to use HTTPS to secure connections between HTTP client applications and SAP HANA.</p> |
| Encryption in the persistence layer | Data and log volume encryption ensures that anyone who can access the data and log volumes on disk using operating system commands cannot see the actual data and redo log entries. |
| Auditing | Actions performed in the SAP HANA database can be audited. |

Secure Application Development

For more security information about the following aspects related to SAP HANA XS application development, see the section on *Maintaining Application Security* in the *SAP HANA Developer Guide for SAP HANA XS Classic Model*.

Related Information

[Security for SAP HANA Extended Application Services, Advanced Model \[page 293\]](#)

[SAP HANA User Management \[page 70\]](#)

[SAP HANA Authentication and Single Sign-On \[page 92\]](#)

[SAP HANA Authorization \[page 119\]](#)

[Securing Data Communication \[page 40\]](#)

[Data and Log Volume Encryption \[page 209\]](#)

[Auditing Activity in SAP HANA Systems \[page 240\]](#)

[SAP Note 2465027 !\[\]\(6a9b39b98eb945faa14c645ec99e4eaa_img.jpg\)](#)

5 SAP HANA Network and Communication Security

Several mechanisms are possible for securing network communication in the SAP HANA landscape.

SAP HANA supports encrypted communication for network communication channels. We recommend using encrypted channels in all cases where your network isn't protected by other security measures against attacks such as eavesdropping, for example, when your network is accessed from public networks. Alternatively, use virtual private network (VPN) tunnels to transfer encrypted information.

For more information about network administration, see the *SAP HANA Administration Guide*.

[Communication Channels \[page 35\]](#)

The network communication channels used by SAP HANA can be categorized into those used for database clients connecting to SAP HANA and those used for internal database communication. SAP recommends using encrypted communication channels where possible.

[Network Security \[page 37\]](#)

To integrate SAP HANA securely into your network environment, several general recommendations apply.

[Securing Data Communication \[page 40\]](#)

SAP HANA supports encrypted communication for client-server (external) communication and internal communication.

5.1 Communication Channels

The network communication channels used by SAP HANA can be categorized into those used for database clients connecting to SAP HANA and those used for internal database communication. SAP recommends using encrypted communication channels where possible.

The following is an overview of the network communication channels used by SAP HANA.

To support the different SAP HANA scenarios and set-ups, SAP HANA has different types of network communication channels:

- Channels used for external access to SAP HANA functionality by end-user clients, administration clients, application servers, and for data provisioning through SQL or HTTP
- Channels used for SAP HANA internal communication within the database, between hosts in multiple-host systems, and between systems in system-replication scenarios

i Note

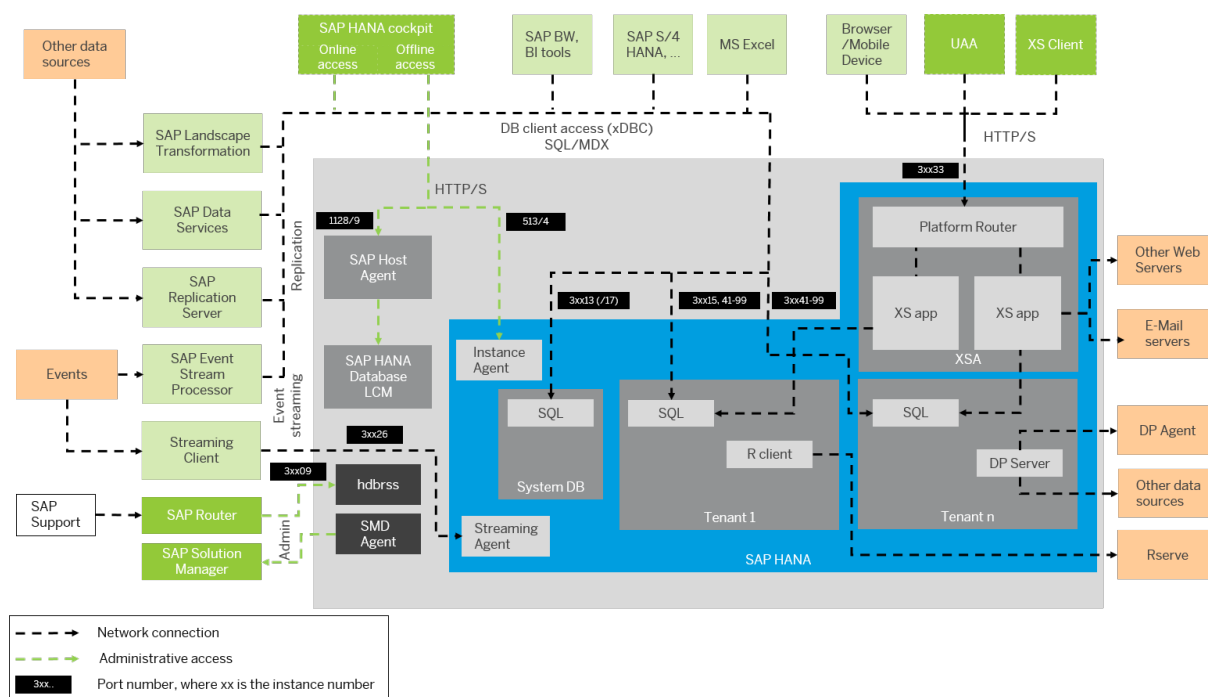
SAP HANA internal communication has sometimes been unofficially referred to as TREXNet communication. However, the term TREXNet is not valid in the context of SAP HANA.

The connections between SAP HANA and external components and applications come under these categories:

- Connections for administrative purposes
- Connections for data provisioning
- Connections from database clients that access the SQL/MDX interface of the SAP HANA database
- Connections from HTTP/S clients
- Outbound connections

You can see an example of what these connections look like in the figure below. Network connections are depicted by dotted arrows. The direction of each arrow indicates which component is the initiator and which component is the listener. Administrative access to and from SAP HANA is depicted by the green dashed arrows. Port numbers are shown with a black background. The "xx" in the port numbers stands for the number of your SAP HANA instance.

The figure below shows all the network channels used by SAP HANA. For the purposes of illustration, a single-host installation with two tenant databases is depicted. However, the connections shown apply equally to a distributed scenario.



Connections Between SAP HANA and External Components

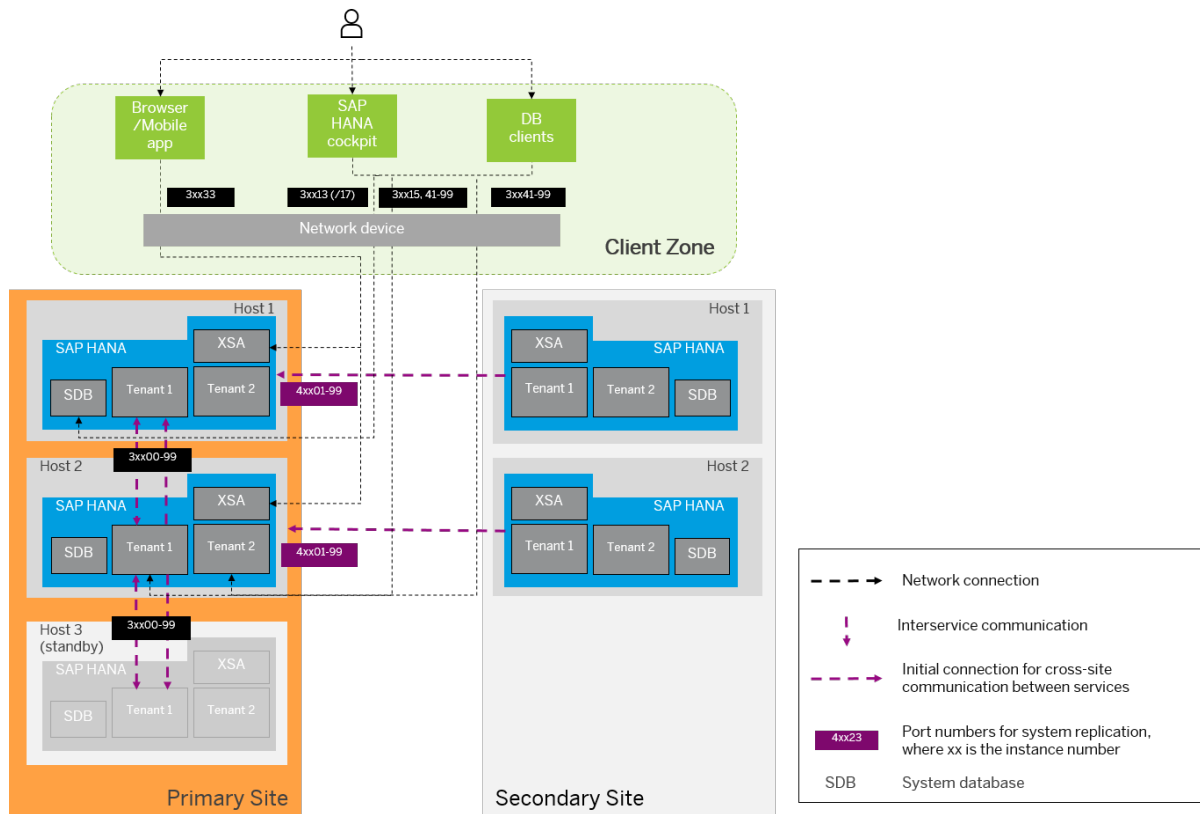
Note

Some components depicted in the figure are supported on Intel-based hardware platforms only (for example, SAP HANA Streaming Analytics). Refer to the Product Availability Matrix (PAM).

In addition, the different components of SAP HANA, as well as the hosts in a distributed scenario, communicate with each other over internal SAP HANA connections. These connections are also used in system replication scenarios for communication between a primary site and secondary site(s) to ensure high availability in the event of a data center failure.

The following figure shows an example of a distributed SAP HANA system with two active hosts and an extra standby host, both fully replicated to a secondary site to provide full disaster recovery support.

SAP HANA Internal Connections



Related Information

[Securing Data Communication \[page 40\]](#)

[Product Availability Matrix](#)

5.2 Network Security

To integrate SAP HANA securely into your network environment, several general recommendations apply.

The components of an SAP HANA landscape communicate through different network communication channels. It is recommended security practice to have a well-defined network topology to control and limit network access to SAP HANA to only those communication channels needed for your scenario, and to apply appropriate additional security measures, such as encryption, where necessary. This can be achieved through different means, such as separate network zones and network firewalls, and through the configuration options provided by SAP HANA (for example, encryption). The exact setup depends on your environment, your implementation scenario, and your security requirements and policies.

The detailed network set-up and recommendations are described in network administration section of the *SAP HANA Administration Guide*. This section contains some additional security-relevant information.

⚠ Caution

It is strongly recommended that you apply the measures described in this section to protect access to the SAP HANA database's internal communication channels and to mitigate the risk of unauthorized access to these services.

Network Zones

- We recommend that you operate the different components of the SAP HANA platform in separate network zones.
To prevent unauthorized access to the SAP HANA environment and the SAP HANA database through the network, use network firewall technology to create network zones for the different components and to restrictively filter the traffic between these zones implementing a "minimum required communication" approach. The relevant network zones depend on your specific application scenario and your network infrastructure. For more information about network zones, see the *SAP HANA Administration Guide*.
- We recommend that you operate SAP HANA in a protected data-center environment. Allow only dedicated authorized network traffic from other network zones (for example, user access from the client network zone) to follow these rules:
 - Clients accessing external standard database functionality, for example by SQL, have only access to the database client access port.
 - Clients (for example, browser applications) accessing the SAP HANA environment through the HTTP access feature of SAP HANA Extended Application Services, classic model (XS classic), for example SAP HANA UI Toolkit for Info Access, have only access to the SAP HANA XS ports.
 - Some administrative functions (for example, starting and stopping the SAP HANA instance) have access to the administrative ports.
 - XS classic exposes some administrative applications (for example, administration of Security Assertion Markup Language (SAML) for user authentication). We recommend using URL filtering (for example, reverse proxy) to control the exposure of different applications to different network zones.

Internal Communication

Database internal communication channels are only used for the following:

- Communication within the database
- Communication between hosts in distributed (multiple-host) scenarios
- Communication between multiple sites in system replication (high-availability) scenarios
- Communication between the SAP HANA database and server components, such as extended storage (SAP HANA dynamic tiering)

Note the following network security considerations for single-host, multiple-host, and system replication (high-availability) scenarios.

Single-Host Scenario

In a single-host scenario, access to the network ports for database internal communication from other network hosts is blocked by default. We recommend that you do not change this setting. The internal communication ports are bound to `localhost`.

i Note

In single-host scenarios, the same communication channels are used for communication between the different processes on a single host, and the internal IP addresses/ports are by default bound to the `localhost` interface. Note that this does not apply to dynamic tiering. The dynamic tiering service is bound to all interfaces, although the internal communication between SAP HANA database and dynamic tiering uses the `localhost` interface.

Multiple-Host Scenario

In a distributed scenario (that is, one instance of SAP HANA on multiple hosts), internal network communication takes place between the hosts at one site via internal. Certified SAP HANA hosts contain either dedicated or virtualized network interfaces that are configured as part of a private network using separate IP addresses and ports.

We recommend operating all hosts in a dedicated sub-network.

To prevent unauthorized access to the database via the internal communication channels in distributed systems, we recommend that you prevent access to these network channels and ports from outside the system. There are a number of ways to isolate internal network ports from the client network:

- Using the SAP HANA configuration option to route communication between the hosts of a distributed environment onto a specified network and binding those internal network services exclusively to the network interface (**recommended option**)
For more information about configuring inter-service communication, see the *SAP HANA Administration Guide*.

i Note

In system replication scenarios, you can use this feature in the presence of a secondary site. However, note that additional ports used for communication between primary and secondary sites are opened on the network interface. These ports need to be protected.

- Using operating system commands (for example, `iptables`), and/or network device configuration
- Using network firewall functions to block access to internal ports in specific network zones

If your setup does not permit isolating internal network communication, consider using encryption to protect the internal communication. For more information, see the section on securing internal communication.

System Replication Scenario

In a system replication scenario, you can protect the channels used in the following ways:

- Configuring SAP HANA to use exclusively a separate network dedicated to system replication for communication between primary and secondary site
- Configuring secure communication using the TLS/SSL protocol for encryption and mutual authentication between sites
- Specifying the IP addresses allowed to connect to system replication ports

Additional Measures for Securing Internal Communication

We recommend that you protect internal communication further by applying additional mechanisms. This may include filtering access to the relevant ports and channels by firewalls, implementing network separation, or applying additional protection at the network level (for example, VPN, IPSec).

We recommend routing the connection between the sites over a special site-to-site high-speed network, which typically already implements security measures such as separation from other network access and encryption or authentication between sites. The details of security measures and additional network security measures needed will depend on your specific environment. For more information about network administration, see the *SAP HANA Administration Guide*

SAP HANA Extended Application Services, Advanced Model

Security mechanisms are applied to protect the communication paths used by the SAP HANA XS advanced server infrastructure. SAP provides network topology recommendations to restrict access at the network level. For more information, see the section on SAP HANA XS advanced security.

Data Replication Technologies

Additional network configurations may be required depending on the implemented data replication technology. For more information, see the section on security for SAP HANA replication technologies.

Related Information

[Secure Internal Communication \[page 60\]](#)

[Security for SAP HANA Extended Application Services, Advanced Model \[page 293\]](#)

[Security for SAP HANA Replication Technologies \[page 290\]](#)

5.3 Securing Data Communication

SAP HANA supports encrypted communication for client-server (external) communication and internal communication.

Communication Channels That Can Be Secured

Communication between the following components can be secured using the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol.

External Channels

- SAP HANA and clients via ODBC or JDBC connections, including the SAP HANA cockpit and SAP HANA studio
- SAP HANA XS classic/advanced server and clients via HTTP

i Note

For JDBC and ODBC client connections, user passwords are always transmitted in encrypted hashed form during the user authentication process, never in plain text. For HTTP connections via SAP HANA XS classic, HTTPS must be configured. In SSO environments, we recommend using encrypted communication channels for **all** client connections.

- SAP HANA lifecycle manager and the SAP HANA cockpit and SAP HANA studio
- SAP HANA lifecycle manager and SAP Service Marketplace
- SAP HANA lifecycle manager and SAP Host Agent
- SAP HANA and an LDAP directory server
- SAP HANA and the Rserve server
- SAP HANA and data providers
- SAP HANA information composer and Web browser

i Note

SAP HANA information composer is supported on Intel-based hardware platforms only.

Internal Channels

- Internal database communication
- Internal communication between hosts in a distributed (multiple-host) SAP HANA system
- Internal communication between systems at the different sites in a system replication (high availability) scenario
- Internal communication between the SAP HANA database and server components, such as extended storage (SAP HANA dynamic tiering).

Trust and Key Stores for Securing Communication

Different trust and key stores exist for internal communication and external communication. Depending on your implementation, these will either be in the form of:

- In-database **certificate collections** (recommended)
- Personal security environments (**PSEs**) stored in the file system

A certificate collection (or PSE) is a secure location where the public information (public-key certificates) and private information (private keys) of the SAP HANA server are stored. A certificate collection may also contain the public information (public-key certificates) of trusted communication partners or root certificates from trusted Certification Authorities.

⚠ Caution

We recommend creating certificate collections for individual purposes in the database directly, rather than using trust stores (PSE) in the file system. By default, the same PSE in the file system is shared by all

databases for all external communication channels (including HTTP) and certificate-based authentication. Different PSEs must be explicitly configured for tenant databases.

Internal Communication

The keys and certificates in the certificate collection for internal communication are used to secure the following:

- Communication between database services
- Communication between hosts in a multiple-host system
- Communication between hosts and sites in a system replication scenario
- Communication between the SAP HANA database and additional server components, such as an extended storage server (SAP HANA dynamic tiering) or a streaming analytics server (SAP HANA Streaming Analytics).

External Communication

Certificates used for external communication (for example, JDBC and HTTP client access) are typically signed by an externally available Certification Authority (CA) because the CA certificates need to be integrated in the relevant clients.

For information about certificate handling, see the section on certificate management.

Related Information

External Communication

[Secure Communication Between SAP HANA and JDBC/ODBC Clients \[page 43\]](#)

[Secure Communication Between SAP HANA and an LDAP Directory Server \[page 57\]](#)

[Secure Communication Between SAP HANA XS Classic and HTTP Clients \[page 59\]](#)

[Security Aspects of SAP HANA Platform Lifecycle Management \[page 277\]](#)

[Security Aspects of SAP HANA R Integration \[page 281\]](#)

[Security for SAP HANA Replication Technologies \[page 290\]](#)

Internal Communication

[Secure Internal Communication \[page 60\]](#)

[Secure Internal Communication Between Sites in System Replication Scenarios \[page 65\]](#)

[Database Isolation \[page 24\]](#)

Certificate Collections/PSEs

[Certificate Management in SAP HANA \[page 261\]](#)

5.3.1 Secure Communication Between SAP HANA and JDBC/ODBC Clients

You can use the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol to secure communication between the SAP HANA database and clients that access the SQL interface of the database.

Enabling TLS/SSL for client-server communication provides the following by default:

- **Server certificate validation**
The database identifies itself to the client when the connection is established. This reduces the risk of man-in-the-middle attacks and fake servers gaining information from clients.
- **Data encryption**
In addition to server authentication, the data being transferred between the client and server is encrypted, which provides integrity and privacy protection. An eavesdropper cannot access or manipulate the data.

It is also possible to enable client certificate validation, if the identity of the client connecting to SAP HANA should be validated (mutual authentication).

TLS/SSL must be configured on both the SAP HANA database (server) and the client.

→ Remember

Secure communication between the SAP HANA database and HTTP clients (HTTPS) via the SAP HANA XS server classic must be configured separately. For more information, see *Secure Communication Between SAP HANA XS Classic and HTTP Clients*.

Enforced TLS/SSL for Client Connections

If you want to force all clients communicating with the SAP HANA database via the SQL interface to use a secured connection, you can set the parameter `sslEnforce` in the `communication` section of the `global.ini` configuration file to `true`. The database subsequently refuses SQL connection attempts that don't use SSL.

i Note

Communication properties are in the default configuration change blacklist (`multidb.ini`). This means that they cannot initially be changed in tenant databases. They must be changed from the system database. If appropriate for your scenario, you can remove these properties from the change blacklist. SAP HANA deployment scenarios are described in the *SAP HANA Master Guide*. For more information about how to edit the change blacklist, see the *SAP HANA Administration Guide*.

Related Information

[Secure Communication Between SAP HANA XS Classic and HTTP Clients \[page 59\]](#)

[Default Blacklisted System Properties in Tenant Databases \[page 361\]](#)

5.3.1.1 TLS/SSL Configuration on the SAP HANA Server

To use the Transport Layer Secure (TLS)/Secure Sockets Layer (SSL) protocol to secure communication between the SAP HANA database and clients that access the SQL interface of the database, TLS/SSL must be configured on both server and client side.

Before You Start

Before you can configure TLS/SSL on the SAP HANA database, the following general prerequisites must be met:

- The SAP Cryptographic Library CommonCryptoLib is available in the SAP HANA system. CommonCryptoLib (`libsapcrypto.so`) is installed by default as part of SAP HANA installation at `$DIR_EXECUTABLE`.

i Note

If you are using trust and key stores located in the file system instead of in the database, OpenSSL is also supported and installed by default as part of the operating system installation. However, OpenSSL is deprecated so you must migrate to CommonCryptoLib. For more information, see SAP Note 2093286.

- The SAP HANA database possesses a public and private key pair, and a public-key certificate. The TLS/SSL protocol uses public key technology to provide its protection. The server must possess a public and private key pair and a corresponding public-key certificate. It uses these to identify itself as the server component to a requesting client. All databases (system database and tenant databases) can have their own key pair and public key certificate. In distributed SAP HANA systems, every host must have its own key pair and public key certificate. You can use the tools provided with OpenSSL to create server certificates. If you are using CommonCryptoLib, you can also use the SAP Web Dispatcher administration tool or the SAPGENPSE tool, both of which are delivered with SAP HANA. For more information about the SAP Web Dispatcher administration tool, see SAP Note 2009483.

i Note

Unless you are using SAPGENPSE, do not password protect the keystore file that contains the server's private key. When using the SAP Web Dispatcher administration tool to create a personal security environment (PSE) for the server, do not specify a PIN. With SAPGENPSE, you can also set a PIN later using the `seclogin` command as follows:

```
sapgenpse seclogin -p <PSE path and file name> -x <PIN> -O <sid>adm
```

The PIN is stored in the credentials file `cred_v2` in the `$SECUDIR` directory.

⚠ Caution

If your server's keys are compromised, you must replace the key and the certificate.

Configuring TLS/SSL

The properties for configuring TLS/SSL on the server for external communication are available in the `communication` section of the `global.ini` configuration file.

In general, it's not necessary to configure any of the properties explicitly. The default configuration can be used.

i Note

Communication properties are in the default configuration change blacklist (`multidb.ini`). This means that they cannot initially be changed in tenant databases. They must be changed from the system database. If appropriate for your scenario, you can remove these properties from the change blacklist. SAP HANA deployment scenarios are described in the *SAP HANA Master Guide*. For more information about how to edit the change blacklist, see the *SAP HANA Administration Guide*.

However, you do need to create a certificate collection. You do this as follows:

1. Create a certificate collection in the database.
2. Add the server's public key certificate(s) and private key(s).
3. Add the public key certificates of trusted communication partners.
4. Assign the purpose *SSL/TLS* to the PSE.

→ Tip

For connections to **tenant databases**, certificate validation may not work due to how SAP HANA handles host name resolution. If this is the case, change the value of the parameter `[public_hostname_resolution] use_default_route` in the `global.ini` file on the SYSTEM layer from **ip** to **fqdn**. SAP HANA then uses the fully qualified domain name and certificate validation for secured JDBC/ODBC connections is allowed.

File System-Based Trust and Key Store

While we recommend using a certificate collection that exists in the database, it is possible to use a PSE located in the file system and configured in the `global.ini` file.

If files located in the file system are being used, they are shared by default. It is still possible to configure different trust and key stores for tenant databases for every database in the `global.ini` file. However, bear the following points in mind:

- If different trust and key stores are not explicitly configured for tenant databases, the same ones will be used for all external communication channels (including HTTP) for all databases.

⚠ Caution

If you have configured in tenant databases or the system database single sign-on mechanisms that rely on trust stores located in the file system (such as SAP logon and assertion tickets or SAML) and the trust stores are shared, users of one tenant database may be able to log on to other databases in the system.

- Only the system administrator can configure separate trust and key stores for tenant databases by changing the relevant properties in the `global.ini` file. This is because tenant database administrators are prevented from changing any communication properties. They are in the default configuration change blacklist (`multidb.ini`).

For more information about certificate collections in the database and PSEs in the file system, see the section on certificate management.

Related Information

[Server-Side TLS/SSL Configuration Properties for External Communication \(JDBC/ODBC\) \[page 46\]](#)

[Default Blacklisted System Properties in Tenant Databases \[page 361\]](#)

[Certificate Management in SAP HANA \[page 261\]](#)

Related SAP Notes

[SAP Note 2093286](#)

[SAP Note 1718944](#)

[SAP Note 1848999](#)

[SAP Note 2009483](#)

[SAP Note 2009878](#)

5.3.1.2 Server-Side TLS/SSL Configuration Properties for External Communication (JDBC/ODBC)

The parameters for configuring TLS/SSL for external communication on the SAP HANA server are available in the `communication` section of the `global.ini` configuration file.

The following table lists the configuration properties that can be used to configure TLS/SSL on the server. In general, it's not necessary to configure any of the parameters explicitly. The default configuration can be used.

i Note

Communication properties are in the default configuration change blacklist (`multidb.ini`). This means that they cannot initially be changed in tenant databases. They must be changed from the system database. If appropriate for your scenario, you can remove these properties from the change blacklist. SAP HANA deployment scenarios are described in the *SAP HANA Master Guide*. For more information about how to edit the change blacklist, see the *SAP HANA Administration Guide*.

| Parameter | Value | Default | Description |
|------------------------------------|--|---------|--|
| <code>sslMinProtocolVersion</code> | <code>{SSL30,TLS10,TLS11,TLS12}</code> | TLS10 | The minimum available TLS/SSL protocol version |

i Note

`sslMinProtocolVersion` must be less than or equal to `sslMaxProtocolVersion`.

| Parameter | Value | Default | Description |
|---|--------------------------------------|---------|--|
| <code>sslMaxProtocolVersion</code> | <code>{TLS10,TLS11,TLS12,MAX}</code> | MAX | The maximum available TLS/SSL protocol version |
| <code>sslValidateCertificate</code> | <Boolean value> | false | If set to true , the certificate of the communication partner is validated. |
| <code>sslCreateSelfSignedCertificate</code> | <Boolean value> | false | If set to true , a self-signed certificate is created if the keystore cannot be found. |
| <code>sslBlindCAResponse</code> | <Boolean value> | off | If set to on , a client may send a certificate in response to a client certificate request from the server even if it contains an empty Certificate Authority list By default, the client cannot respond to such a certificate request; the connection is refused. |

Additionally, the parameter `sslCipherSuites` can be used to specify the encryption algorithms available for TLS/SSL connections. Its value depends on the cryptographic service provider used. The default values are **PFS:HIGH::EC_HIGH:+EC_OPT** (CommonCryptoLib) and **ALL:!ADH:!LOW:!EXP:NULL:@STRENGTH** (OpenSSL*). For more information, see the documentation of the cryptographic library.

Parameters for Configuring Trust and Key Stores in the File System

The following parameters are used to configure trust and key stores located in the file system. In general, it's not necessary to configure a cryptographic provider nor any of the parameters explicitly. The default configuration can be used.

Caution

We recommend creating certificate collections for individual purposes in the database directly, rather than using trust stores (PSE) in the file system. By default, the same PSE in the file system is shared by all databases for all external communication channels (including HTTP) and certificate-based authentication. Different PSEs must be explicitly configured for tenant databases.

In general, if certificate collections with the purpose *SAML*, *X509*, *JWT*, or *SSL/TLS* exist in the database, the parameters below are ignored. If you explicitly want the in-database collection for one of these purposes to be ignored, configure the parameter `skip_in_memory_pse_store_for_purposes`.

| Parameter | Value | Default | Description |
|-------------------|--------------------------------------|--|---|
| sslCryptoProvider | {commoncrypto sapcrypto openssl} | <ol style="list-style-type: none"> 1. commoncrypto 2. openssl* | <p>Cryptographic provider used for TLS/SSL connection</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If you specify a value for this parameter, you must also explicitly specify paths in both the <code>sslKeyStore</code> and <code>sslTrustStore</code> parameters to avoid configuration issues.</p> </div> |
| sslKeyStore | file | <ul style="list-style-type: none"> • <code>\$SECUDIR/sapsrv.pse</code> (CommonCryptoLib) • <code>\$HOME/.ssl/key.pem</code> (OpenSSL*) | <p>Path to the keystore file that contains the server's private key</p> <p>You must specify an absolute path to the keystore file if using OpenSSL*.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If you specify a value for this parameter, you must also explicitly specify a cryptographic provider in the <code>sslCryptoProvider</code> parameter to avoid configuration issues.</p> </div> |
| sslTrustStore | file | <ul style="list-style-type: none"> • <code>\$SECUDIR/sapsrv.pse</code> (CommonCryptoLib) • <code>\$HOME/.ssl/trust.pem</code> (OpenSSL*) | <p>Path to trust store file that contains the server's public certificate</p> <p>You must specify an absolute path to the keystore file if using OpenSSL*.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If you specify a value for this parameter, you must also explicitly specify a cryptographic provider in the <code>sslCryptoProvider</code> parameter.</p> </div> |

| Parameter | Value | Default | Description |
|---------------------------------------|---------------------------------|---------|--|
| skip_in_memory_pse_store_for_purposes | {SSL,SAML,SAP LOGON, X509, JWT} | empty | <p>Purposes that exclusively use a trust and key store (PSE) located in the file system.</p> <p>For each purpose listed, the in-memory certificate collection is skipped and a trust and key store in the file system is used instead.</p> <p>For purposes SSL, SAML, X509, and JWT the trust and key store specified in <code>sslTrustStore</code> and <code>sslKeyStore</code> are used instead.</p> |

i Note

The PSE for SAP logon and assertion tickets can be specified with the parameter `[authentication] saplogontickettruststore` in the `indexserver.ini` file (default `saplogon.pse`).

i Note

*OpenSSL is deprecated. If you are using OpenSSL, migrate to CommonCryptoLib. For more information, see SAP Note 2093286.

Related Information

[Default Blacklisted System Properties in Tenant Databases \[page 361\]](#)

[Certificate Management in SAP HANA \[page 261\]](#)

[Single Sign-On Using SAP Logon and Assertion Tickets \[page 111\]](#)

[SAP Note 2093286](#)

5.3.1.3 TLS/SSL Configuration on the Client

You can use the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol to secure communication between the SAP HANA database and clients that access the SQL interface of the database. TLS/SSL must be configured on both the server and the client.

The client-side configuration required to secure client-to-server communication depends on whether the client communicates with the server via an ODBC-based or a JDBC-based connection.

ODBC

For ODBC-based connections, the configuration properties and their names are the same as the server parameters. In addition, the `encrypt` property is available to initiate an TLS/SSL-secured connection. You set the properties according to the client operating system.

i Note

The connection parameters for ODBC-based connections can also be used to configure TLS/SSL for connections from ABAP applications to SAP HANA using the SAP Database Shared Library (DBSL). To pass the connection parameters to the DBSL, use the following profile parameter:

```
dbshdb/connect_property = param1, param2, ..., paramN
```

The connection parameters are used for both the primary ABAP connection and secondary connections.

JDBC

For clients connecting via the JDBC interface, TLS/SSL is configured at the Java virtual machine (JVM) level using system properties. There are several ways of configuring these properties. For more information, see the Java Platform documentation.

SAP HANA Studio

For connections from the SAP HANA studio, which uses the JDBC interface, you configure the TLS/SSL properties directly in the system's properties in the SAP HANA studio (for example, while adding it in the studio).

SAP HANA Cockpit and Database Explorer

Connections from the SAP HANA cockpit and SAP HANA database explorer also use JDBC. Secure communication can be enabled and configured when you register the database as a resource in the cockpit or database explorer. The server certificate or root certificate must be imported as trusted certificates into the cockpit or database explorer.

Related Information

[Configure TLS/SSL for SAP HANA Studio Connections \[page 55\]](#)

[Security Aspects of SAP HANA Cockpit \[page 282\]](#)

[Securing the SAP HANA Database Explorer \[page 286\]](#)

5.3.1.4 Client-Side TLS/SSL Connection Properties (ODBC)

For ODBC-based connections, the configuration properties and their names are the same as the server parameters with the addition of the `encrypt` property, which initiates a TLS/SSL-secured connection.

The following table lists the configuration properties that are used to configure SSL for ODBC client access.

For more information about other ODBC connection properties and how to set them, see the *SAP HANA Client Interfaces Programming Reference*.

| Property | Value | Default | Description |
|--------------------------------|---|---|--|
| <code>ENCRYPT</code> | boolean | False | Enables or disables TLS 1.1 – TLS1.2 encryption. The server choose the highest available. |
| <code>sslCryptoProvider</code> | { commoncrypto sapcrypto openssl mscrypto } | <ol style="list-style-type: none">1. commoncrypto or sapcrypto (if installed)2. openssl/mscrypto | <p>Specifies the cryptographic library provider used for SSL communication. If you specify a value for this property, then you must also explicitly specify paths in both the <code>sslKeyStore</code> and <code>sslTrustStore</code> properties to avoid configuration issues.</p> <p>If CommonCryptoLib is not available, OpenSSL is used by default in Linux environments and msCrypto in Windows environments.</p> |

i Note

CommonCryptoLib must be manually installed on the host where the SAP HANA client is installed. For more information, see the *SAP HANA Client Installation and Update Guide*.

| Property | Value | Default | Description |
|--------------------------|--------------|---|---|
| sslHostNameInCertificate | string value | empty | <p>Specifies the host name used to verify server's identity.</p> <p>The host name specified here verifies the identity of the server instead of the host name with which the connection was established.</p> <p>For example, in a single-host system, if a connection is established from a client on the same host as the server, then a mismatch would arise between the host named in the certificate (actual host name) and the host used to establish the connection (localhost).</p> <p>If you specify * as the host name, then the server's host name is not validated. Other wildcards are not permitted.</p> |
| sslKeyStore | <file> | <ul style="list-style-type: none"> \$SECUDIR/sapcli.pse (CommonCryptoLib) \$HOME/.ssl/key.pem(OpenSSL) MY (msCrypto) | <p>Specifies the path to the keystore file that contains the client's private key and, if using CommonCryptoLib, the server's public certificates.</p> <p>If using CommonCryptoLib, you use the SAPGENPSE tool (installed with CommonCryptoLib) to create the client PSE (sapcli.pse) and generate the client's keys. You must also import the server's public certificates into sapcli.pse. Typically, this is the root certificate or the certificate of the certification authority that signed the server's public certificates. See SAP Note 1718944.</p> <p>If using OpenSSL, use OpenSSL tools to create the required keystore file.</p> <div style="border: 1px solid #0070c0; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If you specify a value for this property, then you must also explicitly specify a cryptographic provider in the sslCryptoProvider property to avoid configuration issues.</p> </div> |

| Property | Value | Default | Description |
|------------------------|---------|--|---|
| sslTrustStore | <file> | \$HOME/.ssl/trust.pem (OpenSSL) sapcli.pse (Common-CryptoLib) | Specifies the path to a trust store file that contains the server's public certificates if using OpenSSL. Typically, the trust store contains the root certificate or the certificate of the certification authority that signed the server's public certificates. If you specify a value for this property, then you must also explicitly specify a cryptographic provider in the <code>sslCryptoProvider</code> property to avoid configuration issues. If you are using the cryptographic library <code>CommonCryptoLib</code> or <code>msCrypto</code> , then leave this property empty. |
| sslValidateCertificate | boolean | true | Specifies whether to validate the server's certificate. |

Related Information

[SAP Note 1718944](#)

5.3.1.5 Client-Side TLS/SSL Connection Properties (JDBC)

For clients connecting via the JDBC interface, TLS/SSL is configured using connection properties.

The following table lists the connection properties that can be used to configure TLS/SSL for JDBC client access.

For more information about other JDBC connection properties and how to set them, see the *SAP HANA Client Interfaces Programming Reference*.

| Property | Value | Default | Description |
|----------|---------|---------|--|
| encrypt | boolean | false | Enables or disables TLS/SSL encryption |

| Property | Value | Default | Description |
|-----------------------|-----------------------|--------------|--|
| hostNameInCertificate | string value | empty | <p>Specifies the host name used to verify server's identity.</p> <p>The host name specified here is used to verify the identity of the server instead of the host name with which the connection was established.</p> <p>For example, in a single-host system, if a connection is established from a client on the same host as the server, a mismatch would arise between the host named in the certificate (actual host name) and the host used to establish the connection (localhost).</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If you specify * as the host name, this property has no effect. Other wildcards are not permitted.</p> </div> |
| keyStore | <file> <store name> | <VM default> | Specifies the location of the Java key-store. |
| keyStorePassword | <password> | <VM default> | Specifies the password used to access the private key from the keystore file. |
| | | | <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>This property is not used for SAP HANA studio connections.</p> </div> |
| keyStoreType | <JKS> <PKCS12> | <VM default> | Specifies the Java keystore file format. |
| sniHostname | <string> | empty | Specifies the name of the host that is attempting to connect at the start of the TLS handshaking process. |
| trustStore | <file> <store name> | <VM default> | <p>Specifies the path to the trust store file that contains the server's public certificate(s).</p> <p>Typically, the trust store contains the root certificate or the certificate of the certification authority that signed the server's certificate(s).</p> |
| trustStorePassword | <password> | <VM default> | Specifies the password used to access the trust store file. |

5.3.1.6 Configure TLS/SSL for SAP HANA Studio Connections

Secure communication between the SAP HANA studio and the SAP HANA database using the Transport Security Layer (TLS)/Secure Sockets Layer (SSL) protocol.

Prerequisites

- You have configured the SAP HANA database for secure client-server communication over JDBC/ODBC. For more information, see *SSL Configuration on the SAP HANA Server* in the *SAP HANA Security Guide*.
- You have added the SAP HANA system in the SAP HANA studio.

Context

The SAP HANA studio communicates with the SAP HANA database via the JDBC client interface. The client-side configuration of the SAP HANA studio uses Java TLS/SSL properties.

Procedure

1. Using the keytool command line tool, import the truststore file that contains the server root certificate into either the Java keystore or your personal user keystore.

By default, the SAP HANA studio client validates server certificate(s) against the root certificate stored in the Java keystore of the running VM (virtual machine). This keystore is part of the Java installation and is located in the Java home directory under `${JAVA_HOME}/lib/security/cacerts` (Linux) or `%JAVA_HOME%/lib/security/cacerts` (Windows).

However, it is not recommended that you store the root certificate in this keystore, but in your personal user keystore instead. The user keystore is located in the home directory of the current operating system user. The file name is `.keystore`.

2. Enable and configure TLS/SSL secure communication between the SAP HANA studio and the server:

In the SAP HANA studio, open the system's properties and choose *Connect Using SSL*.

This corresponds to setting the Java SSL property `encrypt` to **true**.

3. Configure how the identity of the server is to be validated during connection (server-side authentication):
 - a. In the system's properties dialog, choose the *Additional Properties* tab.
 - b. If you want server certificate(s) to be validated using the default truststore, choose *Validate SSL Certificate*.

This corresponds to setting the Java SSL property `validateCertificate` to **true**.

When an TLS/SSL connection is established, the host name in the certificate being connected to and the host name in the server certificate must match. This may not always be the case. For example, in a

single-host system, if a connection is established from the SAP HANA studio on the same host as the SAP HANA server, a mismatch would arise between the host named in the certificate (fully qualified host name) and the host used to establish the connection (localhost)*.

You can override the host name specified in the server certificate by entering a host name with a defined certificate in the *Override Host Name Certificate* field. This corresponds to setting the Java SSL property `hostNameInCertificate`.

- c. If you want the server certificate to be validated using the user's keystore and not the default Java keystore, choose *Use user keystore as trust store*.

This corresponds to changing the value of the Java SSL property `trustStore`.

i Note

If you do not have a working public key infrastructure (PKI), you can also suppress server certificate validation entirely by selecting neither of these options (*Validate SSL Certificate* or *Use user keystore as trust store*). However, this is not recommended.


4. Optional: If the identity of the client is to be validated by the SAP HANA server (client certificate validation), perform the following additional steps:
 - a. In the *Additional Properties* tab of the system properties, specify the path to the user keystore that contains your private key, as well as the pass phrase required to access this file.
 - b. Enable validation of the client's identity on the server by changing the parameter `[communication] sslValidateCertificate` in the `global.ini` file to **true**.

You can do this on the *Configuration* tab of the Administration editor.

- c. Import the client root certificate into the server truststore used for client-server communication.

If you manage client certificates directly in the database (recommended), this means importing the certificate into the certificate store and adding it to the certificate collection with the purpose *SSL*.

Results

In the *Systems* view, a lock icon appears next to the system name () , indicating that SSL communication is active.

Related Information

[TLS/SSL Configuration on the SAP HANA Server \[page 44\]](#)

5.3.1.7 SOCKS Proxy Communication Protocol

SOCKS proxy is a communication protocol that uses a proxy server to exchange network packets between a client and server.

SOCKS proxy is an insecure network communication protocol.

Connect to a SOCKS proxy server by using either SAP HANA HDBSQL or the ODBC driver.

SAP HANA hdbsql Specify the `-proxyhost <hostname>` option. You can also specify additional, optional SOCKS proxy hdbsql options.

ODBC driver Specify the `PROXY_HOST <hostname>` connection property. You can also specify additional, optional SOCKS proxy ODBC connection properties.

5.3.2 Secure Communication Between SAP HANA and an LDAP Directory Server

You can use the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol or Secure LDAP protocol (LDAPS) to secure communication between SAP HANA and an LDAP directory server.

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing directory services. If you use an LDAP-compliant directory server to manage users and their access to resources, you can leverage LDAP-based authentication to access SAP HANA and LDAP group membership to authorize users. With LDAP-based authentication, users can also be automatically provisioned in SAP HANA. For more information, see the sections on LDAP authentication and LDAP group authorization.

Communication between SAP HANA and the LDAP server can be secured using the TLS/SSL protocol. The trust store used to authenticate communication must be a certificate collection in the SAP HANA database with the purpose *LDAP*. The certificate of the Certificate Authority (CA) that signed the certificate used by the LDAP server must be available in this certificate collection. For more information, see the section on certificate management.

⚠ Caution

If the LDAP server is being used for user authentication, communication between SAP HANA and the LDAP server must be secured to protect the transmission of user passwords.

i Note

TLS/SSL-secured communication uses the SAP cryptographic library, CommonCryptoLib. For more information, see SAP Note 1848999.

Properties for configuring communication with the LDAP server (including TLS/SSL configuration) are available in the `ldap` section of the `global.ini` configuration file. For more information, see the section on LDAP configuration properties.

Connections to the LDAP server may also be secured using Secure LDAP protocol (LDAPS).

Related Information

[LDAP Group Authorization \[page 180\]](#)

[LDAP User Authentication \[page 114\]](#)

[Communication Configuration Properties for LDAP \[page 58\]](#)

[SAP Note 1848999](#)

[SAP Note 2438641](#)

5.3.2.1 Communication Configuration Properties for LDAP

Properties for configuring LDAP communication

Communication

The parameters for configuring communication between the SAP HANA database and an LDAP server are available in the `ldap` section of the `global.ini` configuration file. In general, it's not necessary to configure any of the following properties explicitly. The default configuration can be used.

TLS/SSL Properties

| Parameter | Value | Default | Description |
|------------------------------------|--------------------------------------|-------------------------------|---|
| <code>sslMinProtocolVersion</code> | <code>{SSL30,TLS10}</code> | TLS10 | The minimum available TLS/SSL protocol version |
| <code>sslMaxProtocolVersion</code> | <code>{TLS10,TLS11,TLS12,MAX}</code> | MAX | The maximum available TLS/SSL protocol version |
| | | | i Note The MAX value is internally mapped to TLS12. |
| <code>sslCipherSuites</code> | String value | PFS:HIGH::EC_HIGH: +EC_OPT | The encryption algorithms available for TLS/SSL connections |
| | | | i Note CommonCryptoLib is used. |

Other Communication Properties

| Parameter | Value | Default | Description |
|-----------|--------------------|---------|---|
| timeout | Timeout in seconds | 0 | Timeout for LDAP operations in seconds The default value of 0 means there is no timeout. |

Note

SAP HANA does not support referral chasing and aliases are never dereferenced.

5.3.3 Secure Communication Between SAP HANA XS Classic and HTTP Clients

You can use the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol to secure communication between the SAP HANA XS classic server and HTTP clients.

The SAP HANA XS classic server allows Web-based applications to access SAP HANA via HTTP. The internal Web Dispatcher of the SAP HANA system manages these incoming HTTP requests.

Therefore, to secure communication between the SAP HANA system and HTTP clients, you must configure the internal SAP Web Dispatcher to use TLS/SSL for inbound application requests. You can do this using the SAP HANA Web Dispatcher Administration tool.

A per-database configuration of TLS/SSL keys and certificates is possible. It is also possible to configure HTTPS on the basis of a single "wildcard" server certificate that covers all databases.

Caution

Do not use a wildcard server certificate if strict isolation between tenant databases is required. If authentication relies on a wildcard certificate and a shared trust store, users of one tenant database will be able to log on to other databases in the system.

For more information, see the *SAP HANA Administration Guide*.

Related Information

[Network and Communication Security with SAP HANA XS Advanced \[page 328\]](#)

5.3.3.1 HTTP Access Log

To monitor all HTTP(s) requests processed in an SAP HANA system, you can set up the internal Web Dispatcher to write a standardized HTTP log for each request.

To configure the Web Dispatcher to log all HTTP(s) requests, you add the property `icm/http/logging_0` to the `[profile]` section of the `webdispatcher.ini` configuration file, specifying the following value:

```
PREFIX=/, LOGFILE=$(DIR_INSTANCE)/trace/access_log-%y-%m-%d, MAXSIZEKB=10000,  
SWITCHTF=day, LOGFORMAT=SAP
```

This will generate access log files in the following directory: `/usr/sap/<sid>/HDB<instance>/<host>/trace/access_log-<timestamp>`.

❖ Example

```
Sample log file entry: [26/Nov/2014:13:42:04 +0200] 10.18.209.126 BOB - "GET /sap/xse/  
test/InsertComment.xsjs HTTP/1.1" 200 5 245
```

The last three numbers are the HTTP response code, the response time in milliseconds, and the size in bytes. For more information about logging and alternative log formats, see the Internet Communication Manager (ICM) documentation on SAP Help Portal.

You can configure the `webdispatcher.ini` configuration file and view log files in the SAP HANA cockpit.

5.3.4 Secure Internal Communication

All internal SAP HANA communication can be secured using the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol. A simple public-key infrastructure (PKI) is set up during installation for this purpose.

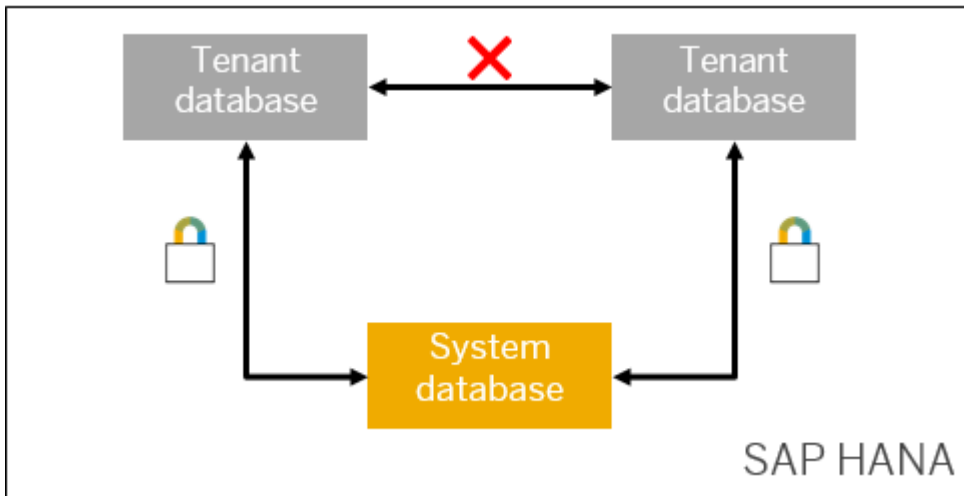
Internal Communication Channels

The internal communication channels shown below can be secured using TLS/SSL.

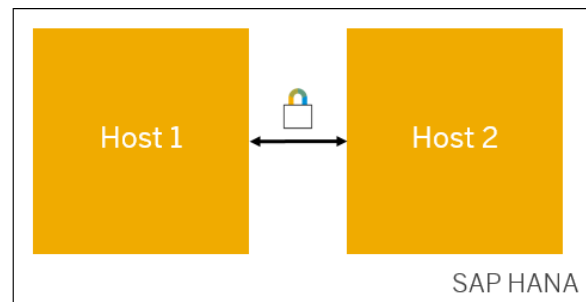
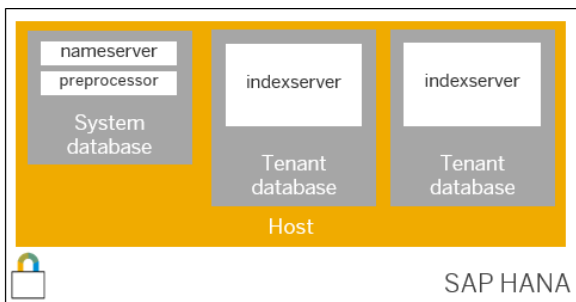
i Note

SAP HANA internal communication has sometimes been unofficially referred to as TREXNet communication. However, the term TREXNet is not valid in the context of SAP HANA.

Communication between databases



Communication between the hosts in a multiple-host system and between processes on a single host

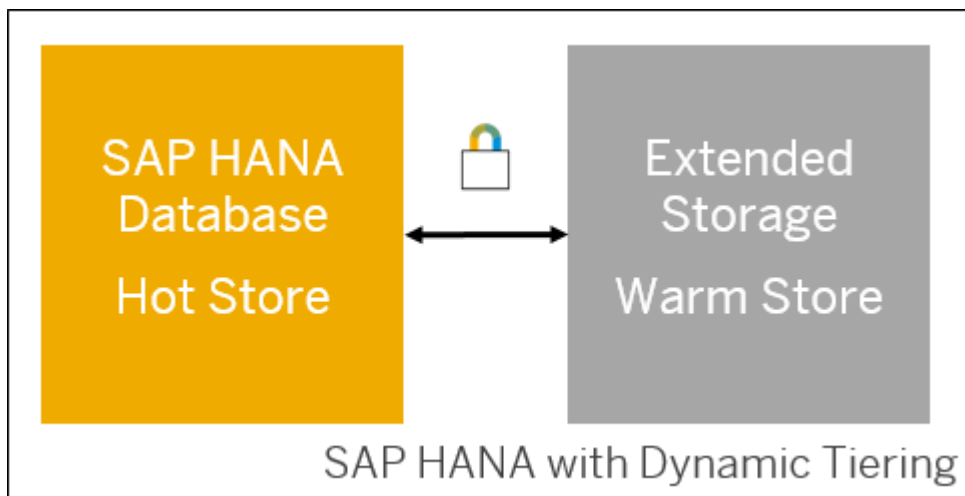


Communication between the sites in a system with system replication enabled



Communication between the SAP HANA database and additional server components

For example: Extended storage server (SAP HANA dynamic tiering) or a streaming analytics server (SAP HANA Streaming Analytics).



System Public Key Infrastructure

A dedicated PKI is created for internal communication automatically during system installation. Every host on which a database server and optional component server is running, as well as every tenant database in the system, are integrated into this PKI, which uses CommonCryptoLib as the cryptographic library.

Each host and database receive a public and private key pair and a public-key certificate for mutual authentication. These certificates are all signed by a dedicated trusted certificate authority (CA) that is unique to the SAP HANA instance. The root personal security environment (PSE) file is stored in the system PKI SSFS (secure store in the file system). All other PSEs are encrypted with an automatically generated random PIN and stored in the file system. Certificates are automatically renewed when they expire.

i Note

A unique master key that protects the system PKI SSFS is generated during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change it immediately after handover to ensure that it is not known outside of your organization. For more information about how to change the SSFS master keys, see the *SAP HANA Administration Guide*.

To secure internal communication between hosts and sites, you can set up and configure your own PKI, but we recommend you use the system PKI. The system PKI is always used to secure communication within tenant databases and communication with optional server components.

i Note

If high isolation is configured for tenant databases, the system PKI **must** also be used to secure communication between hosts.

For more information about migrating to the system PKI from a manually configured PKI, see SAP Note 2175672.

TLS/SSL Configuration Using System PKI

No interaction is required to set up the system PKI, but you may need to explicitly enable TLS/SSL depending on the channel as follows:

| Communication Channel | Configuration Required to Enable TLS/SSL |
|---|--|
| Communication between the processes of individual databases | <p>Configure the system for high isolation.</p> <p>High isolation requires that the processes of individual databases run under dedicated operating system (OS) users in dedicated OS groups. In addition, it enables certificate-based authentication so that only the processes belonging to the same database can communicate with each other.</p> <p>If you also want data communication within databases to be encrypted, you must change the value of the property <code>[communication] ssl</code> in the <code>global.ini</code> from off to systemPKI. If the property <code>ssl</code> is not visible (for example, in the SAP HANA studio), add the key ssl with the value systemPKI to the section <code>communication</code>.</p> <div data-bbox="603 925 1396 1070"><p>→ Remember</p><p>Change (or add) the property in the system database in the <code>SYSTEM</code> layer of the configuration file.</p></div> <div data-bbox="603 1084 1396 1223"><p>i Note</p><p>If cross-database access is enabled, communication between configured tenant databases is allowed.</p></div> <p>For more information about how to configure a system for high isolation, see the <i>SAP HANA Administration Guide</i>.</p> |

| Communication Channel | Configuration Required to Enable TLS/SSL |
|---|--|
| Communication between hosts in a multiple-host system and localhost communication | <p>Enable TLS/SSL manually.</p> <p>In the <code>global.ini</code> configuration file, change the value of the property <code>[communication] ssl</code> to systemPKI.</p> <p>This configuration ensures that only hosts belonging to the same system can communicate with each other and that all data communication between hosts is encrypted.</p> <div style="border: 1px solid #0070C0; padding: 5px; margin: 10px 0;"> <p>i Note</p> <p>In a system that is not configured for high isolation, you can still enable secure communication between hosts. Remember you change the property in the system database in the <code>SYSTEM</code> layer.</p> </div> <p>Enabling secure communication between hosts automatically enables secure communication between processes on the same host without any further configuration. Note the following:</p> <ul style="list-style-type: none"> • If you are operating a single-host and require secure localhost communication, you must still enable TLS/SSL for inter-host communication as described above. • If you have enabled TLS/SSL for inter-host communication as described above, but do not require secure localhost communication, you can change the value of the property <code>[communication] ssl_local</code> from on to off. |
| Communication between sites in a system with system replication enabled | Several steps are required to enable TLS/SSL for the communication channel used for system replication. For more information, see <i>Secure Internal Communication Between Sites in System Replication Scenarios</i> . |
| Communication between the SAP HANA database and additional server components | <p>No configuration required</p> <p>TLS/SSL is automatically enabled and cannot be disabled.</p> |

Related Information

[Secure Stores in the File System \(SSFS\) \[page 220\]](#)

[Database Isolation \[page 24\]](#)

[Server-Side TLS/SSL Configuration Properties for Internal Communication \[page 66\]](#)

[Secure Internal Communication Between Sites in System Replication Scenarios \[page 65\]](#)

[Legacy Configuration of Secure Internal Communication \[page 68\]](#)

[SAP Note 2175672](#)

5.3.4.1 Secure Internal Communication Between Sites in System Replication Scenarios

Communication between sites in a system replication scenario is always authenticated. In addition, it is possible to secure internal network communication between primary and secondary sites using TLS/SSL.

System replication is a mechanism for ensuring the high availability of SAP HANA systems, as well as disaster recovery. Through the continuous replication of data from a primary to a secondary system (or systems), including in-memory loading, system replication facilitates rapid failover in the event of a disaster. Production operations can be resumed with minimal downtime.

Communication between the sites in a system replication landscape must be secured. The system PKI (public key infrastructure) that is automatically created during system installation is the default and recommended mechanism for communication. No interaction is required to set up the system PKI. However, you can also set up and configure your own PKI (see *Legacy Configuration of Secure Internal Communication*).

→ Remember

System replication is configured for the system as a whole. This means that the system database and all tenant databases are part of the system replication.

Configuring Authentication Between Sites

To ensure that only configured systems in a system replication landscape can communicate with each other, SAP HANA uses certificate-based authentication based on the system PKI. To establish trust between systems, you must copy the system PKI SSFS data file and key file from the primary system to the same location on the secondary system(s). These files can be found at the following locations:

- `$DIR_INSTANCE/./SYS/global/security/rsecssfs/data/SSFS_<SID>.DAT`
- `$DIR_INSTANCE/./SYS/global/security/rsecssfs/key/SSFS_<SID>.KEY`

Copy the files before you register the secondary system with the primary system.

Configuring TLS/SSL-Secured Communication

In addition to authenticated communication, it is also possible to secure the following communication channels between primary and secondary systems using TLS/SSL:

- Metadata channel used to transmit metadata (for example, topology information) between the sites
- Data channel used to transmit data between the sites

For more information on how to enable TLS/SSL on these communication channels, see the *SAP HANA Administration Guide*.

i Note

On SAP HANA systems with dynamic tiering, the same configuration applies. No additional steps are required. However, before you configure communication for dynamic tiering, see SAP Note 2356851.

Be aware that you need additional licenses for SAP HANA options and capabilities. For more information, see [Important Disclaimer for Features in SAP HANA Platform \[page 382\]](#).

Additional Network Security

You can further secure communication between sites by configuring SAP HANA to use exclusively a separate network dedicated to system replication for communication between primary and secondary sites.

It is also recommended that you configure the IP addresses of those hosts that are allowed to connect to the ports required for system replication.

For more information, see also the section on network security and the section on host name resolution in the *SAP HANA Administration Guide*.

Related Information

[Legacy Configuration of Secure Internal Communication \[page 68\]](#)

[Server-Side TLS/SSL Configuration Properties for Internal Communication \[page 66\]](#)

[Network Security \[page 37\]](#)

[Secure Internal Communication \[page 60\]](#)

[SAP Note 2356851](#)

5.3.4.2 Server-Side TLS/SSL Configuration Properties for Internal Communication

The properties for configuring TLS/SSL for internal SAP HANA communication are available in the `communication` section of the `global.ini` configuration file.

The following properties are available for configuring TLS/SSL for internal communication.

i Note

Only the system administrator can configure these properties. This is because tenant database administrators are prevented from changing any communication properties. They are in the default configuration change blacklist (`multidb.ini`).

| Property | Value | Default | Description |
|-----------|------------------------|---------|--|
| ssl | {on systemPKI off} | off | <p>Enables TLS/SSL on internal communication channels</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> off (default) With this value, TLS/SSL is disabled. In systems configured for high isolation, but with the default value off, host and database authentication is enabled but internal communication is not encrypted. systemPKI This value enables the use of the system PKI for secure communication between hosts (host authentication and encrypted data communication). This value additionally enables encrypted data communication within databases. For more information, see <i>Secure Internal Communication</i>. If you have installed the new runtime environment for application development, SAP HANA Extended Application Services (XS) Advanced Model, the value systemPKI is set automatically during installation. on This value enables the use of a manually configured PKI for secure communication between hosts. Additional properties for trust and key stores apply in this case. For more information, see <i>Legacy Configuration of Secure Internal Communication</i>. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>In a system configured for high isolation, do not set the value of this property to on. This will result in an error.</p> </div> <p>To change the default value of this property, you must first add it to the <code>communication</code> section of the <code>global.ini</code> file.</p> |
| ssl_local | <Boolean value> | on | <p>Enables TLS/SSL for communication between localhost processes</p> <p>This parameter is only evaluated if <code>ssl</code> has the value systemPKI or on.</p> |

| Property | Value | Default | Description |
|--------------------------------|-----------------|---------|--|
| sslInternalValidateCertificate | <Boolean value> | true | <p>If true, the certificate of the communication partner is validated.</p> <p>In a system configured for high isolation, this parameter is ignored. The certificate of the communication partner is always validated.</p> |

Related Information

[Secure Internal Communication \[page 60\]](#)

[Legacy Configuration of Secure Internal Communication \[page 68\]](#)

[Default Blacklisted System Properties in Tenant Databases \[page 361\]](#)

5.3.4.3 Legacy Configuration of Secure Internal Communication

Although it is recommended that you use the system PKI (public key infrastructure) that is automatically created during system installation to secure internal communication channels, you can set up and configure your own PKI. This manually configured PKI is also used if system replication is configured for the system.

TLS/SSL Configuration for Communication Between Hosts

Since a host can both initiate a connection with another host (client role) as well as be the target of a connection initiated by another host (server role), every host in the system requires a public and private key pair, and a public-key certificate (server certificate) with which it can identify itself to other hosts. Each host also needs the certificate or certificates with which it can validate the identity of other hosts. Typically, this is the root certificate or the certificate of the certification authority (CA) that signed the other hosts' certificates.

Note

SAP HANA dynamic tiering does not support legacy configuration (using a manually configured PKI). If you are using SAP HANA dynamic tiering, use the system PKI configuration.

Use CommonCryptoLib as the cryptographic library. It is installed by default as part of SAP HANA server installation.

To manually configure secure communication between hosts:

1. Create a CA for the SAP HANA installation using external tools, for example, the OpenSSL command line tool.
We recommend that you use a dedicated Certification Authority (CA) to sign all certificates used. We recommend storing your CA certificate in `$DIR_INSTANCE/ca`. This is typically the root certificate.

→ Recommendation

Create one private CA for each SAP HANA host. Do not use public CA for securing internal SAP HANA communication.

2. On every host, create the required server certificates.
Every host is verified with its fully qualified domain name (FQDN). The common name (CN) must be the FQDN of the host you get by reverse DNS look-up. The other fields describe your organization.
3. Sign the certificates with the CA.
4. On every host, create a local keystore named `sapsrv_internal.pse` in directory `$SECUDIR` and import the private key and certificate, and the CA certificate (or root certificate).
In the `communication` section of the file `global.ini`, create the property `ssl` with the value `on`.

TLS/SSL Configuration for Cross-Site Communication in System Replication Scenarios

In a system with system replication enabled, communication between sites (metadata and data channels) can be secured using the same configuration described above. For the data communication, you also need to enable SSL with the property `[system_replication_communication] enable_ssl` in the `global.ini` configuration file. For more information, see *Secure Internal Communication Between Sites in System Replication Scenarios*.

Keystore Configuration

The `[communication] sslInternalKeyStore` parameter in the `global.ini` configuration file specifies the path to the keystore file that contains the certificates for the following internal communication channels:

- Communication between hosts
- Communication between sites in system replication scenarios (data communication channel).

The default value is `$SECUDIR/sapsrv_internal.pse`.

Related Information

[Secure Internal Communication Between Sites in System Replication Scenarios \[page 65\]](#)

6 SAP HANA User Management

SAP HANA database users may be technical users or correspond to real end users. Several tools are available for user management.

Every user who wants to work directly with the SAP HANA database must have a database user with the necessary privileges. Depending on the scenario, the user accessing SAP HANA may either be a technical system user or an individual end user.

After successful logon, the user's authorization to perform the requested operations on the requested objects is verified. This is determined by the privileges that the user has been granted. Privileges can be granted to database users either directly, or indirectly through roles. Several tools are available for provisioning and managing users. For more information about the authorization model of the SAP HANA database, see the section on authorization.

[User Types \[page 71\]](#)

It is often necessary to specify different security policies for different types of database user. In the SAP HANA database, we differentiate between database users that correspond to real people and technical database users.

[User Groups \[page 73\]](#)

User groups allow you to manage related users together. Group administrators can be assigned to manage individual user groups exclusively and independently of each other.

[User Administration Tools \[page 80\]](#)

Depending on your organization and its user provisioning strategy, people with different job functions may be involved in the process of user administration. Different tools are used for different tasks.

[Predefined Users \[page 83\]](#)

A number of predefined users are required for installing, upgrading, and operating the SAP HANA database.

[Deactivate the SYSTEM User \[page 90\]](#)

As the most powerful database user, `SYSTEM` is not intended for use in production systems. Use it to create lesser privileged users for particular purposes and then deactivate it.

Related Information

[SAP HANA Authentication and Single Sign-On \[page 92\]](#)

[SAP HANA Authorization \[page 119\]](#)

6.1 User Types

It is often necessary to specify different security policies for different types of database user. In the SAP HANA database, we differentiate between database users that correspond to real people and technical database users.

Technically, database users that correspond to real people and technical database users are the same. The only difference between them is conceptual.

Database Users that Correspond to Real People

Every person who needs to work with SAP HANA must have a database user. Depending on your system configuration and scenario, database users can be created by:

- User administrators
- User group administrators
- An LDAP provider used for user authentication and enabled for automatic user creation

i Note

In SAP HANA, the user is technically created by the user `SYS` (see `CREATOR` column in system view `USERS`).

Database users that correspond to real people are dropped when the person leaves the organization. This means that any database objects that they own are also automatically dropped, and any privileges that they granted are automatically revoked.

i Note

Database users created by an LDAP provider must be dropped manually by a user administrator. If the user is dropped on the LDAP server, the corresponding database user in SAP HANA is **not** automatically dropped.

Database users are created with either the `CREATE USER` or `CREATE RESTRICTED USER` statement, or using the SAP HANA cockpit.

Standard Users

Standard users correspond to users created with the `CREATE USER` statement. By default they can create objects in their own schema and read data in system views. Read access to system views is granted by the `PUBLIC` role, which is granted to every standard user.

Restricted Users

Restricted users, which are created with the `CREATE RESTRICTED USER` statement, initially have no privileges. Restricted users are intended for provisioning users who access SAP HANA through client applications and who are not intended to have full SQL access via an SQL console. If the privileges required to use the application are encapsulated within an application-specific role, then it is necessary to grant the user only this role. In this way, it can be ensured that users have only those privileges that are essential to their work.

Compared to standard database users, restricted users are initially limited in the following ways:

- They cannot create objects in the database as they are not authorized to create objects in their own database schema.
- They cannot view any data in the database as they are not granted the standard PUBLIC role.
- They are only able to connect to the database using HTTP/HTTPS.
For restricted users to connect via ODBC or JDBC, access for client connections must be enabled by executing the SQL statement `ALTER USER <user_name> ENABLE CLIENT CONNECT` or enabling the corresponding option for the user in the SAP HANA cockpit.
For full access to ODBC or JDBC functionality, users also require the predefined role `RESTRICTED_USER_ODBC_ACCESS` or `RESTRICTED_USER_JDBC_ACCESS`.

i Note

Disabling ODBC/JDBC access for a user, either a restricted user or a standard user, does not affect the user's authorizations or prevent the user from executing SQL commands via channels other than JDBC/ODBC. If the user has been granted SQL privileges (for example, system privileges and object privileges), he or she is still authorized to perform the corresponding database operations using, for example, a HTTP/HTTPS client.

A user administrator can convert a restricted user into a standard user (or vice versa) as follows:

- Granting (or revoking) the PUBLIC role
You can do this by editing the user in the SAP HANA cockpit or with the SQL statement `ALTER USER <username> GRANT | REVOKE ROLE PUBLIC`.
- Granting (or revoking) authorization to create objects in the user's own schema
You can do this by editing the user in the SAP HANA cockpit or with the SQL statement `ALTER USER <username> GRANT | REVOKE CREATE ANY ON OWN SCHEMA`.
- Enabling (or disabling) full SQL
You can do this by editing the user in the SAP HANA cockpit or with the SQL statement `ALTER USER <user_name> ENABLE CLIENT CONNECT`.

i Note

A user is only identified as a restricted user in system view `USERS` if he doesn't have the PUBLIC role or authorization for his own schema.

Users created by an LDAP provider can be either standard users or restricted users depending on the configuration.

Technical Database Users

Technical database users do not correspond to real people. They are therefore not dropped if a person leaves the organization. This means that they should be used for administrative tasks such as creating objects and granting privileges for a particular application.

Some technical users are available as standard, for example, the users `SYS` and `_SYS_REPO`.

Other technical database users are created for application-specific purposes. For example, an application server may log on to the SAP HANA database using a dedicated technical database user.

Technical users are standard users created with the CREATE USER statement.

Related Information

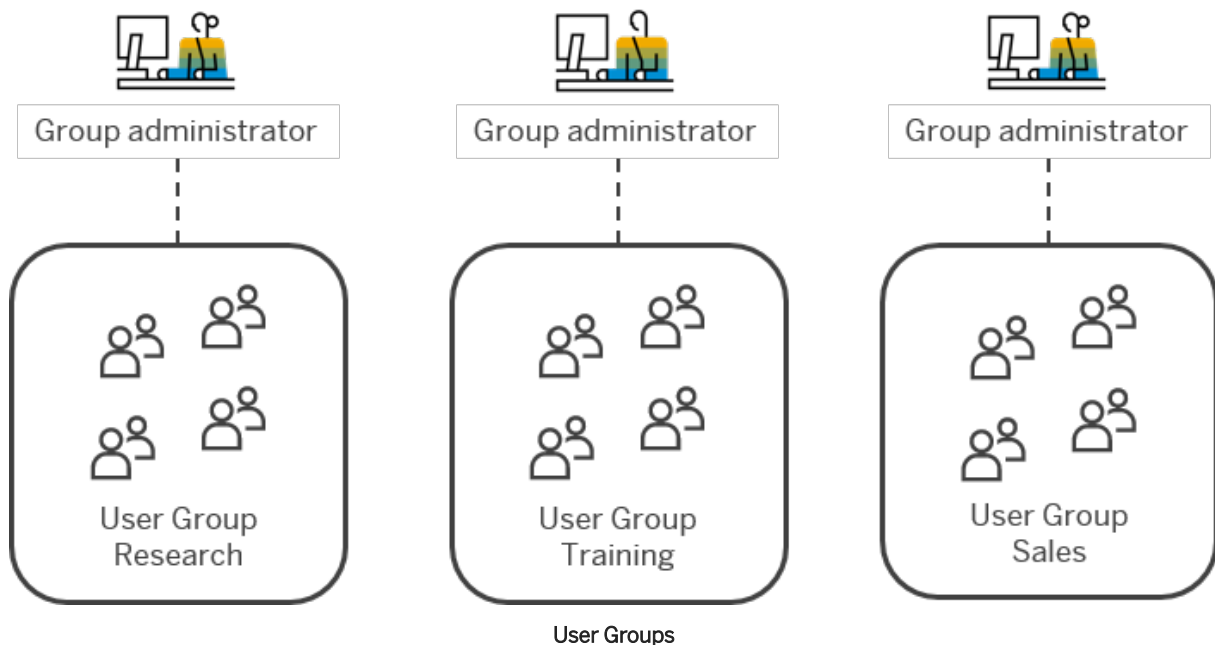
[Predefined Database \(Catalog\) Roles \[page 159\]](#)

6.2 User Groups

User groups allow you to manage related users together. Group administrators can be assigned to manage individual user groups exclusively and independently of each other.

- [Overview \[page 73\]](#)
- [Group Creation and Administration Mode \[page 74\]](#)
- [Group Membership \[page 75\]](#)
- [User Configuration with User Groups \[page 76\]](#)

Overview



User groups are an efficient way to manage users:

- Every user group can have its own **dedicated administrator(s)**. In this way, user management tasks can be delegated to several people independently of each other.

- A user group can be configured for **exclusive administration**, which means that only the designated group administrator(s) can manage the users in the group. This could be useful, for example, to protect highly-privileged users or technical users from accidental deletion or manipulation.
- **Group specific-user configuration** is possible. By setting user properties at the group level, you can configure related users quickly and differently to users in other groups. For example, you could put the technical users required by connecting applications into their own user group with a customized password policy so that the passwords of these users are extra complex.

i Note

User groups do not control data access. They are intended to support a separation of user management duties. A user's authorization (roles and privileges) control data access.

Group Creation and Administration Mode

A global user administrator (that is, a user with system privilege USER ADMIN) creates user groups using the `CREATE USERGROUP` statement. The user administrator can then designate one or more group administrators by granting the object privilege `USERGROUP OPERATOR` on the user group to the relevant user.

i Note

A user can be the group administrator of more than one group.

User administrators can also specify the administration "mode" of the user group. This determines who can administer the group:

- Group administrators **and** user administrators (shared administration)
Not only designated group administrators can modify the group but also any user administrator (that is, any user with system privilege USER ADMIN).
This is the default administration mode.
- Group administrators only (exclusive administration)
Only the designated group administrator(s) can modify the group, for example, adding new users to the group or removing users from the group. In this way, groups of users can be managed completely independently of each other.

i Note

To add an existing user from the global pool of users to the user group, as well as remove a user from the group (and return it to the global pool of users), the group administrator also needs to be a user administrator, that is have system privilege USER ADMIN.

The user administrator can configure the administration mode when creating the group, or later by editing the group. The authorization mode is configured with the option `ENABLE | DISABLE USER ADMIN` of the `CREATE USERGROUP` and `ALTER USERGROUP` statements as follows:

| Statement | Administration Mode |
|---|--|
| <code>CREATE USERGROUP <usergroupname></code> | Group administrators and user administrators (default if authorization mode is not explicitly specified) |

| Statement | Administration Mode |
|---|--|
| CREATE USERGROUP <usergroupname> DISABLE USER ADMIN | Group administrators only (exclusive administration) |
| ALTER USERGROUP <usergroupname> DISABLE USER ADMIN | Group administrators only (exclusive administration) |
| ALTER USERGROUP <usergroupname> ENABLE USER ADMIN | Group administrators and user administrators |

❖ Example

1. User administrator creates a new user group requiring its own exclusive administrator:

```
CREATE USERGROUP TechnicalUsers DISABLE USER ADMIN;
```
2. User administrator assigns the group administrator:

```
GRANT USERGROUP OPERATOR ON USERGROUP TechnicalUsers TO TechnicalUsersAdmin;
```
3. Group administrator adds new user to the group:

```
CREATE USER sapadm PASSWORD <password> SET USERGROUP TechnicalUsers;
```

For more information on who can perform exactly which operations, see the *Reference: Authorization for User Group Administration*.

Group Membership

User administrators and/or group administrators add new or existing users to a user group with the SET USERGROUP option of the CREATE | ALTER USER statements.

| Statement | Group Membership |
|--|---------------------------------------|
| CREATE USER <username> SET USERGROUP <usergroupname> | Creates new user in a user group |
| ALTER USER <username> SET USERGROUP <usergroupname> | Adds an existing user to a user group |
| ALTER USER <username> UNSET USERGROUP | Removes a user from a user group |

A user can belong to only one user group, but a user does not have to be a member of any group. Users who are not in any group are managed as normal by user administrators.

To move a user from one group to another, a user authorized for both user groups simply add the user to the new user group with the ALTER USER <username> SET USERGROUP <usergroupname>. This automatically removes the user from the original user group.

User Configuration with User Groups

In addition to grouping users into meaningful categories, user groups also allow you to mass manage certain user parameters. In this way, you can configure all users in a user group not only quickly but differently to users in other groups.

There are two steps to configuring users with user groups:

- Configuring the group-specific values of parameters
- Enabling the parameter set to which the configured parameters belong so that they take effect

Similarly to the configuration of authorization mode, the user parameters can be configured when the user group is created or by editing the group later using the `CREATE | ALTER USERGROUP` statement. To set parameter values use the `SET PARAMETER` option and to enable parameter sets, use the `ENABLE PARAMETER SET` option.

To see the group-specific values of parameters, query the view `USERGROUP_PARAMETERS`

Password policy is the only set of parameters that can be configured for a user group.

Parameter Set: Password Policy

The users of different user groups may have different requirements when it comes to passwords. For example, you may want the passwords of technical users to be very complex. A group administrator can configure group-specific values for the individual parameters of the password policy.

Example

```
CREATE USERGROUP Training SET PARAMETER 'password_layout' = 'A1a!',  
'minimal_password_length' = '16' ENABLE PARAMETER SET 'password policy';
```

Note

If a group-specific value is not explicitly set for a parameter, the value configured in the password policy of the database appears as the user group value in `USERGROUP_PARAMETERS`.

To see which values are currently in effect for a particular user, query the view `M_EFFECTIVE_PASSWORD_POLICY`.

Sample Code

```
SELECT * from "PUBLIC"."M_EFFECTIVE_PASSWORD_POLICY" where USER_NAME =  
'<user_name>';
```

Related Information

[SAP HANA Authorization \[page 119\]](#)

[Password Policy \[page 96\]](#)

[Password Policy Configuration Options \[page 98\]](#)

[SQL Statements and Authorization for User Group Administration \(Reference\) \[page 77\]](#)

6.2.1 SQL Statements and Authorization for User Group Administration (Reference)

Creating and configuring user groups, and subsequently managing the users in those groups requires different combinations of privileges.

Creating and Configuring User Groups

User administrators create and configure user groups. Group administrators can change the configuration.

| To... | You need... | For example... |
|---|---|---|
| Create a user group | System privilege USER ADMIN | <ul style="list-style-type: none"> • CREATE USERGROUP TechnicalUsers; • CREATE USERGROUP Research DISABLE USER ADMIN; |
| Change the administration mode of a user group | System privilege USER ADMIN | <ul style="list-style-type: none"> • ALTER USERGROUP TechnicalUsers DISABLE USER ADMIN; • ALTER USERGROUP Research ENABLE USER ADMIN; |
| Make another user the group administrator of a user group | <ul style="list-style-type: none"> • System privilege USER ADMIN, or • Object privilege USERGROUP OPERATOR on the group with the option to grant it to others | GRANT USERGROUP OPERATOR ON USERGROUP TechnicalUsers TO TechnicalUsersAdmin; |

| To... | You need... | For example... |
|---|---|---|
| Configure user parameters and enable/disable usergroup parameter sets | <ul style="list-style-type: none"> System privilege USER ADMIN, or Object privilege USERGROUP OPERATOR on the group <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If the user group has been configured for exclusive administration, USERGROUP OPERATOR on the group is required.</p> </div> | <ul style="list-style-type: none"> CREATE USERGROUP Training SET PARAMETER 'password_layout' = 'A1a!', 'minimal_password_length' = '7' ENABLE PARAMETER SET 'password policy'; ALTER USERGROUP TechnicalUsers SET PARAMETER 'force_first_password_change' = 'false'; ALTER USERGROUP TechnicalUsers ENABLE PARAMETER SET 'password policy'; ALTER USERGROUP TechnicalUsers DISABLE PARAMETER SET 'password policy'; |
| Delete a user group | System privilege USER ADMIN <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>You cannot delete a user group if there are still users in the group.</p> </div> | DROP USERGROUP Training; |

Managing Users

Managing users who are not in any user group

User administrators manage users who do not belong to a user group.

| To... | You need... | For example... |
|---|-----------------------------|---|
| Create, change, delete a user not in any user group | System privilege USER ADMIN | CREATE USER Michael PASSWORD <password>; |

Managing users in a user group configured for shared administration (default)

Group administrators and user administrators can manage users in user groups configured with the option ENABLE USER ADMIN.

| To.. | You need... | For example... |
|--|---|---|
| Create a user in the user group | <ul style="list-style-type: none"> System privilege USER ADMIN, or Object privilege USERGROUP OPERATOR on the group | <code>CREATE USER John PASSWORD <password> SET USERGROUP Research;</code> |
| Delete a user in the user group | <ul style="list-style-type: none"> System privilege USER ADMIN, or Object privilege USERGROUP OPERATOR on the group | <code>DROP USER John CASCADE;</code> |
| Add an existing user from the global pool of users to the user group | System privilege USER ADMIN | <code>ALTER USER Julie SET USERGROUP Research;</code> |
| Move a user from another user group to the user group | <ul style="list-style-type: none"> System privilege USER ADMIN, or Object privilege USERGROUP OPERATOR on both groups | <code>ALTER USER Julie SET USERGROUP Training;</code> |
| Remove a user from the user group (and return to the global pool of users) | System privilege USER ADMIN | <code>ALTER USER Julie UNSET USERGROUP;</code> |

Users in a user group configured for exclusive administration

Only group administrators can manage users in user groups configured with the option `DISABLE USER ADMIN`.

| To.. | You need... | For example... |
|--|--|---|
| Create a user in the user group | Object privilege USERGROUP OPERATOR on the user group | <code>CREATE USER sapsid PASSWORD <password> SET USERGROUP TechnicalUsers;</code> |
| Delete a user in the user group | Object privilege USERGROUP OPERATOR on the user group | <code>DROP USER sapsid CASCADE;</code> |
| Add an existing user from the global pool of users to the user group | <ul style="list-style-type: none"> System privilege USER ADMIN and Object privilege USERGROUP OPERATOR on the group | <code>ALTER USER Thomas SET USERGROUP TechnicalUsers;</code> |
| Move a user from another user group to the user group | <ul style="list-style-type: none"> Object privilege USERGROUP OPERATOR on both groups | <code>ALTER USER Julie SET USERGROUP TechnicalUsers;</code> |
| <div style="border: 1px solid #0070C0; padding: 5px; background-color: #E6F2FF;"> <p>i Note</p> <p>If the user's current group is not configured for exclusive administration, object privilege USERGROUP OPERATOR on this group is not required; system privilege USER ADMIN is sufficient.</p> </div> | | |
| Remove a user from the user group | <ul style="list-style-type: none"> System privilege USER ADMIN and Object privilege USERGROUP OPERATOR on the group | <code>ALTER USER Thomas UNSET USERGROUP;</code> |

Related Information

[Password Policy Configuration Options \[page 98\]](#)

6.3 User Administration Tools

Depending on your organization and its user provisioning strategy, people with different job functions may be involved in the process of user administration. Different tools are used for different tasks.

Native SAP HANA User Administration

The recommended process for provisioning users in SAP HANA is as follows:

1. Define and create roles.
2. Define and create user groups.
3. Create users in user groups.
4. Grant roles to users.

iNote

Creating user groups and assigning users to user groups is an optional step and depends on the requirements in your setup.

Further administration tasks include:

- Deleting users when they leave the organization
- Reactivating users after too many failed logon attempts
- Deactivating users if a security violation has been detected
- Resetting user passwords

The following table provides an overview of who does which of these tasks and the SAP HANA tools available:

| Job Function | Task | Environment | Tool |
|---|--|-------------|--|
| Role designer or creator | <p>Create roles and role hierarchies that reflect the access requirements, job function, and responsibilities of system users</p> <div data-bbox="805 443 1093 913" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p>i Note</p> <p>It is also possible to create roles in runtime on the basis of SQL statements. However, it is recommended that you create roles in the repository as they offer more flexibility (for example, they can be transported between systems). For more information, see <i>Catalog Roles and Repository Roles Compared</i>.</p> </div> | Design time | <ul style="list-style-type: none"> SAP HANA XS advanced: SAP Web IDE for SAP HANA SAP HANA XS classic: <i>Developer Workbench</i> of the SAP HANA studio or <i>Editor</i> tool of the SAP HANA Web-based Development Workbench |
| Application developer | <p>Create roles for new applications developed on SAP HANA XS classic</p> <div data-bbox="502 1048 790 1668" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p>i Note</p> <p>SAP HANA XS advanced has the additional concept of application roles and role collections. These are independent of database roles in SAP HANA itself. In the XS advanced context, SAP HANA database roles are used only to control access to database objects (for example, tables, views, and procedures) for XS advanced applications. For more information about the authorization concept of XS advanced.</p> </div> | Design time | <ul style="list-style-type: none"> SAP HANA XS classic: <i>Developer Workbench</i> of the SAP HANA studio or <i>Editor</i> tool of the SAP HANA Web-based Development Workbench |
| User, user group, or system administrator | <p>Create user groups</p> <hr/> <p>Create SAP HANA database users in user groups</p> <hr/> <p>Grant roles to database users</p> <hr/> <p>Delete, deactivate, and reactivate database users</p> | Runtime | <ul style="list-style-type: none"> <i>User</i> page of the SAP HANA cockpit <i>User</i> editor of the SAP HANA studio <i>Security</i> tool of the SAP HANA Web-based Development Workbench SAP HANA hdbsql |

| Job Function | Task | Environment | Tool |
|------------------------------|-------------------------------|-------------|--|
| | Reset user passwords | | hdsq1 is useful when using scripts for automated processing. |
| User or system administrator | Grant roles to database users | Runtime | <ul style="list-style-type: none"> • Assign Roles page of the SAP HANA cockpit • User editor of the SAP HANA studio • Security tool of the SAP HANA Web-based Development Workbench |

SAP HANA Lifecycle Management Tool hdblcm(gui)

You can use the SAP HANA lifecycle management tools to perform post-installation steps including changing the passwords of the system database user SYSTEM and operating system administrator <sid>adm as part of system rename. For more information, see the *SAP HANA Administration Guide*.

Integration into Other Identity Management Tools

LDAP-Compliant Identity Management Server

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing directory services. If you use an LDAP-compliant directory server to manage users and their access to resources, you can leverage LDAP-based authentication to access SAP HANA and LDAP group membership to authorize users. With LDAP-based authentication, users can be automatically provisioned in SAP HANA.

SAP NetWeaver Identity Management

SAP NetWeaver Identity Management 7.2 Support Package Stack (SPS) 3 and higher contains a connector to the SAP HANA database. With SAP NetWeaver ID Management you can perform several user administration tasks in the SAP HANA database, including:

- Creating and deleting user accounts
- Granting roles

i Note

Roles created in runtime are supported as of SAP NetWeaver ID Management SPS 8. Roles created in design time are supported as of SPS 9.

- Setting passwords for users

To use the SAP HANA connector for SAP NetWeaver ID Management, a dedicated SAP HANA database user must be created with the following roles and privileges:

- Standard role MONITORING
- System privilege ROLE ADMIN and USER ADMIN
- Object privilege EXECUTE on the procedure GRANT_ACTIVATED_ROLE

SAP Access Control

SAP Access Control 10.1 can be used to manage access and risk analysis for SAP HANA-based authorizations.

Related Information

User Administration in SAP HANA

[Database Roles \[page 158\]](#)

[Catalog Roles and Design-Time Roles Compared \[page 163\]](#)

[SAP HANA DI Roles \[page 166\]](#)

[Repository Roles \[page 169\]](#)

[Authorization in SAP HANA XS Advanced \[page 313\]](#)

[SAP Note 1986645](#)

Integration into Other Identity Management Tools

[LDAP User Authentication \[page 114\]](#)

[LDAP Group Authorization \[page 180\]](#)

[SAP NetWeaver Identity Management](#)

[SAP Access Control](#)

6.4 Predefined Users

A number of predefined users are required for installing, upgrading, and operating the SAP HANA database.

The following tables list the users that are available by default in SAP HANA.

Note

If you have installed the runtime environment for application development, SAP HANA Extended Application Services (XS) Advanced Model, several additional predefined users are available. For more information, see the section *Predefined XS Advanced Users*.

Operating System Users

`<sid>adm`

User

`<sid>adm`, where `<sid>` is the ID of the SAP HANA system

Description The <sid>adm user is an operating system user and is also referred to as the operating system administrator.

This operating system user has unlimited access to all local resources related to SAP systems.

This user is not a database user but a user at the operating system level.

i Note

In a system configured for high isolation, additional OS users will exist for every tenant database. Access to database-specific data is limited accordingly.

Password Specification The initial password is specified during installation by your hardware partner or certified administrator. After handover, it is important that you change this password. A system administrator can do this at the operating system level. It is also possible as part of a system rename with SAP HANA lifecycle manager.

More Information For more information about file and directory permissions in systems configured for high isolation, see the *SAP HANA Administration Guide*.

Database Users

SYSTEM

| | |
|------|--------|
| User | SYSTEM |
|------|--------|

Description

The `SYSTEM` database user is created during the creation of the SAP HANA database. It is the most powerful database user with irrevocable system privileges, such as the ability to create other database users, access system tables, and so on.

The `SYSTEM` user of the **system database** has additional privileges, namely the privileges required for managing tenant databases, for example, creating and dropping databases, changing configuration (*.ini) files of databases, performing database-specific data backups, stopping and starting databases.

The `SYSTEM` user does **not** automatically have access to objects created in the SAP HANA repository.

⚠ Caution

Do not use the `SYSTEM` user for day-to-day activities. Instead, use this user to create dedicated database users for administrative tasks and to assign privileges to these users. It is recommended that you then deactivate the `SYSTEM` user. You may temporarily reactivate the `SYSTEM` user for emergency or bootstrapping tasks. See *Deactivate the SYSTEM User*.

i Note

The `SYSTEM` user is not required to update the SAP HANA database system; a lesser-privileged user can be created for this purpose. However, to upgrade SAP support package stacks, SAP enhancement packages and SAP systems using the Software Update Manager (SUM) and to install, migrate, and provision SAP systems using the Software Provisioning Manager (SWPM), the `SYSTEM` user **is required** and needs to be temporarily reactivated for the duration of the upgrade, installation, migration or provisioning.

Password Specification

The initial password of the `SYSTEM` user of the system database and the first tenant database (if applicable) is specified by your hardware partner or certified administrator during installation. After handover, it is important that you change these passwords.

The initial password of the `SYSTEM` user of subsequently created tenant databases is specified at the time of database creation.

i Note

If you updated a single-container system to SAP HANA 2.0 SPS 02, the `SYSTEM` user of the tenant database created during the update process has the password of the original `SYSTEM` user. You are required to specify the password of the `SYSTEM` user of the new system database during the update process.

A user administrator (that is, a user with the system privilege `USER ADMIN`) can change the `SYSTEM` user password of a database in the SAP HANA cockpit. An administrator in the system database with system privilege `DATABASE ADMIN` can reset the password of the `SYSTEM` user in a tenant database.

i Note

It is also possible to change the `SYSTEM` user password of the system database as part of a system rename with SAP HANA lifecycle manager.

More Information

For more information about how to change the `SYSTEM` user password, see the *SAP HANA Administration Guide*.

SYS

| | |
|------------------------|---|
| User | SYS |
| Description | SYS is a technical database user. It is the owner of database objects such as system tables and monitoring views. |
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |

XSSQLCC_AUTO_USER_<generated_ID>

| | |
|-------------|---|
| User | XSSQLCC_AUTO_USER_<generated_ID> |
| Description | <p>In the runtime configuration of an SAP HANA XS application (SAP HANA XS, classic model), a technical user is automatically generated for an SQL connection configuration (SQLCC) if no user is specified.</p> <p>The user is created on activation of the SQLCC and is automatically granted the role specified in the configuration. If the SQLCC is deactivated, the user cannot be used in runtime.</p> <p>With the standard SAP HANA XS application <i>SAP HANA XS Admin Tools</i>, available with the deployment of delivery unit HANA_XS_BASE, two such users are created:</p> <ul style="list-style-type: none">• The technical user used by <i>User Self-Service Administration</i> tool to execute tasks associated with user self-service requests, for example, sending e-mails in response to user requests. This user is associated with the SQLCC artifact <code>sap.hana.xs.selfService.userSelfService.xssqlcc</code> and is assigned the role <code>sap.hana.xs.selfService.user.roles::USSExecutor</code>. This user cannot be used to log on to SAP HANA.• The technical user used by the <i>SAP Web Dispatcher HTTP Tracing</i> tool to connect to the database for the purpose of executing HTTP tracing of SAP HANA XS applications. This user is associated with the SQLCC artifact <code>sap.hana.xs.admin.webdispatcher.server.common.httpTracing.xssqlcc</code> and is assigned the role <code>sap.hana.xs.admin.roles::WebDispatcherHTTPTracingAdministrator</code>. This user cannot be used to log on to SAP HANA. |

Note

Since the above users don't have human readable names, check the assigned roles to see which user is which.

| | |
|------------------------|--|
| Password Specification | Password-based logon is disabled by default for an automatically generated SQLCC user. Therefore, a password is not required. |
| More Information | <p>For more information about the roles mentioned above, see <i>HANA_XS_BASE</i> in the reference section of the <i>SAP HANA Security Guide</i>.</p> <p>For more information about SQLCCs and the above applications, see the section about maintaining the SAP HANA XS classic model runtime in the <i>SAP HANA Administration Guide</i>.</p> |

_SYS_AFL

| | |
|-------------------------------|---|
| User | _SYS_AFL |
| Description | _SYS_AFL is a technical user that owns all objects for Application Function Libraries. |
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |
| More Information | For more information, see the <i>SAP HANA Business Function Library (BFL)</i> reference. |

_SYS_EPM

| | |
|-------------------------------|---|
| User | _SYS_EPM |
| Description | _SYS_EPM is a technical database used by the SAP Performance Management (SAP EPM) application |
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |

_SYS_REPO

| | |
|-------------------------------|---|
| User | _SYS_REPO |
| Description | <p>_SYS_REPO is a technical database user used by the SAP HANA repository (SAP HANA XS, classic model). The repository consists of packages that contain design time versions of various objects, such as attribute views, analytic views, calculation views, procedures, analytic privileges, and roles. _SYS_REPO is the owner of all objects in the repository, as well as their activated runtime versions.</p> <div data-bbox="517 1254 1394 1512" style="background-color: #f0f0f0; padding: 10px;"><p>i Note</p><p>Objects in the repository can be modeled on data objects that are not part of design time, such as tables that are used in replication scenarios. _SYS_REPO does not automatically have authorization to access these objects. _SYS_REPO must therefore be granted the SELECT privilege (with grant option) on all data objects underneath all objects modeled in the repository. If this privilege is missing, the activated objects will be invalidated. This is true even for objects belonging to schema SYS.</p></div> |
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |

_SYS_SQL_ANALYZER

| | |
|--------------------|---|
| User | _SYS_SQL_ANALYZER |
| Description | <p>_SYS_SQL_ANALYZER is a technical user used by the query performance analysis tool of SAP HANA, the SQL analyzer.</p> <p>This tool can be used to view detailed information on each query and can help you evaluate potential bottlenecks and optimizations for these queries. The SQL analyzer is accessible from the SAP HANA cockpit and SAP Web IDE for SAP HANA.</p> |

| | |
|-------------------------------|---|
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |
| More Information | For more information about the SQL analyzer, see the SAP HANA Administration Guide. |

_SYS_STATISTICS

| | |
|-------------------------------|---|
| User | <code>_SYS_STATISTICS</code> |
| Description | <code>_SYS_STATISTICS</code> is a technical database user used by the internal monitoring mechanism of the SAP HANA database. It collects information about status, performance, and resource usage from all components of the database and issues alerts if necessary. |
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |

_SYS_TASK

| | |
|-------------------------------|--|
| User | <code>_SYS_TASK</code> |
| Description | <code>_SYS_TASK</code> is a technical database user in SAP HANA smart data integration. This user owns all task framework objects. |
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |
| More Information | For more information, see SAP HANA Smart Data Integration and SAP HANA Smart Data Quality on SAP Help Portal. |

_SYS_WORKLOAD_REPLAY

| | |
|-------------------------------|--|
| User | <code>_SYS_WORKLOAD_REPLAY</code> |
| Description | <code>_SYS_WORKLOAD_REPLAY</code> is a technical database user used by capture and replay capability of the SAP HANA Performance Management tool. This tool allows administrators to capture and replay workloads from an SAP HANA system in order to check the impact of a system change (for example, hardware change). The user <code>_SYS_WORKLOAD_REPLAY</code> manages control and preprocessing data. Performance results are also stored in this user's schema (<code>_SYS_WORKLOAD_REPLAY</code>), but are only accessible by internal procedures. |
| Password Specification | Not applicable This is a technical database user. It is not possible to log on with this user. |
| More Information | For more information about SAP HANA Workload Capture and Replay, see the <i>SAP HANA Administration Guide</i> . |

_SYS_XB

| | |
|--------------------|---|
| User | <code>_SYS_XB</code> |
| Description | <code>_SYS_XB</code> is a technical user for internal use only. |

| | |
|-------------------------------|---|
| Password Specification | Not applicable |
| | This is a technical database user. It is not possible to log on with this user. |

Database Users Related to the SAP HANA Deployment Infrastructure (HDI)

i Note

The following users exist only if HDI is enabled in the database.

_SYS_DI*

| | |
|-------------------------------|---|
| User | <code>_SYS_DI, _SYS_DI_*_CATALOG, _SYS_DI_SU, _SYS_DI_TO</code> |
| Description | <p>Technical users of the SAP HANA Deployment Infrastructure (HDI):</p> <ul style="list-style-type: none"> • <code>_SYS_DI</code> Owns all HDI SQL-based APIs, for example all API procedures in the <code>_SYS_DI</code> schema and API procedures in containers • <code>_SYS_DI_*_CATALOG</code> Technical users used by the HDI to access database system catalog tables and views • <code>_SYS_DI_SU</code> Technical superuser of the HDI created at installation time • <code>_SYS_DI_TO</code> Owns transaction and connections of all internal HDI transactions |
| Password Specification | <p>Not applicable</p> <p>These are technical database users. It is not possible to log on with these users.</p> |

Technical Users for HDI Schema-Based Containers

The deployment of database objects with SAP HANA Deployment Infrastructure (HDI) is based on a container model where each container corresponds roughly to a database schema. Each schema, and the database objects deployed into the schema, are owned by a dedicated technical database user.

For every container deployed, a new technical database user and schema with the same name as the container are created. Additional schemas and technical users required for metadata and deployment APIs are also created.

For example, for a container named `s`, HDI creates the following users:

- `s`:
The user who is the owner of the container schema `s`
- User `s#DI`:
The user who is the owner of the schema containing metadata and deployment APIs
- User `s#OO`:
The user who is the owner of database objects in schema `s`
- Users `_DI#s#METADATA_COM_SAP_HANA_DI_<metadata>`:
The users who are the owners of schemas containing build plug-in metadata

These technical users are used internally by HDI only. They are created as restricted database users who do not have any privileges by default (not even the role `PUBLIC`). They cannot be used to log on to the database.

For more information, see the section on maintaining HDI containers in the *SAP HANA Developer Guide (For SAP HANA XS Advanced Model)*.

Related Information

[Predefined XS Advanced Users \[page 305\]](#)

[Deactivate the SYSTEM User \[page 90\]](#)

[HANA_XS_BASE \[page 365\]](#)

[SAP HANA Smart Data Integration and SAP HANA Smart Data Quality](#)

6.5 Deactivate the SYSTEM User

As the most powerful database user, `SYSTEM` is not intended for use in production systems. Use it to create lesser privileged users for particular purposes and then deactivate it.

Prerequisites

You have the system privilege `USER ADMIN`.

Context

It is highly recommended that you do not use `SYSTEM` for day-to-day activities in production environments. Instead, use it to create database users with the minimum privilege set required for their duties (for example, user administration, system administration). Then deactivate `SYSTEM`. You may temporarily reactivate the `SYSTEM` user for emergency or bootstrapping tasks.

i Note

The `SYSTEM` user is not required to update the SAP HANA database system; a lesser-privileged user can be created for this purpose. However, to upgrade SAP support package stacks, SAP enhancement packages and SAP systems using the Software Update Manager (SUM) and to install, migrate, and provision SAP systems using the Software Provisioning Manager (SWPM), the `SYSTEM` user **is required** and needs to be temporarily reactivated for the duration of the upgrade, installation, migration or provisioning.

Procedure

Execute the following statement:

```
ALTER USER SYSTEM DEACTIVATE USER NOW
```

Results

The SYSTEM user is deactivated and can no longer **connect** to the SAP HANA database. However, it may appear as though the SYSTEM user is still active in the system (for example when a procedure that was created by SYSTEM with DEFINER MODE is called).

You can verify that this is the case in the USERS system view. For user SYSTEM, check the values in the columns USER_DEACTIVATED, DEACTIVATION_TIME, and LAST_SUCCESSFUL_CONNECT.

i Note

You can still use the SYSTEM user as an emergency user even if it has been deactivated. Any user with the system privilege USER ADMIN can reactivate SYSTEM with the statement `ALTER USER SYSTEM ACTIVATE USER NOW`. To ensure that an administrator does not do this surreptitiously, it is recommended that you create an audit policy monitoring ALTER USER statements. Also change the password of the SYSTEM user after reactivating it.

7 SAP HANA Authentication and Single Sign-On

The identity of database users accessing SAP HANA is verified through a process called authentication. SAP HANA supports several authentication mechanisms, several of which can be used for the integration of SAP HANA into single sign-on environments (SSO). The mechanisms used to authenticate individual users is specified as part of the user definition.

i Note

For JDBC and ODBC client connections, user passwords are always transmitted in encrypted hashed form during the user authentication process, never in plain text. For HTTP connections via SAP HANA XS classic, HTTPS must be configured. In SSO environments, we recommend using encrypted communication channels for **all** client connections.

[User Authentication Mechanisms \[page 93\]](#)

Authentication mechanisms supported in SAP HANA. Mechanisms that are not required can be disabled.

[SAP HANA Logon Checks \[page 96\]](#)

Before a user can connect to the SAP HANA database, the system performs several checks as part of the logon process.

[Password Policy \[page 96\]](#)

The passwords of database users are subject to certain rules. These are defined in the password policy.

[Single Sign-On Integration \[page 106\]](#)

Integrate SAP HANA into single sign-on environments using Kerberos, SAML 2.0, JSON web tokens, and logon and assertion tickets.

[LDAP User Authentication \[page 114\]](#)

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing directory services. If you use an LDAP-compliant directory server to manage users and their passwords, you can leverage LDAP-based authentication to access SAP HANA.

7.1 User Authentication Mechanisms

Authentication mechanisms supported in SAP HANA. Mechanisms that are not required can be disabled.

Supported Authentication Mechanisms

| Mechanism | Description | Can Be Used for SSO |
|--|---|---------------------|
| SAP HANA user name and password | Users accessing the SAP HANA database authenticate themselves by entering their database user name and their local SAP HANA password. | No |
| Kerberos, SPNEGO | <p>A Kerberos authentication provider can be used to authenticate users accessing SAP HANA in the following ways:</p> <ul style="list-style-type: none">• Directly from ODBC and JDBC database clients within a network (for example, the SAP HANA studio)• Indirectly from front-end applications such as SAP BusinessObjects applications and other SAP HANA databases using Kerberos delegation• Via HTTP/HTTPS access by means of SAP HANA Extended Services (SAP HANA XS), advanced model and classic model <p>In this case, Kerberos authentication is enabled with Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO).</p> | Yes |
| i Note A user who connects to the database using an external authentication provider must also have a database user known to the database. The external identity is mapped to the identity of an internal database user. | | |
| Security assertion markup language (SAML) | A SAML bearer assertion can be used to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients. SAP HANA can act as a service provider to authenticate users accessing via HTTP/HTTPS by means of SAP HANA XS classic and advanced. | Yes |
| i Note A user who connects to the database using an external authentication provider must also have a database user known to the database. The external identity is mapped to the identity of an internal database user. | | |

| Mechanism | Description | Can Be Used for SSO |
|-----------------------------|---|--|
| Logon and assertion tickets | <p>Users can be authenticated by SAP logon or assertion tickets issued to them when they log on to an SAP system that is configured to create tickets (for example, the SAP Web Application Server or Portal).</p> <p>i Note To implement logon/assertion tickets, the user specified in the logon/assertion ticket must already exist in SAP HANA; there is no support for user mapping.</p> | Yes |
| X.509 client certificates | <p>For HTTP/HTTPS access to SAP HANA by means of SAP HANA XS advanced model and classic model, users can be authenticated by client certificates signed by a trusted Certification Authority (CA), which can be stored in the SAP HANA XS trust store.</p> <p>i Note To implement X.509 client certificates, the user specified in the certificate must already exist in SAP HANA; there is no support for user mapping.</p> | Yes for HTTP/HTTPS access to SAP HANA by means of SAP HANA XS (advanced and classic) |
| JSON Web Token (JWT) | <p>A JSON Web Token can be used to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients or indirectly through SAP HANA extended application services, advanced model (SAP HANA XS, advanced).</p> <p>i Note A user who connects to the database using an external authentication provider must also have a database user known to the database. The external identity is mapped to the identity of an internal database user.</p> | Yes |
| LDAP | <p>A password stored in an LDAP directory server can be used to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients, if authentication using users' local SAP HANA authentication has been disabled.</p> <p>i Note A user who connects to the database using an external authentication provider must also have a database user known to the database. The external identity and the database user name are the same. If the LDAP provider is enabled to create database users in SAP HANA, the required user is created automatically if it doesn't exist.</p> | No |

| Mechanism | Description | Can Be Used for SSO |
|-----------------|---|---------------------|
| Session cookies | Session cookies are not technically an authentication mechanism. However, they reconnect users who have already been authenticated by Kerberos or SAML and extend the validity period of logon and assertion tickets. | Yes |

Isolated Single Sign-On for Tenant Databases

Separate, database-specific authentication is possible for every certificate-based authentication mechanism since it is possible to create different certificate collections for individual purposes directly in every database, and every database can have its own key pair and public key certificate.

For SAML assertions, X.509 certificates, JSON web tokens, and logon tickets, it is also possible to use certificate collections (or PSEs) located on the file system. It is still possible to configure different trust and key stores for every database in the `global.ini` file. However, bear the following points in mind:

- If different trust and key stores are not explicitly configured for tenant databases, the same ones will be used for all external communication channels (including HTTP) for all databases.

⚠ Caution

If you have configured in tenant databases or the system database single sign-on mechanisms that rely on trust stores located in the file system (such as SAP logon and assertion tickets or SAML) and the trust stores are shared, users of one tenant database may be able to log on to other databases in the system.

- By default, only the system administrator can configure separate trust and key stores for tenant databases by changing the relevant properties in the `global.ini` file. This is because tenant database administrators are prevented from changing any communication properties. They are in the default configuration change blacklist (`multidb.ini`).

For more information about certificate collections in the database and PSEs in the file system, see the section on certificate management.

For Kerberos-based authentication, a per-database configuration is not possible – databases users in all databases must be mapped to users in the same Key Distribution Center.

Disabling Authentication Mechanisms

By default all authentication mechanisms are enabled, but it is possible and recommended to disable those that are not used in your environment. You do this by configuring the parameter `[authentication] authentication_methods` in the `global.ini` configuration file. The value of this parameter specifies all enabled methods as a comma-separated list.

The default value is `password,kerberos,spnego,saml,saplogon,x509xs,jwt,sessioncookie,ldap`.

i Note

If you are using SAP HANA dynamic tiering, it is not possible to disable logon and assertion tickets (`saplogon`) as an authentication mechanism.

Changes to this parameter are audited by default if auditing is enabled.

Related Information

[Password Policy \[page 96\]](#)

[Single Sign-On Integration \[page 106\]](#)

[LDAP User Authentication \[page 114\]](#)

[Certificate Management in SAP HANA \[page 261\]](#)

[Actions Audited by Default Audit Policy \[page 244\]](#)

7.2 SAP HANA Logon Checks

Before a user can connect to the SAP HANA database, the system performs several checks as part of the logon process.

1. The database authenticates the user using the configured mechanism.
For example, if user name/password authentication is being enforced, the provided user name and password are verified.
2. The database verifies that the user's account is within its validity period.
In the system view `USERS`, the columns `VALID_FROM` and `VALID_UNTIL` must contain effective values for the user in question.
The validity period is an optional parameter that a user administrator can set during user provisioning.
3. The database verifies that the user's account is active.
In the system view `USERS`, the column `IS_DEACTIVATED` must contain the value `FALSE` for the user in question.
User accounts may be deactivated explicitly by a user administrator or by the system, for example, due to too many invalid logon attempts.

If all of the above checks are successful, the user is logged on to SAP HANA.

7.3 Password Policy

The passwords of database users are subject to certain rules. These are defined in the password policy.

The SAP HANA database comes with a default password policy. You can change this default password policy in line with your organization's security requirements. If you manage users in user groups, you can also configure group-specific password policies.

Database Password Policy

The password policy of the database is defined by parameters in the `password_policy` section of the `indexserver.ini` configuration file for tenant databases and the `nameserver.ini` configuration file for the system database. The database password policy is valid for all database users unless the user is in a user group with its own dedicated password policy.

⚠ Caution

Do not change configuration files directly. Such changes cannot be audited.

You can view and change the database password policy on the [Password Policy and Blacklist](#) page of the SAP HANA cockpit or the [Security](#) editor of the SAP HANA studio.

You can also query the system view `M_PASSWORD_POLICY` to see the current database password policy.

User Group Password Policy

If the users of a user group have different password requirements, you can configure group-specific values for the individual parameters of the password policy in the definition of the user groups.

🔗 Example

```
CREATE USERGROUP Training SET PARAMETER 'password_layout' = 'Ala!',  
'minimal_password_length' = '16' ENABLE PARAMETER SET 'password policy';
```

i Note

If a group-specific value is not explicitly set for a parameter, the value configured in the password policy of the database appears as the user group value in `USERGROUP_PARAMETERS`.

→ Remember

The parameter set `password_policy` must be enabled for the user group in order for configured password policy parameters to take effect.

Effective Password Policy

To determine which password policy a user is currently subject to, query the system view `M_EFFECTIVE_PASSWORD_POLICY`.

📄 Sample Code

```
SELECT * from "PUBLIC"."M_EFFECTIVE_PASSWORD_POLICY" where USER_NAME =  
'<user_name>';
```

Related Information

[Auditing Activity in SAP HANA Systems \[page 240\]](#)

[User Groups \[page 73\]](#)

7.3.1 Password Policy Configuration Options

The password policy is defined by parameters in the `password_policy` section of the `indexserver.ini` configuration file (tenant databases) or `nameserver.ini` configuration file (system database). Password policy parameters may also be individually configured in the definition of a user group.

The following sections describe these parameters, which correspond to the configuration options available in the SAP HANA cockpit and SAP HANA studio.

- [Minimum Password Length \[page 98\]](#)
- [Lowercase Letters/Uppercase Letters/Numerical Digits/Special Characters Required \[page 99\]](#)
- [Password Change Required on First Logon \[page 100\]](#)
- [Number of Last Used Passwords That Cannot Be Reused \[page 100\]](#)
- [Number of Allowed Failed Logon Attempts \[page 101\]](#)
- [User Lock Time \[page 101\]](#)
- [Minimum Password Lifetime \[page 102\]](#)
- [Maximum Password Lifetime \[page 102\]](#)
- [Lifetime of Initial Password \[page 103\]](#)
- [Maximum Duration of User Inactivity \[page 103\]](#)
- [Notification of Password Expiration \[page 103\]](#)
- [Exempt SYSTEM User from Locking \[page 104\]](#)
- [Detailed Error Information on Failed Logon \[page 104\]](#)

Minimum Password Length

The minimum number of characters that the password must contain

| | |
|-------------------------------|--|
| Parameter | <code>minimal_password_length</code> |
| Default Value | 8 (characters) |
| Additional Information | You must enter a value between 6 and 64. |
| UI Label | <i>Minimum Password Length</i> |

Lowercase Letters/Uppercase Letters/Numerical Digits/ Special Characters Required

The character types that the password must contain and how many

| | |
|-------------------------------|---|
| Parameter | password_layout |
| Default Value | Aa1, that is, at least one uppercase letter, at least one number, and at least one lowercase letter |
| Additional Information | <p>The following character types are possible:</p> <ul style="list-style-type: none">• Lowercase letter (a-z)• Uppercase letter (A-Z)• Numerical digits (0-9)• Special characters (underscore (_), hyphen (-), and so on) Any character that is not an uppercase letter, a lowercase letter, or a numerical digit is considered a special character. <p>The following formats are supported for passwords:</p> <pre><password> ::= { <letter> [{ <letter_or_digit> # \$ } [...]] <digit> [<letter_or_digit> [...]] <any_quoted_string> }</pre> <p>If configuring this option in the <code>indexserver.ini</code> file using the <code>password_layout</code> parameter, you can use any specific letters, numbers and special characters, and the characters can be in any order. For example, the default value example could also be represented by a1A, hQ5, or 9fG. To enforce the use of at least one of each character type including special characters, you specify A1a_ or 2Bg?. To enforce the use of a specific number of a particular character type, specify the character type multiple times. For example, if passwords must contain at least 3 digits, you could specify the layout with a123A or 789fG.</p> <div data-bbox="603 1429 1391 1630"><p>Note</p><p>Passwords containing special characters other than underscore must be enclosed in double quotes ("). The SAP HANA studio does this automatically. When a password is enclosed in double quotes ("), any Unicode characters may be used.</p></div> <div data-bbox="603 1644 1391 1816"><p>Caution</p><p>The use of passwords enclosed in double quotes (") may cause logon issues depending on the client used. The SAP HANA studio and <code>hdsql</code> support passwords enclosed in double quotes (").</p></div> |
| UI Labels | <i>Lowercase Letters/Uppercase Letters/Numerical Digits/Special Characters Required</i> |

Password Change Required on First Logon

Defines whether users have to change their initial passwords immediately the first time they log on

| | |
|-------------------------------|--|
| Parameter | <code>force_first_password_change</code> |
| Default Value | True |
| Additional Information | <p>If this parameter is set to true, users can still log on with the initial password but every action they try to perform will return the error message that they must change their password.</p> <p>If this parameter is set to false, users are not forced to change their initial password immediately the first time they log on. However, if a user does not change the password before the number of days specified in the parameter <code>maximum_unused_initial_password_lifetime</code>, then the password still expires and must be reset by a user administrator.</p> <p>A user administrator (that is, a user with the system privilege USER ADMIN) can force a user to change his or her password at any time with the following SQL statement: <code>ALTER USER <user_name> FORCE PASSWORD CHANGE</code></p> <p>A user administrator can override this password policy setting for individual users (for example, technical users) with the following SQL statement:</p> <ul style="list-style-type: none">• <code>CREATE USER <user_name> PASSWORD <password> [NO FORCE_FIRST_PASSWORD_CHANGE]</code>• <code>ALTER USER <user_name> PASSWORD <password> [NO FORCE_FIRST_PASSWORD_CHANGE]</code> |
| UI Label | <i>Password Change Required on First Logon</i> |

Note

This parameter is only valid for users connecting with their SAP HANA database user name and password. It is not valid for connections established through other authentication mechanisms.

Number of Last Used Passwords That Cannot Be Reused

The number of last used passwords that the user is not allowed to reuse when changing his or her current password

| | |
|-------------------------------|---|
| Parameter | <code>last_used_passwords</code> |
| Default Value | 5 (previous passwords) |
| Additional Information | If you enter the value 0 , the user can reuse his or her old password. |
| UI Label | <i>Number of Last Used Passwords That Cannot Be Reused</i> |

Number of Allowed Failed Logon Attempts

The maximum number of failed logon attempts that are possible; the user is locked as soon as this number is reached

| | |
|-------------------------------|--|
| Parameter | <code>maximum_invalid_connect_attempts</code> |
| Default Value | 6 (failed logon attempts) |
| Additional Information | <p>You must enter a value of at least 1.</p> <p>A user administrator can reset the number of invalid logon attempts with the following SQL statement: <code>ALTER USER <user_name> RESET CONNECT ATTEMPTS</code></p> <p>The first time a user logs on successfully after an invalid logon attempt, an entry is made in the <code>INVALID_CONNECT_ATTEMPTS</code> system view containing the following information:</p> <ul style="list-style-type: none">• The number of invalid logon attempts since the last successful logon• The time of the last successful logon <p>A user administrator can delete information about invalid logon attempts with the following SQL statement: <code>ALTER USER <user_name> DROP CONNECT ATTEMPTS</code></p> <div data-bbox="603 1084 1394 1267"><p>→ Recommendation</p><p>Create an audit policy to log activity in the <code>INVALID_CONNECT_ATTEMPTS</code> system view. For example, create an audit policy that logs data query and manipulation statements executed on this view.</p></div> <div data-bbox="603 1285 1394 1518"><p>i Note</p><p>Although this parameter is not valid for the <code>SYSTEM</code> user, the <code>SYSTEM</code> user will still be locked if the parameter <code>password_lock_for_system_user</code> is set to true. If <code>password_lock_for_system_user</code> is set to false, the <code>SYSTEM</code> user will not be locked regardless of the number of failed logon attempts.</p></div> |
| UI Label | <i>Number of Allowed Failed Logon Attempts</i> |

User Lock Time

The number of minutes for which a user is locked after the maximum number of failed logon attempts

| | |
|----------------------|---------------------------------|
| Parameter | <code>password_lock_time</code> |
| Default Value | 1440 (minutes) |

Additional Information

If you enter the value **0**, the user is unlocked immediately. This disables the functionality of parameter `maximum_invalid_connect_attempts`.

A user administrator can reset the number of invalid logon attempts and reactivate the user account with the following SQL statement: `ALTER USER <user_name> RESET CONNECT ATTEMPTS`. It is also possible to reactivate the user in the user editor of the SAP HANA Studio.

To lock a user indefinitely, enter the value **-1**. On the [Password Policy and Blacklist](#) page of the SAP HANA cockpit or in the [Security](#) editor of the SAP HANA studio, this corresponds to selecting the [Lock User Indefinitely](#) checkbox. The user remains locked until reactivated by a user administrator as described above.

| | |
|----------|--------------------------------|
| UI Label | User Lock Time |
|----------|--------------------------------|

Minimum Password Lifetime

The minimum number of days that must elapse before a user can change his or her password

| | |
|------------------------|---|
| Parameter | <code>minimum_password_lifetime</code> |
| Default Value | 1 (day) |
| Additional Information | If you enter the value 0 , the password has no minimum lifetime. |
| UI Label | Minimum Password Lifetime |

Maximum Password Lifetime

The number of days after which a user's password expires

| | |
|------------------------|---|
| Parameter | <code>maximum_password_lifetime</code> |
| Default Value | 182 (days) |
| Additional Information | <p>You must enter a value of at least 1.</p> <p>A user administrator can exclude users from this password check with the following SQL statement: <code>ALTER USER <user_name> DISABLE PASSWORD LIFETIME</code>. However, this is recommended only for technical users only, not database users that correspond to real people.</p> <p>A user administrator can re-enable the password lifetime check for a user with the following SQL statement: <code>ALTER USER <user_name> ENABLE PASSWORD LIFETIME</code>.</p> |
| UI Label | Maximum Password Lifetime |

Lifetime of Initial Password

The number of days for which the initial password or any password set by a user administrator for a user is valid

| | |
|-------------------------------|---|
| Parameter | <code>maximum_unused_initial_password_lifetime</code> |
| Default Value | 7 (days) |
| Additional Information | You must enter a value of at least 1 . If a user has not logged on using the initial password within the given period of time, the user will be deactivated until their password is reset. |
| | i Note In SAP HANA 1.0 SPS 12 and earlier, this parameter was misspelled as <code>maximum_unused_inital_password_lifetime</code> . If this parameter had a user-specified value before upgrade, this value will be set as the value of the parameter <code>maximum_unused_initial_password_lifetime</code> . The misspelled parameter is unset and disappears from the custom configuration file. |
| UI Label | <i>Lifetime of Initial Password</i> |

Maximum Duration of User Inactivity

The number of days after which a password expires if the user has not logged on

| | |
|-------------------------------|---|
| Parameter | <code>maximum_unused_productive_password_lifetime</code> |
| Default Value | 365 (days) |
| Additional Information | You must enter a value of at least 1 . If a user has not logged on within the given period of time using any authentication method, the user will be deactivated until their password is reset. |
| UI Label | <i>Maximum Duration of User Inactivity</i> |

Notification of Password Expiration

The number of days before a password is due to expire that the user receives notification

| | |
|----------------------|---|
| Parameter | <code>password_expire_warning_time</code> |
| Default Value | 14 (days) |

Additional Information

Notification is transmitted via the database client (ODBC or JDBC) and it is up to the client application to provide this information to the user.

If you enter the value **0**, the user does not receive notification that his or her password is due to expire.

The system also monitors when user passwords are due to expire and issues a medium priority alert (check 62). This may be useful for technical database users since password expiration results in the user being locked, which may affect application availability. It is recommended that you disable the password lifetime check of technical users so that their password never expires (`ALTER USER <technical_username> DISABLE PASSWORD LIFETIME`).

| | |
|----------|---|
| UI Label | Notification of Password Expiration |
|----------|---|

Exempt SYSTEM User from Locking

Indicates whether or not the user SYSTEM is locked for the specified lock time (`password_lock_time`) after the maximum number of failed logon attempts (`maximum_invalid_connect_attempts`)

| | |
|------------------------|---|
| Parameter | <code>password_lock_for_system_user</code> |
| Default Value | true |
| Additional Information | This parameter cannot be configured for a user group. |
| UI Label | Exempt SYSTEM User from Locking |

Detailed Error Information on Failed Logon

Indicates the detail level of error information returned when a logon attempt fails

| | |
|------------------------|--|
| Parameter | <code>detailed_error_on_connect</code> |
| Default Value | false |
| Additional Information | <p>If set to false, only the information <code>authentication failed</code> is returned.</p> <p>If set to true, the specific reason for failed logon is returned:</p> <ul style="list-style-type: none">• Invalid user or password• User is locked• Connect try is outside validity period• User is deactivated |
| UI Label | Detailed Error Information on Failed Logon |

7.3.2 Password Blacklist

A password blacklist is a list of words that are not allowed as passwords or parts of passwords. A password blacklist can be managed for every database individually.

The password blacklist in SAP HANA is implemented with the table `_SYS_PASSWORD_BLACKLIST` in the schema `_SYS_SECURITY`. This table is empty when you create a new instance.

You can enter words in the password blacklist as part of password policy configuration on the [Password Policy and Blacklist](#) page of the SAP HANA cockpit. Alternatively, you can add words to table directly using the INSERT statement. For more information about the structure of this table, see `_SYS_PASSWORD_BLACKLIST`.

Sample Code

```
INSERT INTO _SYS_SECURITY._SYS_PASSWORD_BLACKLIST VALUES ('sap', 'TRUE', 'FALSE');
```

Note

Changes to the password blacklist do not affect the existing passwords of users. The modified blacklist applies the next time the user changes his or her password.

Note

Object privileges SELECT, INSERT, and DELETE on the `_SYS_PASSWORD_BLACKLIST` table are required to edit the password blacklist.

Related Information

[_SYS_PASSWORD_BLACKLIST \[page 105\]](#)

7.3.3 _SYS_PASSWORD_BLACKLIST

Shows the password blacklist

| Column Name | Data Type | Description |
|----------------|----------------|----------------------------------|
| BLACKLIST_TERM | NVARCHAR (256) | Blacklisted word or partial word |

| Column Name | Data Type | Description |
|------------------------|-------------|--|
| CHECK_PARTIAL_PASSWORD | VARCHAR (6) | Specifies whether passwords that contain the blacklisted word are excluded (TRUE) or only passwords that match the blacklisted word exactly are excluded (FALSE) |
| | | i Note Either TRUE or FALSE must be specified during an insert. |
| CHECK_CASE_SENSITIVE | VARCHAR (6) | Specifies whether the blacklisted word is case sensitive (TRUE) or case insensitive (FALSE) |
| | | i Note Either TRUE or FALSE must be specified during an insert. |

Sample Code

Add a new blacklisted word:

```
INSERT INTO _SYS_SECURITY._SYS_PASSWORD_BLACKLIST VALUES ('sap', 'TRUE', 'FALSE');
```

Sample Code

Remove a blacklisted word:

```
DELETE FROM _SYS_SECURITY._SYS_PASSWORD_BLACKLIST WHERE BLACKLIST_TERM = 'sap'
```

Related Information

[Password Blacklist \[page 105\]](#)

7.4 Single Sign-On Integration

Integrate SAP HANA into single sign-on environments using Kerberos, SAML 2.0, JSON web tokens, and logon and assertion tickets.

[Single Sign-On Using Kerberos \[page 107\]](#)

For integration into Kerberos-based SSO scenarios, SAP HANA supports Kerberos version 5 based on Active Directory (Microsoft Windows Server) or Kerberos authentication servers. For HTTP access

using SAP HANA Extended Services (SAP HANA XS) advanced and classic, Kerberos authentication is enabled with Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO).

[Single Sign-On Using SAML 2.0 \[page 108\]](#)

SAP HANA supports the Security Assertion Markup Language (SAML) for user authentication in single sign-on environments. SAML is used for authentication purposes only and not for authorization.

[Single Sign-On Using SAP Logon and Assertion Tickets \[page 111\]](#)

Users can be authenticated in SAP HANA by logon or assertion tickets issued to them when they log on to an SAP system configured to create tickets (for example, the SAP Web Application Server or Portal).

[Single Sign-On Using JSON Web Tokens \[page 112\]](#)

SAP HANA supports JSON Web Tokens (JWT) for user authentication in single sign-on environments.

7.4.1 Single Sign-On Using Kerberos

For integration into Kerberos-based SSO scenarios, SAP HANA supports Kerberos version 5 based on Active Directory (Microsoft Windows Server) or Kerberos authentication servers. For HTTP access using SAP HANA Extended Services (SAP HANA XS) advanced and classic, Kerberos authentication is enabled with Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO).

Kerberos is a network authentication protocol that provides authentication for client-server applications across an insecure network connection using secret-key cryptography.

ODBC and JDBC database clients support the Kerberos protocol, for example, the SAP HANA studio. Access from front-end applications (for example, SAP BusinessObjects XI applications) can also be implemented using Kerberos delegation. Support for constrained delegation and protocol transition is limited to scenarios in which the middle-tier application connects to SAP HANA as the database layer via JDBC.

Kerberos is supported for HTTP access using SAP HANA XS advanced and classic with Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO). It is up to the HTTP client whether it uses Kerberos directly or SPNEGO.

→ Recommendation

To avoid replay attacks, we recommend that you set up secure communication between the individual components of the SAP HANA database and client connections using the secure sockets layer (SSL) protocol when implementing Kerberos authentication, in particular when using Kerberos with insecure encryption algorithms such as RC4.

Configuration

To allow users to log on to the SAP HANA database from a client using Kerberos authentication, the following configuration steps are necessary:

1. Install MIT Kerberos client libraries on the host(s) of the SAP HANA system.
2. Configure the SAP HANA system for Kerberos and/or SPNEGO authentication.
3. Map SAP HANA database users to their external identities stored in the Kerberos key distribution center (KDC).

→ Remember

A per-database configuration is not possible – databases users in all databases must be mapped to users in the same KDC.

For more information about how to set up SSO with SAP HANA using Kerberos and Microsoft Active Directory, see SAP Note 1837331.

In distributed SAP HANA systems that use Kerberos delegation (SSO2DB), application disruptions resulting from expired authentication are avoided through the use of session cookies. This mechanism is active by default but can be disabled in the `indexserver.ini` configuration file with the `[authentication]` `session_cookie_for_kerberos` property.

Related Information

[SAP Note 1837331](#)

[SAP Note 2354473](#)

[SAP Note 1813724](#)

[SAP Note 2354556](#)

7.4.2 Single Sign-On Using SAML 2.0

SAP HANA supports the Security Assertion Markup Language (SAML) for user authentication in single sign-on environments. SAML is used for authentication purposes only and not for authorization.

SAML provides the mechanism by which the identity of users accessing the SAP HANA database from client applications is authenticated by XML-based assertions issued by a trusted identity provider. The internal database user to which the external identity is mapped is used for authorization checks during the database session.

SAML can be implemented to authenticate users accessing the SAP HANA database from the following client applications:

- Database clients that access the SQL interface of the SAP HANA database directly
This covers standard ODBC and JDBC database clients.
In this scenario, a SAML bearer assertion is used to authenticate the user directly. It is the client application's responsibility to retrieve the SAML bearer assertion used for logon. To log on using a SAML bearer assertion, you must set the user name to an empty string and the SAML bearer assertion as the password in your ODBC/JDBC connection properties.

i Note

The SAP HANA studio does not support SAML.

- Clients that connect to SAP HANA through the SAP HANA XS classic server via HTTP
In this scenario, SAP HANA acts as the service provider that authenticates users on the basis of their SAML bearer assertion.

→ Recommendation

To avoid replay attacks, we recommend that you set up secure communication between the individual components of the SAP HANA database and client connections using the Transport Layer Security (TLS) protocol when implementing SAML authentication.

SAML Assertion Specification

SAP HANA supports plain SAML 2.0 assertions as well as unsolicited SAML responses that include an unencrypted SAML assertion. SAML assertions and responses must be signed using XML signatures.

The following features of XML signatures are supported:

- SHA1, SHA256, and MD5 for hash algorithms
- RSA-SHA1 and RSA-SHA256 as signature algorithms

The following SAML assertion features are supported:

- Assertion Subject with `NameID`
- Qualified `NameID` with `SPProvidedID` and `SPNameQualifier`
- Validity conditions (`NotBefore`, `NotOnOrAfter`)
- Audience restrictions

The following properties of a SAML assertion are evaluated to log on the requesting user to SAP HANA:

| Property | Note |
|---|---|
| <code>saml:Assertion/@Version</code> | Required entry: 2.0 |
| <code>saml:Subject/saml:NameID</code> | Must be specified |
| <code>saml:Subject/saml:NameID/@Format</code> | Optional entry If present, entry can be <code>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</code> or <code>"urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"</code> |
| <code>saml:Subject/saml:NameID/@SPProvidedID</code> | Must either match an explicit user mapping in the SAP HANA database, or a wildcard mapping must be configured for the user |
| <code>saml:Subject/saml:SubjectConfirmation</code> | Optional If present, entry must be <code>{ "urn:oasis:names:tc:SAML:2.0:cm:bearer" }</code> |
| <code>saml:Conditions</code> | Condition <code>@NotOnOrAfter</code> must be set. <ul style="list-style-type: none">• <code>@NotBefore</code>• <code>@NotOnOrAfter</code>• <code>AudienceRestriction</code> |

Configuration for ODBC/JDBC Client Access

To enable logon using SAML bearer assertions, you must configure identity providers and then map them to the required database users. Two types of user mapping are supported:

- SAP HANA-based user mappings
An external identity is mapped to a database user explicitly in the SAP HANA in the user definition for each identity provider.

The corresponding assertion subject looks like this:

```
<NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">zgc2VLavgYy4hsohfYPM21</NameID>
```

Mapping to a database user's e-mail address is also possible. The corresponding assertion subject looks like this:

```
<NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
```

- Identity provider-based user mappings
The identity provider maps its users to SAP HANA database users and provides this information using the SPProvidedID attribute. This "wildcard" mapping is configured for a database user in the user definition using the keyword `ANY`.

For more information, see the CREATE USER statement in the *SAP HANA SQL and System Views Reference*.

The corresponding assertion subject looks like this:

```
<NameIDFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified" SPProvidedID="BILLG">zgc2VLavgYy4hsohfYPM21</NameID>
```

You can configure SAML identity providers and configure user mapping in the SAP HANA cockpit.

In addition, you must configure the trust store used to validate incoming SAML assertions against certificates signed by a trusted Certification Authority (CA). We recommend creating a certificate collection with the purpose **SAML** that contains the required certificates directly in the database. It is also possible to use a trust store located on the file system.

⚠ Caution

We recommend creating certificate collections for individual purposes in the database directly, rather than using trust stores (PSE) in the file system. By default, the same PSE in the file system is shared by all databases for all external communication channels (including HTTP) and certificate-based authentication. Different PSEs must be explicitly configured for tenant databases.

Configuration for HTTP Client Access via SAP HANA XS Classic

While you can configure SAML providers for ODBC/JDBC-based SAML authentication using the SAP HANA studio or SQL statements, you should always use the SAP HANA XS Administration Tool to configure SAML providers that will be used for HTTP access via the classic XS server.

You also use the SAP HANA XS Administration Tool to configure an SAP HANA system to act as an SAML service provider. For more information about maintaining SAML providers for HTTP access via the SAP HANA XS classic server, see the *SAP HANA Administration Guide*.

Related Information

[Secure Communication Between SAP HANA and JDBC/ODBC Clients \[page 43\]](#)

[Server-Side TLS/SSL Configuration Properties for External Communication \(JDBC/ODBC\) \[page 46\]](#)

[SAP Note 2284620](#)

7.4.3 Single Sign-On Using SAP Logon and Assertion Tickets

Users can be authenticated in SAP HANA by logon or assertion tickets issued to them when they log on to an SAP system configured to create tickets (for example, the SAP Web Application Server or Portal).

If you want to integrate an SAP HANA system into a landscape that uses logon or assertion tickets for user authentication, you must configure SAP HANA to accept logon/assertion tickets.

Trust Store Configuration

SAP HANA validates incoming logon/assertion tickets against certificates signed by a trusted Certification Authority (CA) stored in a dedicated trust store. This trust store must contain all root certificate(s) used to validate logon/assertion tickets. We recommend creating a certificate collection with the purpose **SAP LOGON** and the required certificates directly in the database.

It is also possible to use a trust store located in the file system. The default location of the trust store in the file system depends on the cryptographic library configured for SSL:

- `$SECUDIR/saplogon.pse` (CommonCryptoLib)

i Note

The `saplogon.pse` trust store is available automatically.

- `$HOME/.ssl/saplogon.pem` (OpenSSL)

i Note

Deprecated: OpenSSL is deprecated. You must migrate to CommonCryptoLib. For more information, see SAP Note 2093286.

⚠ Caution

By default, the same PSE in the file system is shared by all databases. Different PSEs must be explicitly configured for tenant databases.

You can configure the path to the trust store by setting the parameter `[authentication] saplogontickettruststore` in the `indexserver.ini` configuration file.

i Note

You must restart SAP HANA after you change this parameter.

i Note

This property is in the default configuration change blacklist (`multidb.ini`). This means that it cannot initially be changed in tenant databases. It must be changed from the system database. If appropriate for your scenario, you can remove this property from the change blacklist. For more information about how to edit the change blacklist, see the *SAP HANA Administration Guide*.

User Configuration

The user named in an incoming logon ticket must exist as a database user. The database user also must be configured for authentication using logon/assertion tickets. This can be done on the *User* page of the SAP HANA cockpit or in the user editor of the SAP HANA studio.

For more information about using logon tickets, see the SAP NetWeaver Library on SAP Help Portal.

Related Information

[Default Blacklisted System Properties in Tenant Databases \[page 361\]](#)

[SAP Note 2093286](#) 

7.4.4 Single Sign-On Using JSON Web Tokens

SAP HANA supports JSON Web Tokens (JWT) for user authentication in single sign-on environments.

The identity of users accessing the SAP HANA database from client applications can be authenticated by tokens issued by a trusted identity provider. The internal database user to which the external identity is mapped is used for authorization checks during the database session.

JWT can be implemented to authenticate users accessing the SAP HANA database from the following client applications:

- Database clients that access the SQL interface of the SAP HANA database directly
- Clients that connect to SAP HANA through the SAP HANA XS advanced server

→ Recommendation

To avoid replay attacks, we recommend that you set up secure communication between the individual components of the SAP HANA database and client connections using the Transport Layer Security (TLS) protocol when implementing JWT authentication.

JWT Structure

JSON Web Token (JWT) is an open standard. SAP HANA validates tokens according to the IETF standard, with the following restrictions and requirements.

Header

In the header part of the token, the "alg" (algorithm) claim, which specifies the hashing algorithm used to generate the signature, must be RS256.

The token header therefore looks like this:

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

Payload

SAP HANA evaluates the following claims in the payload part of the token:

- "iss" (issuer)
This claim is required to map the token to an identity provider configured in the SAP HANA database.
- "user_name" (user name)
This claim name is configurable. It is required for mapping the database user to an external user name. It is defined in the identity provider when it is created in SAP HANA, for example:

```
CREATE JWT PROVIDER my_jwt_provider WITH ISSUER 'http://example.com:8080/uaa/
oauth/token' CLAIM 'user_name' AS EXTERNAL IDENTITY;
```

- "nbf" (not before) and "exp" (expiration time)
These claims define the validity period of the token.

Sample Code

```
{
  "iss": "http://localhost:8080/uaa/oauth/token",
  "user_name": "testuser",
  "nbf": 1489571999,
  "exp": 1489572899,
}
```

Configuration

To enable logon using JSON Web Tokens, you must create identity providers and then map them to the required database users in SAP HANA. Two types of user mapping are supported:

- SAP HANA-based user mappings
Database users are mapped explicitly to their external identities in the identity provider.
- Identity provider-based user mappings
The identity provider maps its users to SAP HANA database users.

SAP HANA-based user mappings can be configured in the user definition, for example, using the SAP HANA cockpit.

In addition, you must configure the trust store used to validate incoming tokens against certificates signed by a trusted Certification Authority (CA). To do this, you create a certificate collection with the purpose JWT that contains the required certificates directly in the database.

Related Information

<https://tools.ietf.org/html/rfc7519> 

[User Administration and Authentication in SAP HANA XS Advanced \[page 302\]](#)

[Certificate Management in SAP HANA \[page 261\]](#)

7.5 LDAP User Authentication

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing directory services. If you use an LDAP-compliant directory server to manage users and their passwords, you can leverage LDAP-based authentication to access SAP HANA.

- [Overview \[page 114\]](#)
- [Authorization of LDAP-Authenticated Users \[page 115\]](#)
- [LDAP Authentication with Automatic User Creation \[page 116\]](#)
- [Configuration of LDAP Authentication \[page 117\]](#)

Overview

LDAP authentication can be implemented for users accessing SAP HANA directly via JDBC/ODBC database clients. The user is authenticated against an LDAP directory server using the user name and password provided by the client. The password is transmitted securely from the client to SAP HANA using a hybrid encryption-based protocol that uses a combination of symmetric and asymmetric encryption. The SAP HANA server then decrypts the password and uses it to authenticate the user with the LDAP server.

Using LDAP user passwords for authentication eliminates the need to manage user passwords and password policies in the SAP HANA database.

Note

Users configured for LDAP authentication cannot simultaneously be configured for local password authentication. Local password authentication must be disabled before a user can be configured for LDAP authentication, for example, using `ALTER USER <user_name> DISABLE PASSWORD.`

Caution

To protect the transmission of user passwords between SAP HANA and the LDAP server, you must secure communication between SAP HANA and the LDAP server using the TLS/SSL protocol. You can do this

while configuring the connection to the LDAP server in SAP HANA. See the section on secure communication between SAP HANA and an LDAP directory server.

⚠ Caution

As part of the authentication process, the SAP HANA server verifies the identity of the client but the client does not verify the identity of the SAP HANA server. For server authentication, you must enable TLS client-server communication. For more information, see the section on secure communication between SAP HANA and JDBC/ODBC clients.

Direct Connections with the SQL CONNECT Statement

When the CONNECT SQL statement is used to connect a user directly from a client program, the statement is independent of the client program. Behavior is solely dependent on the capabilities provided by the SAP HANA server and not client. The existing client network connection to SAP HANA is kept and a new session context is established, after successful authentication, using the specified user and password and the authentication method set for the specified user. The CONNECT SQL statement may be executed on an existing connection to authenticate a user with LDAP authentication even if the client program does not support LDAP authentication.

For example, a technical user MYTECHUSER is created and configured for local password authentication, and user MYLDAPUSER is created and configured for LDAP authentication. If an hdbsql client establishes a connection to SAP HANA using MYTECHUSER and then in that established session, the technical user issues the SQL statement `CONNECT MYLDAPUSER PASSWORD <ldappwd>`, then the current session context becomes user MYLDAPUSER authenticated using LDAP authentication method. The original network connection remains but the current session context is for MYLDAPUSER and the original client program may not directly support LDAP authentication method.

Client Requirements

LDAP-based authentication requires the version of the SAP HANA client available with SAP HANA 2.0 SPS 03. For more information about downloading and installing the SAP HANA client, see the *SAP HANA Client Installation and Update Guide*.

To authenticate with JDBC, the client must use JDK 8 or later and the JDK's policy files must support unlimited strength encryption. Failure to do this will result in the error "Cannot decrypt data: Illegal key size". To update the policy files:

- For JDK 8 update 150 and earlier, download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files for JDK/JRE 8 from Oracle.
- For JDK 8 update 151 and later as well as for JDK 9, edit the file `conf/security/java.security` and uncomment the line `crypto.policy=unlimited`.

Authorization of LDAP-Authenticated Users

The internal database user to be used for subsequent authorization checks in SAP HANA is determined during the logon process. With LDAP authentication, the internal database user name is same as the external identity used to log on. The following situations are possible:

- Database user exists and is configured for LDAP group authorization
If the database user exists and is configured for LDAP group authorization (authorization mode `LDAP`), it is verified that the authenticated user is a member of at least one LDAP group mapped to at least one SAP

HANA role. If this is the case, the user is logged on and the identified roles granted. For more information, see the section on LDAP group authorization for existing users.

- Database user exists and is configured for local authorization
If the database user exists and is configured for local authorization (authorization mode `LOCAL`), the user is logged on. Privileges and roles must be granted directly to the database user by a user administrator.
- Database user does not exist and the LDAP provider is configured for automatic user creation
If the database user does not exist and the LDAP provider is enabled to create database users in SAP HANA, the required database user is created. This is described in more detail in the next section.

LDAP Authentication with Automatic User Creation

If the database user does not exist and the LDAP provider is enabled to create database users in SAP HANA, it is verified that the authenticated user is a member of at least one LDAP group mapped to at least one SAP HANA role. If this is the case, a database user is created. Otherwise, logon fails.

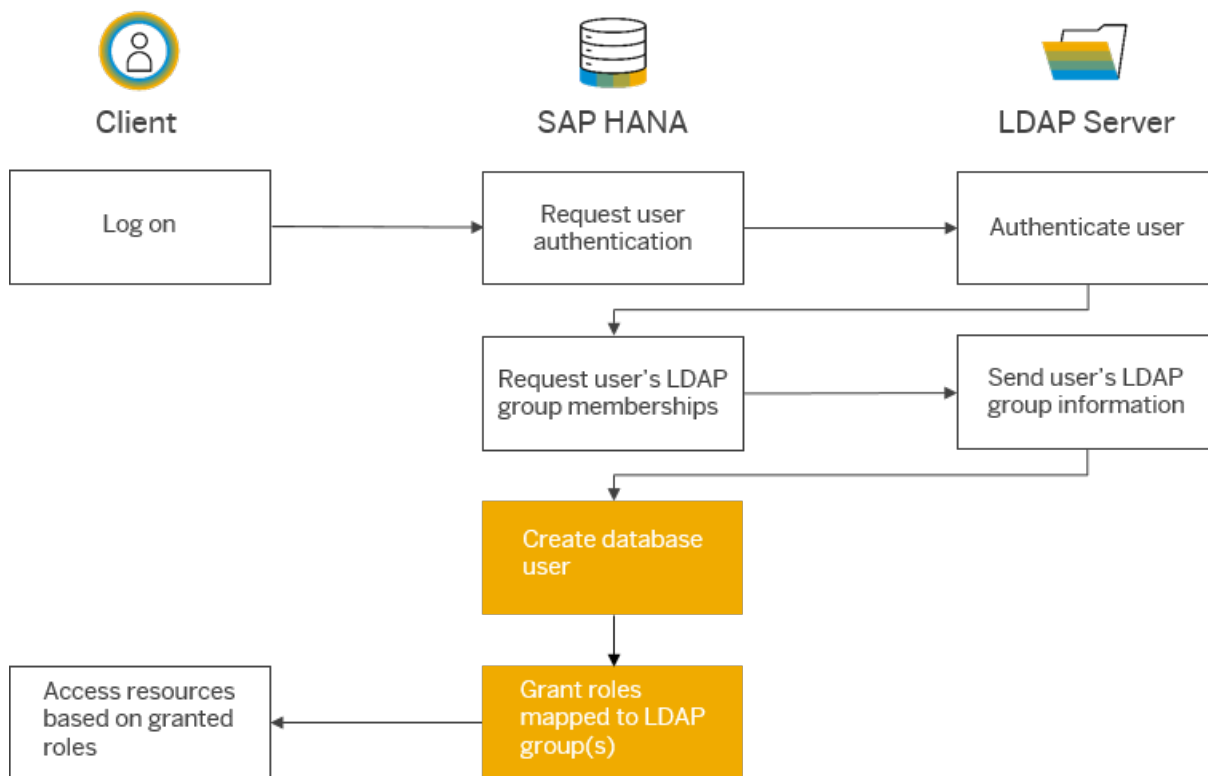
By default, a standard user is created, but it may be a restricted user if the LDAP provider is configured accordingly. The new database user is automatically configured for LDAP authentication and LDAP group authorization (authorization mode `LDAP`).

i Note

Automatic user creation is not supported in the following cases:

- In an active/active (read enabled) scenario from connections to secondary system
- For users connecting directly using the `CONNECT SQL` statement

The following figure illustrates how LDAP authentication with automatic user creation works.



LDAP Authentication with Automatic User Creation

Configuration of LDAP Authentication

To enable LDAP authentication, you must create and configure an LDAP provider in SAP HANA.

i Note

You can create several LDAP providers but only one can be in use at any time. This is the provider configured as the default.

With LDAP authentication, the external identity used to log on is same as the internal database user name. This user must be configured for LDAP authentication. This is done manually and/or automatically:

- As a user administrator, you can configure a new or existing database user for LDAP authentication in the user definition with either the CREATE USER or ALTER USER statement.

Sample Code

```
CREATE USER julie WITH IDENTITY FOR LDAP PROVIDER
```

i Note

A user enabled for LDAP authentication cannot be enabled for local password authentication. Enabling LDAP authentication for a user currently configured for password authentication results in an error. The reverse is also true.

i Note

The user SYSTEM cannot be enabled for LDAP authentication.

- Alternatively or additionally, you can enable the LDAP provider for automatic user creation. You specify this in the provider definition with either the CREATE LDAP PROVIDER or ALTER LDAP PROVIDER statement.

⇐ Sample Code

```
ALTER LDAP PROVIDER my_ldap_provider ENABLE USER CREATION FOR LDAP
```

See also the section on configuring an LDAP server connection for LDAP user authentication in the *SAP HANA Administration Guide*.

Related Information

[Secure Communication Between SAP HANA and an LDAP Directory Server \[page 57\]](#)

[Secure Communication Between SAP HANA and JDBC/ODBC Clients \[page 43\]](#)

[LDAP Group Authorization for Existing Users \[page 181\]](#)

8 SAP HANA Authorization

When a user accesses the SAP HANA database using a client interface (for example, ODBC, JDBC, or HTTP), his or her ability to perform database operations on database objects is determined by the privileges that he or she has been granted.

All the privileges granted to a user, either directly or indirectly through roles, are combined. This means that whenever a user tries to access an object, the system performs an authorization check on the user, the user's roles, and directly granted privileges. It is not possible to explicitly deny privileges. This means that the system does not need to check all the user's privileges. As soon as all requested privileges have been found, the system skips further checks and grants access.

[Privileges \[page 120\]](#)

Several privilege types are used in SAP HANA (system, object, analytic, package, and application).

[Database Roles \[page 158\]](#)

A database role is a collection of privileges that can be granted to either a database user or another role in runtime.

[Authorization in the Repository of the SAP HANA Database \[page 174\]](#)

The authorization concept of SAP HANA applies in the repository of the SAP HANA database.

[Cross-Database Authorization in Tenant Databases \[page 179\]](#)

Read-only queries between tenant databases are possible through the association of the requesting user with a remote identity on the remote database(s). Cross-database access is not enabled by default and must be configured before such user mappings can be set up.

[LDAP Group Authorization \[page 180\]](#)

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing directory services. If you use an LDAP-compliant directory server to manage users and their access to resources, you can leverage LDAP group membership to authorize users.

[Shared Business Authorizations in SAP HANA \[page 184\]](#)

The basic layer of authorization for ABAP-based SAP applications such as S/4 HANA is provided by "authorization objects" in the SAP NetWeaver Application Server for ABAP. It is possible to create analytic privileges in SAP HANA that reuse these authorizations for read access.

[Data Masking \[page 188\]](#)

Data masking represents an additional layer of access control that can be applied to tables and views. A column mask protects sensitive or confidential data in a particular column of a table or view by transforming the data in such a way that it is only visible partially or rendered completely meaningless for an unprivileged user, while still appearing real and consistent.

8.1 Privileges

Several privilege types are used in SAP HANA (system, object, analytic, package, and application).

| Privilege Type | Applicable To | Target User | Description |
|------------------|------------------|----------------------------|--|
| System privilege | System, database | Administrators, developers | <p>System privileges control general system activities. They are mainly used for administrative purposes, such as creating schemas, creating and changing users and roles, performing data backups, managing licenses, and so on.</p> <p>System privileges are also used to authorize basic repository operations.</p> <p>System privileges granted to users in a particular tenant database authorize operations in that database only. The only exception is the system privileges DATABASE ADMIN, DATABASE STOP, and DATABASE START. These system privileges can only be granted to users of the system database. They authorize the execution of operations on individual tenant databases. For example, a user with DATABASE ADMIN can create and drop tenant databases, change the database-specific properties in configuration (*.ini) files, and perform database-specific backups.</p> |

| Privilege Type | Applicable To | Target User | Description |
|--------------------|---|----------------------------|---|
| Object privilege | Database objects (schemas, tables, views, procedures and so on) | End users, technical users | <p>Object privileges are used to allow access to and modification of database objects, such as tables and views. Depending on the object type, different actions can be authorized (for example, SELECT, CREATE ANY, ALTER, DROP, and so on).</p> <p>Schema privileges are object privileges that are used to allow access to and modification of schemas and the objects that they contain.</p> <p>Source privileges are object privileges that are used to restrict access to and modification of remote data sources, which are connected through SAP HANA smart data access.</p> <p>Object privileges granted to users in a particular database authorize access to and modification of database objects in that database only. That is, unless cross-database access has been enabled for the user. This is made possible through the association of the requesting user with a remote identity on the remote database. For more information, see <i>Cross-Database Authorization in Tenant Databases</i> in the <i>SAP HANA Security Guide</i>.</p> |
| Analytic privilege | Analytic views | End users | <p>Analytic privileges are used to allow read access to data in SAP HANA information models (that is, analytic views, attribute views, and calculation views) depending on certain values or combinations of values. Analytic privileges are evaluated during query processing.</p> <p>Analytic privileges granted to users in a particular database authorize access to information models in that database only.</p> |

| Privilege Type | Applicable To | Target User | Description |
|-----------------------|---|---|--|
| Package privilege | Packages in the classic repository of the SAP HANA database | Application and content developers working in the classic SAP HANA repository | <p>Package privileges are used to allow access to and the ability to work in packages in the classic repository of the SAP HANA database.</p> <p>Packages contain design time versions of various objects, such as analytic views, attribute views, calculation views, and analytic privileges.</p> <p>Package privileges granted to users in a particular database authorize access to and the ability to work in packages in the repository of that database only.</p> <div data-bbox="927 725 1396 987" style="border: 1px solid #ccc; padding: 5px;"> <p>i Note</p> <p>With SAP HANA XS advanced, source code and web content are not versioned and stored in the SAP HANA database, so package privileges are not used in this context. For more information, see <i>Authorization in SAP HANA XS Advanced</i>.</p> </div> |
| Application privilege | SAP HANA XS classic applications | Application end users, technical users (for SQL connection configurations) | <p>Developers of SAP HANA XS classic applications can create application privileges to authorize user and client access to their application. They apply in addition to other privileges, for example, object privileges on tables.</p> <p>Application privileges can be granted directly to users or roles in runtime in the SAP HANA studio. However, it is recommended that you grant application privileges to roles created in the repository in design time.</p> <div data-bbox="927 1368 1396 1653" style="border: 1px solid #ccc; padding: 5px;"> <p>i Note</p> <p>With SAP HANA XS advanced, application privileges are not used. Application-level authorization is implemented using OAuth and authorization scopes and attributes. For more information, see <i>Authorization in SAP HANA XS Advanced</i>.</p> </div> |

i Note

An additional privilege type, privileges on users, can be granted to users. Privileges on users are SQL privileges that users can grant on their user. ATTACH DEBUGGER is the only privilege that can be granted on a user.

For example, User A can grant User B the privilege ATTACH DEBUGGER to allow User B debug SQLScript code in User A's session. User A is only user who can grant this privilege. Note that User B also needs the object privilege DEBUG on the relevant SQLScript procedure.

For more information, see the section on debugging procedures in the *SAP HANA Developer Guide*.

Related Information

[Cross-Database Authorization in Tenant Databases \[page 179\]](#)

[Authorization in SAP HANA XS Advanced \[page 313\]](#)

8.1.1 System Privileges

System privileges control general system activities.

System privileges are mainly used to authorize users to perform administrative actions, including:

- Creating and deleting schemas
- Managing users and roles
- Performing data backups
- Monitoring and tracing
- Managing licenses

System privileges are also used to authorize basic repository operations, for example:

- Importing and exporting content
- Maintaining delivery units (DU)

System privileges granted to users in a particular database authorize operations in that database only. The only exception is the system privileges DATABASE ADMIN, DATABASE STOP, and DATABASE START . These system privileges can only be granted to users of the system database. They authorize the execution of operations on individual tenant databases. For example, a user with DATABASE ADMIN can create and drop tenant databases, change the database-specific properties in configuration (*.ini) files, and perform database-specific or full-system data backups.

Related Information

[System Privileges \(Reference\) \[page 124\]](#)

8.1.1.1 System Privileges (Reference)

System privileges control general system activities.

General System Privileges

System privileges restrict administrative tasks. The following table describes the supported system privileges in an SAP HANA database.

| System Privilege | Description |
|------------------|---|
| ADAPTER ADMIN | Controls the execution of the following adapter-related statements: CREATE ADAPTER, DROP ADAPTER, and ALTER ADAPTER. It also allows access to the ADAPTERS and ADAPTER_LOCATIONS system views. |
| AGENT ADMIN | Controls the execution of the following agent-related statements: CREATE AGENT, DROP AGENT, and ALTER AGENT. It also allows access to the AGENTS and ADAPTER_LOCATIONS system views. |
| ATTACH DEBUGGER | Authorizes debugging across different user sessions. For example, userA can grant ATTACH DEBUGGER to userB to allow userB to debug a procedure in userA's session (userB still needs DEBUG privilege on the procedure, however). |
| AUDIT ADMIN | Controls the execution of the following auditing-related statements: CREATE AUDIT POLICY, DROP AUDIT POLICY, and ALTER AUDIT POLICY, as well as changes to the auditing configuration. It also allows access to the AUDIT_LOG, XSA_AUDIT_LOG, and ALL_AUDIT_LOG system views. |
| AUDIT OPERATOR | Authorizes the execution of the following statement: ALTER SYSTEM CLEAR AUDIT LOG. It also allows access to the AUDIT_LOG system view. |
| BACKUP ADMIN | Authorizes BACKUP and RECOVERY statements for defining and initiating backup and recovery procedures. It also authorizes changing system configuration options with respect to backup and recovery. |
| BACKUP OPERATOR | Authorizes the BACKUP statement to initiate a backup. |

| System Privilege | Description |
|--------------------------------------|---|
| CATALOG READ | Authorizes unfiltered access to the data in the system views that a user has already been granted the SELECT privilege on. Normally, the content of these views is filtered based on the privileges of the user. CATALOG READ does not allow a user to view system views on which they have not been granted the SELECT privilege. |
| CERTIFICATE ADMIN | Authorizes the changing of certificates and certificate collections that are stored in the database. |
| CLIENT PARAMETER ADMIN | Authorizes a user to override the value of the CLIENT parameter for a database connection or to overwrite the value of the \$\$client\$\$ parameter in a SQL query. |
| CREATE CLIENTSIDE ENCRYPTION KEYPAIR | Authorizes a user to create client-side encryption key pairs. |
| CREATE R SCRIPT | Authorizes the creation of a procedure by using the language R. |
| CREATE REMOTE SOURCE | Authorizes the creation of remote data sources by using the CREATE REMOTE SOURCE statement. |
| CREATE SCENARIO | Controls the creation of calculation scenarios and cubes (calculation database). |
| CREATE SCHEMA | Authorizes the creation of database schemas using the CREATE SCHEMA statement. |
| CREATE STRUCTURED PRIVILEGE | Authorizes the creation of structured (analytic privileges). Only the owner of the privilege can further grant or revoke that privilege to other users or roles. |
| CREDENTIAL ADMIN | Authorizes the use of the statements CREATE CREDENTIAL, ALTER CREDENTIAL, and DROP CREDENTIAL. |
| DATA ADMIN | Authorizes reading all data in the system views. It also enables execution of Data Definition Language (DDL) statements in the SAP HANA database. A user with this privilege cannot select or change data in stored tables for which they do not have access privileges, but they can drop tables or modify table definitions. |
| DATABASE ADMIN | Authorizes all statements related to tenant databases, such as CREATE, DROP, ALTER, RENAME, BACKUP, and RECOVERY. |
| DATABASE START | Authorizes a user to start any database in the system and to select from the M_DATABASES view. |

| System Privilege | Description |
|------------------------------------|--|
| DATABASE STOP | Authorizes a user to stop any database in the system and to select from the M_DATABASES view. |
| DROP CLIENTSIDE ENCRYPTION KEYPAIR | Authorizes a user to drop other users' client-side encryption key pairs. |
| ENCRYPTION ROOT KEY ADMIN | Authorizes all statements related to management of root keys: Allows access to the system views pertaining to encryption (for example, ENCRYPTION_ROOT_KEYS, M_ENCRYPTION_OVERVIEW, M_PERSISTENCE_ENCRYPTION_STATUS, M_PERSISTENCE_ENCRYPTION_KEYS, and so on). |
| EXPORT | Authorizes EXPORT to a file on the SAP HANA server. The user must also have the SELECT privilege on the source tables to be exported. |
| EXTENDED STORAGE ADMIN | Authorizes the management of SAP HANA dynamic tiering and the creation of extended storage. |
| IMPORT | Authorizes the import activity in the database using the IMPORT statements. The user must also have the INSERT privilege on the target tables to be imported. |
| INIFILE ADMIN | Authorizes making changes to system settings. |
| LDAP ADMIN | Authorizes the use of the CREATE ALTER DROP VALIDATE LDAP PROVIDER statements. |
| LICENSE ADMIN | Authorizes the use of the SET SYSTEM LICENSE statement to install a new license. |
| LOG ADMIN | Authorizes the use of the ALTER SYSTEM LOGGING [ON OFF] statements to enable or disable the log flush mechanism. |
| MONITOR ADMIN | Authorizes the use of the ALTER SYSTEM statements for events. |
| OPTIMIZER ADMIN | Authorizes the use of the ALTER SYSTEM statements concerning SQL PLAN CACHE and ALTER SYSTEM UPDATE STATISTICS statements, which influence the behavior of the query optimizer. |

| System Privilege | Description |
|---------------------------|--|
| RESOURCE ADMIN | Authorizes statements concerning system resources (for example, the ALTER SYSTEM RECLAIM DATAVOLUME and ALTER SYSTEM RESET MONITORING VIEW statements). It also authorizes many of the statements available in the Management Console. |
| ROLE ADMIN | <p>Authorizes the creation and deletion of roles by using the CREATE ROLE and DROP ROLE statements. It also authorizes the granting and revoking of roles by using the GRANT and REVOKE statements.</p> <p>Activated repository roles, meaning roles whose creator is the predefined user <code>_SYS_REPO</code>, can neither be granted to other roles or users nor dropped directly. Not even users with the ROLE ADMIN privilege can do so. Check the documentation concerning activated objects.</p> |
| SAVEPOINT ADMIN | Authorizes the execution of a savepoint using the ALTER SYSTEM SAVEPOINT statement. |
| SCENARIO ADMIN | Authorizes all calculation scenario-related activities (including creation). |
| SERVICE ADMIN | Authorizes the ALTER SYSTEM [START CANCEL RECONFIGURE] statements for administering system services of the database. |
| SESSION ADMIN | Authorizes the ALTER SYSTEM commands concerning sessions to stop or disconnect a user session or to change session variables. |
| SSL ADMIN | Authorizes the use of the SET...PURPOSE SSL statement. It also allows access to the PSES system view. |
| STRUCTUREDPRIVILEGE ADMIN | Authorizes the creation, reactivation, and dropping of structured (analytic) privileges. |
| TENANT ADMIN | Authorizes the tenant operations performed by the ALTER SYSTEM [RESUME SUSPEND] TENANT statements. |
| TABLE ADMIN | Authorizes LOAD, UNLOAD and MERGE of tables and table placement. |
| TRACE ADMIN | Authorizes the use of the ALTER SYSTEM...TRACES statements for operations on database trace files and authorizes changing trace system settings. |
| TRUST ADMIN | Authorizes the use of statements to update the trust store. |

| System Privilege | Description |
|---------------------------|--|
| USER ADMIN | Authorizes the creation and modification of users by using the CREATE ALTER DROP USER statements. |
| VERSION ADMIN | Authorizes the use of the ALTER SYSTEM RECLAIM VERSION SPACE statement of the multi-version concurrency control (MVCC) feature. |
| WORKLOAD ADMIN | Authorizes execution of the workload class and mapping statements (for example, CREATE ALTER DROP WORKLOAD CLASS, and CREATE ALTER DROP WORKLOAD MAPPING). |
| WORKLOAD ANALYZE ADMIN | Used by the Analyze Workload, Capture Workload, and Replay Workload applications when performing workload analysis. |
| WORKLOAD CAPTURE ADMIN | Authorizes access to the monitoring view M_WORKLOAD_CAPTURES to see the current status of capturing and captured workloads, as well of execution of actions with the WORKLOAD_CAPTURE procedure. |
| WORKLOAD REPLAY ADMIN | Authorizes access to the monitoring views M_WORKLOAD_REPLAY_PREPROCESSES and M_WORKLOAD_REPLAYS to see current status of preprocessing, preprocessed, replaying, and replayed workloads, as well as the execution of actions with the WORKLOAD_REPLAY procedure. |
| <identifier>.<identifier> | Components of the SAP HANA database can create new system privileges. These privileges use the component-name as the first identifier of the system privilege and the component-privilege-name as the second identifier. |

Repository System Privileges

i Note

The following privileges authorize actions on individual packages in the SAP HANA repository, used in the SAP HANA Extended Services (SAP HANA XS) classic development model. With SAP HANA XS advanced, source code and web content are no longer versioned and stored in the repository of the SAP HANA database.

| System Privilege | Description |
|------------------|---|
| REPO.EXPORT | Authorizes the export of delivery units for example |

| System Privilege | Description |
|----------------------------------|---|
| REPO.IMPORT | Authorizes the import of transport archives |
| REPO.MAINTAIN_DELIVERY_UNITS | Authorizes the maintenance of delivery units (DU, DU vendor and system vendor must be the same) |
| REPO.WORK_IN_FOREIGN_WORKSPACE | Authorizes work in a foreign inactive workspace |
| REPO.CONFIGURE | Authorize work with SAP HANA Change Recording, which is part of SAP HANA Application Lifecycle Management |
| REPO.MODIFY_CHANGE | |
| REPO.MODIFY_OWN_CONTRIBUTION | |
| REPO.MODIFY_FOREIGN_CONTRIBUTION | |

Related Information

[Developer Authorization in the Repository \[page 175\]](#)

8.1.2 Object Privileges

Object privileges are SQL privileges that are used to allow access to and modification of database objects.

For each SQL statement type (for example, SELECT, UPDATE, or CALL), a corresponding object privilege exists. If a user wants to execute a particular statement on a simple database object (for example, a table), he or she must have the corresponding object privilege for either the actual object itself, or the schema in which the object is located. This is because the schema is an object type that contains other objects. A user who has object privileges for a schema automatically has the same privileges for all objects currently in the schema and any objects created there in the future.

Object privileges are not only grantable for database catalog objects such as tables, views and procedures. Object privileges can also be granted for non-catalog objects such as development objects in the repository of the SAP HANA database.

Initially, the owner of an object and the owner of the schema in which the object is located are the only users who can access the object and grant object privileges on it to other users.

An object can therefore be accessed only by the following users:

- The owner of the object
- The owner of the schema in which the object is located
- Users to whom the owner of the object has granted privileges
- Users to whom the owner of the parent schema has granted privileges

Caution

The database owner concept stipulates that when a database user is deleted, all objects created by that user and privileges granted to others by that user are also deleted. If the owner of a schema is deleted, all

objects in the schema are also deleted even if they are owned by a different user. All privileges on these objects are also deleted.

i Note

The owner of a table can change its ownership with the `ALTER TABLE` SQL statement. In this case, the new owner becomes the grantor of all privileges on the table granted by the original owner. The original owner is also automatically granted all privileges for the table with the new owner as grantor. This ensures that the original owner can continue to work with the table as before.

Authorization Check on Objects with Dependencies

The authorization check for objects defined on other objects (that is, stored procedures and views) is more complex. In order to be able to access an object with dependencies, both of the following conditions must be met:

- The user trying to access the object must have the relevant object privilege on the object as described above.
- The user who created the object must have the required privilege on all underlying objects **and** be authorized to grant this privilege to others.

If this second condition is not met, only the owner of the object can access it. He cannot grant privileges on it to any other user. This cannot be circumvented by granting privileges on the parent schema instead. Even if a user has privileges on the schema, he will still not be able to access the object.

i Note

This applies to procedures created in DEFINER mode only. This means that the authorization check is run against the privileges of the user who created the object, not the user accessing the object. For procedures created in INVOKER mode, the authorization check is run against the privileges of the accessing user. In this case, the user must have privileges not only on the object itself but on all objects that it uses.

→ Tip

The SAP HANA studio provides a graphical feature, the authorization dependency viewer, to help troubleshoot authorization errors for object types that typically have complex dependency structures: stored procedures and calculation views.

Related Information

[Object Privileges \(Reference\) \[page 131\]](#)

[Cross-Database Authorization in Tenant Databases \[page 179\]](#)

8.1.2.1 Object Privileges (Reference)

Object privileges are used to allow access to and modification of database objects, such as tables and views.

The following table describes the supported object privileges in an SAP HANA database.

| Object Privilege | Command Types | Applies to | Privilege Description |
|--------------------------|---------------|---|--|
| ALL PRIVILEGES | DDL & DML | <ul style="list-style-type: none"> Schemas Tables Views | <p>This privilege is a collection of all Data Definition Language (DDL) and Data Manipulation Language (DML) privileges that the grantor currently possesses and is allowed to grant further. The privilege it grants is specific to the particular object being acted upon.</p> <p>This privilege collection is dynamically evaluated for the given grantor and object.</p> |
| ALTER | DDL | <ul style="list-style-type: none"> Schemas Tables Views Functions/procedures | Authorizes the ALTER statement for the object. |
| CREATE ANY | DDL | <ul style="list-style-type: none"> Schemas Tables Views Sequences Functions/procedures Remote sources Graph workspaces | Authorizes all CREATE statements for the object. |
| CREATE VIRTUAL FUNCTION | DDL | <ul style="list-style-type: none"> Remote sources | Authorizes creation of virtual functions (the REFERENCES privilege is also required). |
| CREATE VIRTUAL PROCEDURE | DDL | <ul style="list-style-type: none"> Remote sources | Authorizes creation of virtual procedure to create and run procedures on a remote source. |
| CREATE VIRTUAL PACKAGE | DDL | <ul style="list-style-type: none"> Schemas | Authorizes creation of virtual packages that can be run on remote sources. |

| Object Privilege | Command Types | Applies to | Privilege Description |
|------------------------|---------------|---|--|
| CREATE VIRTUAL TABLE | DDL | <ul style="list-style-type: none"> Remote sources | Authorizes the creation of proxy tables pointing to remote tables from the source entry. |
| CREATE TEMPORARY TABLE | DDL | <ul style="list-style-type: none"> Schemas | Authorizes the creation of a temporary local table, which can be used as input for procedures, even if the user does not have the CREATE ANY privilege for the schema. |
| DEBUG | DML | <ul style="list-style-type: none"> Schemas Calculation Views Functions/procedures | Authorizes debug functionality for the procedure or calculation view or for the procedures and calculation views of a schema. |
| DEBUG MODIFY | DDL | <ul style="list-style-type: none"> Functions/procedures | For internal use only. |
| DELETE | DML | <ul style="list-style-type: none"> Schemas Tables Views Functions/procedures | <p>Authorizes the DELETE and TRUNCATE statements for the object.</p> <p>While DELETE applies to views, it only applies to updatable views (that is, views that do not use a join, do not contain a UNION, and do not use aggregation).</p> |
| DROP | DDL | <ul style="list-style-type: none"> Schemas Tables Views Sequences Functions/procedures Remote sources Graph workspaces | Authorizes the DROP statements for the object. |
| EXECUTE | DML | <ul style="list-style-type: none"> Schemas Functions/procedures | Authorizes the execution of a SQLScript function or a database procedure by using the CALLS or CALL statement respectively. It also allows a user to execute a virtual function. |

| Object Privilege | Command Types | Applies to | Privilege Description |
|---------------------|---------------|---|---|
| INDEX | DDL | <ul style="list-style-type: none"> • Schemas • Tables | <p>Authorizes the creation, modification, or dropping of indexes for the object.</p> |
| INSERT | DML | <ul style="list-style-type: none"> • Schemas • Tables • Views | <p>Authorizes the INSERT statement for the object.</p> <p>The INSERT and UPDATE privilege are both required on the object to allow the REPLACE and UPSERT statements to be used.</p> <p>While INSERT applies to views, it only applies to updatable views (views that do not use a join, do not contain a UNION, and do not use aggregation).</p> |
| REFERENCES | DDL | <ul style="list-style-type: none"> • Schemas • Tables | <p>Authorizes the usage of all tables in this schema or this table in a foreign key definition, or the usage of a personal security environment (PSE). It also allows a user to reference a virtual function package.</p> |
| SELECT | DML | <ul style="list-style-type: none"> • Schemas • Tables • Views • Sequences • Graph workspaces | <p>Authorizes the SELECT statement for the object or the usage of a sequence.</p> <p>When selection from system-versioned tables, users must have SELECT on both the table and its associated history table.</p> |
| SELECT CDS METADATA | DML | <ul style="list-style-type: none"> • Schemas • Tables | <p>Authorizes access to CDS metadata from the catalog.</p> |
| SELECT METADATA | DML | <ul style="list-style-type: none"> • Schemas • Tables | <p>Authorizes access to the complete metadata of all objects in a schema (including procedure and view definitions), including objects that may be located in other schemas.</p> |

| Object Privilege | Command Types | Applies to | Privilege Description |
|--------------------|---------------|--|--|
| TRIGGER | DDL | <ul style="list-style-type: none"> Schemas Tables | Authorizes the CREATE TRIGGER/DROP TRIGGER statement for the specified table or the tables in the specified schema. |
| UNMASKED | DML | <ul style="list-style-type: none"> Schemas Views Tables | Authorizes access to masked data in user-defined views and tables. This privilege is required to view the original data in views and tables that are defined by using the WITH MASK clause. |
| UPDATE | DML | <ul style="list-style-type: none"> Schemas Tables Views | While UPDATE applies to views, it only applies to updatable views (views that do not use a join, do not contain a UNION, and do not use aggregation). |
| USERGROUP OPERATOR | DML | <ul style="list-style-type: none"> User groups | <p>Authorizes a user to change the settings for a user group, and to add and remove users to/from a user group.</p> <p>Users with the USERGROUP OPERATOR privilege can also create and drop users, but only within the user group they have the USERGROUP OPERATOR privilege on (CREATE USER <user_name> SET USERGROUP <usergroup_name>).</p> <p>A user can have the USERGROUP OPERATOR privilege on more than one user group, and a user group can have more than one user with the USERGROUP OPERATOR privilege on it.</p> |

| Object Privilege | Command Types | Applies to | Privilege Description |
|--|---------------|------------|--|
| <code><identifier>.<identifier></code> | DDL | | Components of the SAP HANA database can create new object privileges. These privileges use the component-name as first identifier of the system privilege and the component-privilege-name as the second identifier. |

8.1.3 Analytic Privileges

Analytic privileges grant different users access to different portions of data in the same view based on their business role. Within the definition of an analytic privilege, the conditions that control which data users see is either contained in an XML document or defined using SQL.

Standard object privileges (`SELECT`, `ALTER`, `DROP`, and so on) implement coarse-grained authorization at object level only. Users either have access to an object, such as a table, view or procedure, or they don't. While this is often sufficient, there are cases when access to data in an object depends on certain values or combinations of values. Analytic privileges are used in the SAP HANA database to provide such fine-grained control at row level of which data individual users can see within the same view.

❁ Example

Sales data for all regions are contained within one analytic view. However, regional sales managers should only see the data for their region. In this case, an analytic privilege could be modeled so that they can all query the view, but only the data that each user is authorized to see is returned.

Creation of Analytic Privileges

Although analytic privileges can be created directly as catalog objects in runtime, we recommend creating them as design-time objects that become catalog objects on deployment (database artifact with file suffix `.hdbanalyticprivilege`). In an SAP HANA XS classic environment, analytic privileges are created in the built-in repository of the SAP HANA database using either the SAP HANA Web Workbench or the SAP HANA studio. In an SAP HANA XS advanced environment, they are created using the SAP Web IDE and deployed using SAP HANA deployment infrastructure (SAP HANA DI).

i Note

HDI supports only SQL-based analytic privileges (see below). Furthermore, due to the container-based model of HDI, where each container corresponds to a database schema, analytic privileges created in HDI are schema specific.

XML- Versus SQL-Based Analytic Privileges

Before you implement row-level authorization using analytic privileges, you need to decide which type of analytic privilege is suitable for your scenario. In general, SQL-based analytic privileges allow you to more easily formulate complex filter conditions using sub-queries that might be cumbersome to model using XML-based analytic privileges.

→ Recommendation

SAP recommends the use of SQL-based analytic privileges. Using the *SAP HANA Modeler* perspective of the SAP HANA studio, you can migrate XML-based analytic privileges to SQL-based analytic privileges. For more information, see the SAP HANA Modeling Guide (For SAP HANA Studio).

i Note

As objects created in the repository, XML-based analytic privileges are deprecated as of SAP HANA SPS 02. For more information, see SAP Note 2465027.

The following are the main differences between XML-based and SQL-based analytic privileges:

| Feature | SQL-Based Analytic Privileges | XML-Based Analytic Privileges |
|---|-------------------------------|-------------------------------|
| Control of read-only access to SAP HANA information models: <ul style="list-style-type: none"> • Attribute views • Analytic views • Calculation views | Yes | Yes |
| Control of read-only access to SQL views | Yes | No |
| Control of read-only access to database tables | No | No |
| Design-time modeling using the SAP HANA Web-based Workbench or the <i>SAP HANA Modeler</i> perspective of the SAP HANA studio | Yes | Yes |
| Design-time modeling using the SAP Web IDE for SAP HANA | Yes | No |
| Transportable | Yes | Yes |
| HDI support | Yes | No |
| Complex filtering | Yes | No |

i Note

This corresponds to development in an SAP HANA XS classic environment using the SAP HANA repository.

i Note

This corresponds to development in an SAP HANA XS advanced environment using HDI.

Enabling an Authorization Check Based on Analytic Privileges

All column views modeled and activated in the SAP HANA modeler and the SAP HANA Web-based Development Workbench automatically enforce an authorization check based on analytic privileges. XML-based analytic privileges are selected by default, but you can switch to SQL-based analytic privileges.

Column views created using SQL must be explicitly registered for such a check by passing the relevant parameter:

- `REGISTERVIEWFORAPCHECK` for a check based on XML-based analytic privileges
- `STRUCTURED PRIVILEGE CHECK` for a check based on SQL-based analytic privileges

SQL views must always be explicitly registered for an authorization check based on analytic privileges by passing the `STRUCTURED PRIVILEGE CHECK` parameter.

i Note

It is not possible to enforce an authorization check on the same view using both XML-based and SQL-based analytic privileges. However, it is possible to build views with different authorization checks on each other.

Related Information

[SAP Note 2465027](#)

8.1.3.1 Structure of SQL-Based Analytic Privileges

An analytic privilege consists of a set of restrictions against which user access to a particular attribute view, analytic view, calculation view, or SQL view is verified. In an SQL-based analytic privilege, these restrictions are specified as filter conditions that are fully SQL based.

SQL-based analytic privileges are created using the `CREATE STRUCTURED PRIVILEGE` statement:

```
CREATE STRUCTURED PRIVILEGE <privilege_name> FOR <action> ON <view_name>  
<filter_condition>
```

The `FOR` clause is used to restrict the type of access (only the `SELECT` action is supported). The `ON` clause is used to restrict access to one or more views with the same filter attributes.

The `<filter condition>` parameter is used to restrict the data visible to individual users. The following methods of specifying filter conditions are possible:

- Fixed filter (`WHERE`) clause
- Dynamically generated filter (`CONDITION PROVIDER`) clause

Fixed Filter Clauses

A **fixed filter clause** consists of an WHERE clause that is specified in the definition of the analytic privilege itself.

You can express fixed filter conditions freely using SQL, including subqueries.

By incorporating built-in SQL functions into the subqueries, in particular SESSION_USER, you can define an even more flexible filter condition.

❁ Example

```
country IN (SELECT a.country FROM authorizationtable a WHERE SESSION_USER=
a.user_name)
```

i Note

A **calculation view** cannot be secured using an SQL-based analytic privilege that contains a complex filter condition if the view is defined on top of analytic and/or attributes views that themselves are secured with an SQL-based analytic privilege with a complex filter condition.

→ Remember

If you use a subquery, you (the creating user) must have the required privileges on the database objects (tables and views) involved in the subquery.

Comparative conditions can be nested and combined using AND and OR (with corresponding brackets).

→ Tip

To create an analytic privilege that allows either access to all data or no data in a view, set a fixed filter condition such as 1=1 or 1!=1.

Dynamically Generated Filter Clauses

With a dynamically generated filter clause, the WHERE clause that specifies the filter condition is generated every time the analytic privilege is evaluated. This is useful in an environment in which the filter clause changes very dynamically. The filter condition is determined by a procedure specified in the CONDITION PROVIDER clause, for example:

≡ Sample Code

```
CREATE STRUCTURED PRIVILEGE dynamic_ap FOR SELECT ON schema1.v1 CONDITION PROVIDER
schema2.procedure1;
```

Procedures in the CONDITION PROVIDER clause must have the following properties:

- They must have the security mode DEFINER.
- They must be read-only procedures.

- They must have a predefined signature. Here, the following conditions apply:
 - No input parameter
 - Only one output parameter for the filter condition string of string type NVARCHAR, VARCHAR, CLOB, or NCLOB
While VARCHAR and NVARCHAR have length limitations of 5000 characters, CLOB and NCLOB can be used to accommodate longer filter strings.
- The procedure may only return conditions expressed with the following operators:
 - =, <=, <, >, >=
 - LIKE
 - BETWEEN
 - IN
 - NOT (...)
 - !=

A complex filter condition, that is a subquery, may not be returned.

→ Tip

A procedure that returns the filter condition `1=1` or `1>1` can be used to create an analytic privilege that allows access to all data or no data in a view.

- The procedure must be executable by `_SYS_REPO`, that is, either `_SYS_REPO` must be the owner of the procedure or the owner of the procedure has all privileges on the underlying tables/views with `GRANT OPTION` and has granted the `EXECUTE` privilege on the procedure to the `_SYS_REPO` user.

If errors occur in procedure execution, the user receives a `Not authorized` error, even if he has the analytic privileges that would grant access.

8.1.3.1.1 Examples: Securing Views Using SQL-Based Analytic Privileges

Use the `CREATE STRUCTURED PRIVILEGE` statement to create SQL-based analytic privileges for different scenarios.

Context

The examples provided here take you through the following scenarios:

- [Example 1: Securing a column view using an SQL-based analytic privilege with a fixed filter clause \[page 140\]](#)
- [Example 2: Securing an SQL view using an SQL-based analytic privilege with a complex filter clause \(subquery\) \[page 141\]](#)
- [Example 3: Securing a column view using an SQL-based analytic privilege with a dynamically generated filter clause \[page 143\]](#)

i Note

The analytic privileges in these example are created using the `CREATE STRUCTURED PRIVILEGE` statement. Under normal circumstances, you create SQL-based analytic privileges using the SAP HANA

Web-based Development Workbench. Analytic privileges created using CREATE STRUCTURED PRIVILEGE are not owned by the user _SYS_REPO. They can be granted and revoked only by the actual database user who creates them.

Example 1: Secure a Column View Using an SQL-Based Analytic Privilege with a Fixed Filter Clause

Prerequisites

The database user TABLEOWNER has set up a calculation scenario based on the table SALES_TABLE, which contains the data to be protected.

Context

All sales data is contained in a single view. You want to restrict user access so that sales managers can see only information about the product "car" in the sales region UK and Germany. You want to do this by creating an analytic privilege with a fixed filter clause.

A fixed filter clause consists of an SQL WHERE clause that is specified in the definition of the analytic privilege itself.

Procedure

1. Create the view containing the sales data:

```
CREATE COLUMN VIEW "TABLEOWNER"."VIEW_SALES" TYPE CALCULATION WITH PARAMETERS
('PARENTCALCINDEXSCHEMA'='TABLEOWNER',
 'PARENTCALCINDEX'='CALCSCEN_SALES',
 'PARENTCALCNODE'='SALES_TABLE',
 'REGISTerviewFORAPCHECK'='0') STRUCTURED PRIVILEGE CHECK
;
```

Note

You can see above that the authorization check using XML-based analytic privileges is disabled with 'REGISTerviewFORAPCHECK'='0', while the authorization check using SQL-based analytic privileges is enabled with STRUCTURED PRIVILEGE CHECK. Both checks cannot be enabled at the same time.

2. Create the analytic privilege:

```
CREATE STRUCTURED PRIVILEGE AP_SALES_1 FOR SELECT
ON TABLEOWNER.VIEW_SALES
WHERE REGION IN ('DE','UK')
OR PRODUCT = 'CAR'
```

;

→ Remember

When specifying filters, remember the following:

- You can specify only the SELECT action in the FOR clause.
- You can specify one or more views with the same filter attributes in the ON clause
- You can specify comparative conditions between attributes and constant values using only the following operators:
 - =, <=, <, >, >=
 - LIKE
 - BETWEEN
 - IN
- You can create complex filter conditions by including SQL statements as subqueries inside the WHERE clause. Example 2 illustrates how you do this. But remember: A **calculation view** cannot be secured using an SQL-based analytic privilege that contains a complex filter condition if the view is defined on top of analytic and/or attributes views that themselves are secured with an SQL-based analytic privilege with a complex filter condition.
Also remember that if you use a subquery, you must have the required privileges on the database objects (tables and views) involved in the subquery.

3. Grant the SELECT privilege on the view TABLEOWNER.VIEW_SALES to the relevant users/roles:

```
GRANT SELECT on TABLEOWNER.VIEW_SALES to <SALES MANAGERS>;
```

→ Remember

Only the view owner or a user who has the SELECT privilege WITH GRANT OPTION on the view can perform the grant.

4. Grant the analytic privilege to the relevant users/roles:

```
GRANT STRUCTURED PRIVILEGE AP_SALES_1 TO <SALES MANAGERS>;
```

→ Remember

Only the owner of the analytic privilege can grant it.

Example 2: Secure an SQL View Using an SQL-Based Analytic Privilege with a Complex Filter Clause (Subquery)

Prerequisites

The database user TABLEOWNER has created a table TABLEOWNER.SALES, which contains the data to be protected.

Context

All sales data is contained in a single view. You want to restrict access of user MILLER so that he can see only product information from the year 2008. You want to do this by creating an analytic privilege with a complex filter clause.

With a complex filter clause, the SQL WHERE clause that specifies the filter condition includes an SQL statement, or a subquery. This allows you to create complex filter conditions to control which data individual users see.

Procedure

1. Create the view containing the sales data which needs to be secured:

```
CREATE VIEW "VIEWOWNER"."ROW_VIEW_SALES_ON_SALES" AS SELECT
  * FROM "TABLEOWNER"."SALES" WITH STRUCTURED PRIVILEGE CHECK
;
```

→ Remember

The user creating the view must have the SELECT privilege WITH GRANT OPTION on the table TABLEOWNER.SALES.

2. Create the table containing user-specific authorization data:

```
CREATE COLUMN TABLE "VIEWOWNER"."AUTHORIZATION_VALUES" ("VALUE" VARCHAR(256),
  "USER_NAME" VARCHAR(20));
```

3. Insert authorization information for user MILLER:

```
INSERT INTO "VIEWOWNER"."AUTHORIZATION_VALUES" VALUES('2008', 'MILLER');
```

4. Create the analytic privilege using a subquery as the condition provider:

```
CREATE STRUCTURED PRIVILEGE AP_ROW_VIEW_SALES_ON_SALES FOR SELECT
  ON "VIEWOWNER"."ROW_VIEW_SALES_ON_SALES"
WHERE (CURRENT_DATE BETWEEN 2015-01-01 AND 2015-01-11) AND YEAR IN (SELECT
  VALUE FROM VIEWOWNER.AUTHORIZATION_VALUES WHERE USER_NAME = SESSION_USER)
;
```

→ Remember

- Subqueries allow you to create complex filter conditions, but remember: A **calculation view** cannot be secured using an SQL-based analytic privilege that contains a complex filter condition if the view is defined on top of analytic and/or attributes views that themselves are secured with an SQL-based analytic privilege with a complex filter condition.
- The user creating the analytic privilege must have the SELECT privilege on the objects involved in the subquery, in this case table VIEWOWNER.AUTHORIZATION_VALUES.
- The session user is the database user who is executing the query to access a secured view. This is therefore the user whose privileges must be checked. For this reason, the table containing the authorization information needs a column to store the user name so that the subquery can filter on this column using the SQL function SESSION_USER.

⚠ Caution

Do not map the executing user to the application user. The application user is unreliable because it is controlled by the client application. For example, it may set the application user to a technical user or it may not set it at all. In addition, the trustworthiness of the client application cannot be guaranteed.

5. Grant the SELECT privilege on the view VIEWOWNER.ROW_VIEW_SALES_ON_SALES to user MILLER.

```
GRANT SELECT ON "VIEWOWNER"."ROW_VIEW_SALES_ON_SALES" TO MILLER;
```

→ Remember

Only the view owner or a user who has the SELECT privilege WITH GRANT OPTION on the view can perform the grant.

6. Grant the analytic privilege to user MILLER.

```
GRANT STRUCTURED PRIVILEGE AP_ROW_SALES_ON_SALES TO MILLER;
```

→ Remember

Only the owner of the analytic privilege can grant it.

Example 3: Secure a Column View Using an SQL-Based Analytic Privilege with a Dynamically Generated Filter Clause

Prerequisites

The database user TABLEOWNER has set up a calculation scenario based on the table SALES_TABLE, which contains the data to be protected.

Context

All sales data is contained in a single view. You want to restrict access of user ADAMS so that he can see only information about cars bought by customer Company A or bikes sold in 2006. You want to do this by creating an analytic privilege with a dynamically generated filter clause.

With a dynamically generated filter clause, the SQL WHERE clause that specifies the filter condition is generated every time the analytic privilege is evaluated. This is useful in an environment in which the filter clause changes very dynamically.

Procedure

1. Create the view containing the sales data:

```
CREATE COLUMN VIEW "TABLEOWNER"."VIEW SALES" TYPE CALCULATION WITH PARAMETERS
('PARENTCALCINDEXSCHEMA'='TABLEOWNER',
 'PARENTCALCINDEX'='CALCSCEN_SALES',
 'PARENTCALCNODE'='SALES_TABLE',
 'REGISTERVIEWFORAPCHECK'='0') STRUCTURED PRIVILEGE CHECK
;
```

2. Create a table containing user-specific filter strings:

```
CREATE COLUMN TABLE "AUTHORIZATION"."AUTHORIZATION_FILTERS" ("FILTER"
VARCHAR(256),
 "USER_NAME" VARCHAR(20))
;
```

3. Create an authorization filter for user ADAMS:

```
INSERT
INTO "AUTHORIZATION"."AUTHORIZATION_FILTERS" VALUES ('(CUSTOMER=''Company A''
AND PRODUCT=''Car'') OR (YEAR=''2006'' AND PRODUCT=''Bike'')',
 'ADAMS')
;
```

→ Remember

Filters containing comparative conditions must be defined as specified in example 1.

4. Create the database procedure that provides the filter clause for the analytic privilege and grant it to user `_SYS_REPO`:

```
CREATE PROCEDURE "PROCOWNER"."GET_FILTER_FOR_USER"(OUT OUT_FILTER
VARCHAR(5000))
LANGUAGE SQLSCRIPT SQL SECURITY DEFINER READS SQL DATA AS
  v_Filter VARCHAR(5000);
  CURSOR v_Cursor FOR SELECT "FILTER" FROM
"PROCOWNER"."AUTHORIZATION_FILTERS" WHERE "USER_NAME" = SESSION_USER;
BEGIN
  OPEN v_Cursor;
  FETCH v_Cursor INTO v_Filter;
  OUT_FILTER := v_Filter;
  CLOSE v_Cursor;
END;
GRANT EXECUTE ON "PROCOWNER"."GET_FILTER_FOR_USER" TO _SYS_REPO;
```

→ Remember

When using procedures as the condition provider in an SQL-based analytic privilege, remember the following:

- Procedures must have the following properties:
 - They must have the security mode DEFINER.
 - They must be read-only procedures.
 - A procedure with a predefined signature must be used. The following conditions apply:
 - No input parameter
 - Only one output parameter for the filter condition string of string type NVARCHAR, VARCHAR, CLOB, or NCLOB

While VARCHAR and NVARCHAR have length limitations of 5000 characters, CLOB and NCLOB can be used to accommodate longer filter strings.

- The procedure may **not** return a complex filter condition, that is a subquery.
- The procedure must be executable by `_SYS_REPO`, that is, either `_SYS_REPO` must be the owner of the procedure or the owner of the procedure has all privileges on the underlying tables/views with `GRANT OPTION` and has granted the `EXECUTE` privilege on the procedure to the `_SYS_REPO` user.
- The session user is the database user who is executing the query to access a secured view. This is therefore the user whose privileges must be checked. For this reason, the table or view used in the procedure should contain a column to store the user name so that the procedure can filter on this column using the SQL function `SESSION_USER`.
- If errors occur in procedure execution, the user receives a `Not authorized` error, even if he has the analytic privileges that would grant access.

5. Create the analytic privilege using the procedure as condition provider:

```
CREATE STRUCTURED PRIVILEGE AP_SALES_2 FOR SELECT ON
"TABLEOWNER"."VIEW_SALES" CONDITION PROVIDER
"AUTHORIZATION"."GET_FILTER_FOR_USER";
```

On evaluation of the analytic privilege for user ADAMS, the WHERE clause (`CUSTOMER='Company A' AND PRODUCT='Car'`) OR (`YEAR='2006' AND PRODUCT='Bike'`), as provided by the procedure `GET_FILTER_FOR_USER`, will be used.

6. Grant the `SELECT` privilege on the view `TABLEOWNER.VIEW_SALES` to user ADAMS:

```
GRANT SELECT on TABLEOWNER.VIEW_SALES to ADAMS;
```

→ Remember

Only the view owner or a user who has the `SELECT` privilege WITH `GRANT OPTION` on the view can perform the grant.

7. Grant the analytic privilege to user ADAMS:

```
GRANT STRUCTURED PRIVILEGE AP_SALES_2 TO ADAMS;
```

→ Remember

Only the owner of the analytic privilege can grant it.

8.1.3.2 Structure of XML-Based Analytic Privileges

An analytic privilege consists of a set of restrictions against which user access to a particular attribute view, analytic view, or calculation view is verified. In an XML-based analytic privilege, these restrictions are specified in an XML document that conforms to a defined XML schema definition (XSD).

i Note

As objects created in the repository, XML-based analytic privileges are deprecated as of SAP HANA SPS 02. For more information, see SAP Note 2465027.

Each restriction in an XML-based analytic privilege controls the authorization check on the restricted view using a set of value filters. A value filter defines a check condition that verifies whether or not the values of the view (or view columns) qualify for user access.

The following restriction types can be used to restrict data access:

- View
- Activity
- Validity
- Attribute

The following operators can be used to define value filters in the restrictions.

i Note

The activity and validity restrictions support only a subset of these operators.

- IN <list of scalar values>
- CP <pattern with *>
- EQ (=), LE (<=), LT (<), GT (>), GE (>=) <scalar value>
- BT <scalar value as lower limit><scalar value as upper limit>
- IS_NULL
- NOT_NULL

All of the above operators, except IS_NULL and NOT_NULL, accept empty strings (" ") as filter operands. IS_NULL and NOT_NULL do not allow any input value.

The following are examples of how empty strings can be used with the filter operators:

- For the IN operator: IN ("", "A", "B") to filter on these exact values
- As a lower limit in comparison operators, such as:
 - BT ("", "XYZ"), which is equivalent to NOT_NULL AND LE "XYZ""GT "", which is equivalent to NOT_NULL
 - LE "", which is equivalent to EQ ""
 - LT "", which will always return false
 - CP "", which is equivalent to EQ ""

The filter conditions CP "*" will also return rows with empty-string as values in the corresponding attribute.

View Restriction

This restriction specifies to which column views the analytic privilege applies. It can be a single view, a list of views, or all views. An analytic privilege must have exactly one cube restriction.

❖ Example

```
IN ("Cube1", "Cube2")
```

i Note

When an analytic view is created in the SAP HANA modeler, automatically generated views are included automatically in the cube restriction.

i Note

The SAP HANA modeler uses a special syntax to specify the cube names in the view restriction:

```
_SYS_BIC:<package_hierarchy>/<view_name>
```

For example:

```
<cubes>
  <cube name="_SYS_BIC:test.sales/AN_SALES" />
  <cube name="_SYS_BIC:test.sales/AN_SALES/olap" />
</cubes>
```

Activity Restriction

This restriction specifies the activities that the user is allowed to perform on the restricted views, for example, read data. An analytic privilege must have exactly one activity restriction.

❁ Example

```
EQ "read", or EQ "edit"
```

i Note

Currently, all analytic privileges created in the SAP HANA modeler are automatically configured to restrict access to READ activity only. This corresponds to SQL SELECT queries. This is due to the fact that the attribute, analytic, and calculation views are read-only views. This restriction is therefore not configurable.

Validity Restriction

This restriction specifies the validity period of the analytic privilege. An analytic privilege must have exactly one validity restriction.

❁ Example

```
GT 2010/10/01 01:01:00.000
```

Attribute Restriction

This restriction specifies the value range that the user is permitted to access. Attribute restrictions are applied to the actual attributes of a view. Each attribute restriction is relevant for one attribute, which can contain multiple value filters. Each value filter represents a logical filter condition.

i Note

The SAP HANA modeler uses different ways to specify attribute names in the attribute restriction depending on the type of view providing the attribute. In particular, attributes from attribute views are

specified using the syntax "<package_hierarchy>/<view_name>\$<attribute_name>", while local attributes of analytic views and calculation views are specified using their attribute name only. For example:

```
<dimensionAttribute name="test.sales/AT_PRODUCT$PRODUCT_NAME">
  <restrictions>
    <valueFilter operator="IN">
      <value value="Car" />
      <value value="Bike" />
    </valueFilter>
  </restrictions>
</dimensionAttribute>
```

Value filters for attribute restrictions can be static or dynamic.

- A **static** value filter consists of an operator and either a list of values as the filter operands or a single value as the filter operand. All data types are supported except those for LOB data types (CLOB, BLOB, and NCLOB).

For example, a value filter (EQ 2006) can be defined for an attribute YEAR in a dimension restriction to filter accessible data using the condition YEAR=2006 for potential users.

i Note

Only attributes, not aggregatable facts (for example, measures or key figures) can be used in dimension restrictions for analytic views.

- A **dynamic** value filter consists of an operator and a stored procedure call that determines the operand value at runtime.

For example, a value filter (IN (GET_MATERIAL_NUMBER_FOR_CURRENT_USER())) is defined for the attribute MATERIAL_NUMBER. This filter indicates that a user with this analytic privilege is only allowed to access material data with the numbers returned by the procedure GET_MATERIAL_NUMBER_FOR_CURRENT_USER.

It is possible to combine static and dynamic value filters as shown in the following example.

❁ Example

```
<dimensionAttribute name=" test.sales/AT_PRODUCT$PRODUCT_NAME ">
  <restrictions>
    <valueFilter operator="CP"> <value value="ELECTRO*" /> </
valueFilter>
    <valueFilter operator="IN"> <procedureCall
schema="PROCEDURE_OWNER" procedure="DETERMINE_AUTHORIZED_PRODUCT_FOR_USER" />
</valueFilter >
  </restrictions>
</dimensionAttribute>
<dimensionAttribute name=" test.sales/AT_TIME$YEAR ">
  <restrictions>
    <valueFilter operator="EQ"> <value value="2012" /> </valueFilter>
    <valueFilter operator="IN"> <procedureCall
schema="PROCEDURE_OWNER" procedure="DETERMINE_AUTHORIZED_YEAR_FOR_USER" /> </
valueFilter >
  </restrictions>
```

An analytic privilege can have multiple attribute restrictions, but it must have at least one attribute restriction. An attribute restriction must have at least one value filter. Therefore, if you want to permit access to the whole content of a restricted view, then the attribute restriction must specify all attributes.

Similarly, if you want to permit access to the whole content of the view with the corresponding attribute, then the value filter must specify all values.

The SAP HANA modeler automatically implements these two cases if you do not select either an attribute restriction or a value filter.

❁ Example

Specification of all attributes:

```
<dimensionAttributes>
  <allDimensionAttributes/ >
</dimensionAttributes>
```

❁ Example

Specification of all values of an attribute:

```
<dimensionAttributes>
  <dimensionAttribute name="PRODUCT">
    <all />
  </dimensionAttribute>
</dimensionAttributes>
```

Logical Combination of Restrictions and Filter Conditions

The result of user queries on restricted views is filtered according to the conditions specified by the analytic privileges granted to the user as follows:

- Multiple analytic privileges are combined with the logical operator OR.
- Within one analytic privilege, all attribute restrictions are combined with the logical operator AND.
- Within one attribute restriction, all value filters on the attribute are combined with the logical operator OR.

❁ Example

You create two analytic privileges AP1 and AP2. AP1 has the following attribute restrictions:

- Restriction R11 restricting the attribute Year with the value filters (EQ 2006) and (BT 2008, 2010)
- Restriction R12 restricting the attribute Country with the value filter (IN ("USA", "Germany"))

Given that multiple value filters are combined with the logical operator OR and multiple attribute restrictions are combined with the logical operator AND, AP1 generates the condition:

```
((Year = 2006) OR (Year BT 2008 and 2010)) AND (Country IN ("USA", "Germany"))
```

AP2 has the following restriction:

Restriction R21 restricting the attribute Country with the value filter (EQ "France")

AP2 generates the condition:

```
(Country = "France")
```

Any query of a user who has been granted both AP1 and AP2 will therefore be appended with the following WHERE clause:

```
((Year = 2006) OR (Year BT 2008 and 2010)) AND (Country IN ("USA", "Germany"))  
OR (Country = "France")
```

Related Information

[SAP Note 2465027](#)

8.1.3.2.1 Dynamic Value Filters in the Attribute Restriction of XML-Based Analytic Privileges

The attribute restriction of an XML-based analytic privilege specifies the value range that the user is permitted to access using value filters. In addition to static scalar values, stored procedures can be used to define filters.

By using stored procedures to define filters, you can have user-specific filter conditions be determined dynamically in runtime, for example, by querying specified tables or views. As a result, the same analytic privilege can be applied to many users, while the filter values for authorization can be updated and changed independently in the relevant database tables. In addition, application developers have full control not only to design and manage such filter conditions, but also to design the logic for obtaining the relevant filter values for the individual user at runtime.

Procedures used to define filter conditions must have the following properties:

- They must have the security mode DEFINER.
- They must be read-only procedures.
- A procedure with a predefined signature must be used. The following conditions apply:
 - No input parameter
 - Only 1 output parameter as table type with one single column for the IN operator
 - Only 1 output parameter of a scalar type for all unary operators, such as EQUAL
 - Only 2 output parameters of a scalar type for the binary operator BETWEEN
- Only the following data types are supported as the scalar types and the data type of the column in the table type:
 - Date/Time types DATE, TIME, SECONDDATE, and TIMESTAMP
 - Numeric types TINYINT, SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, and DOUBLE
 - Character string types VARCHAR and NVARCHAR
 - Binary type VARBINARY

NULL as Operand for Filter Operators

In static value filters, it is not possible to specify NULL as the operand of the operator. The operators IS_NULL or NOT_NULL must be used instead. In dynamic value filters where a procedure is used to determine a filter

condition, NULL or valid values may be returned. The following behavior applies in the evaluation of such cases during the authorization check of a user query:

Filter conditions of operators with NULL as the operand are disregarded, in particular the following:

- EQ NULL, GT NULL, LT NULL, LE NULL, and CP NULL
- BT NULL and NULL

If no valid filter conditions remain (that is, they have all been disregarded because they contain the NULL operand), the user query is rejected with a “Not authorized” error.

❖ Example

Dynamic analytic privilege 1 generates the filter condition (Year >= NULL) and dynamic analytic privilege 2 generates the condition (Country EQ NULL). The query of a user assigned these analytic privileges (combined with the logical operator OR) will return a “Not authorized” error.

❖ Example

Dynamic analytic privilege 1 generates the filter condition (Year >= NULL) and dynamic analytic privilege 2 generates the condition (Country EQ NULL AND Currency = “USD”). The query of a user assigned these analytic privileges (combined with the logical operator OR) will be filtered with the filter Currency = ‘USD’.

In addition, a user query is not authorized in the following cases even if further applicable analytic privileges have been granted to the user.

- The BT operator has as input operands a valid scalar value and NULL, for example, BT 2002 and NULL or BT NULL and 2002
- The IN operator has as input operand NULL among the value list, for example, IN (12, 13, NULL)

Permitting Access to All Values

If you want to allow the user to see all the values of a particular attribute, instead of filtering for certain values, the procedure must return "" and "" (empty string) as the operand for the CP and GT operators respectively. These are the only operators that support the specification of all values.

Implementation Considerations

When the procedure is executed as part of the authorization check in runtime, note the following:

- The user who must be authorized is the database user who executes the query accessing a secured view. This is the session user. The database table or view used in the procedure must therefore contain a column to store the user name of the session user. The procedure can then filter by this column using the SQL function SESSION_USER. This table or view should only be accessible to the procedure owner.

⚠ Caution

Do not map the executing user to the application user. The application user is unreliable because it is controlled by the client application. For example, it may set the application user to a technical user or it may not set it at all. In addition, the trustworthiness of the client application cannot be guaranteed.

- The user executing the procedure is the `_SYS_REPO` user. In the case of procedures activated in the SAP HANA modeler, `_SYS_REPO` is the owner of the procedures. For procedures created in SQL, the `EXECUTE` privilege on the procedure must be granted to the `_SYS_REPO` user.
- If the procedure fails to execute, the user's query stops processing and a "Not authorized" error is returned. The root cause can be investigated in the error trace file of the indexserver, `indexserver_alert_<host>.trc`.

When designing and implementing procedures as filter for dynamic analytic privileges, bear the following in mind:

- To avoid a recursive analytic privilege check, the procedures should only select from database tables or views that are not subject to an authorization check based on analytic privileges. In particular, views activated in the SAP HANA modeler are to be avoided completely as they are automatically registered for the analytic privilege check.
- The execution of procedures in analytic privileges slows down query processing compared to analytic privileges containing only static filters. Therefore, procedures used in analytic privileges must be designed carefully.

8.1.3.3 Runtime Authorization Check of Analytic Privileges

When a user requests access to data stored in an attribute, analytic, calculation, or SQL views, an authorization check based on analytic privileges is performed and the data returned to the user is filtered accordingly. The `EFFECTIVE_STRUCTURED_PRIVILEGES` system view can help you to troubleshoot authorization problems.

Access to a view and the way in which results are filtered depend on whether the view is independent or associated with other views (dependent views).

Independent Views

The authorization check for a view that is not defined on another column view is as follows:

1. The user's authorization to access the view is checked.
A user can access the view if **both** of the following prerequisites are met:
 - The user has been granted the `SELECT` privilege on the view or the schema in which it is located.

i Note

The user does not require `SELECT` on the underlying base tables or views of the view.

- The user has been granted an analytic privilege that is applicable to the view.
Applicable analytic privileges are those that meet **all** of the following criteria:

XML-Based Analytic Privilege

A view restriction that includes the accessed view

SQL-Based Analytic Privilege

An `ON` clause that includes the accessed view

| XML-Based Analytic Privilege | SQL-Based Analytic Privilege |
|---|--|
| A validity restriction that applies now | If the filter condition specifies a validity period (for example, <code>WHERE (CURRENT_TIME BETWEEN ... AND ...)</code> AND <code><actual filter></code>), it must apply now |
| <p>An action in the activity restriction that covers the action requested by the query</p> <div data-bbox="288 551 837 748"> <p>i Note</p> <p>All analytic privileges created and activated in the SAP HANA modeler and SAP HANA Web-based Development Workbench fulfill this condition. The only action supported is read access (SELECT).</p> </div> | <p>An action in the FOR clause that covers the action requested by the query</p> <div data-bbox="847 551 1402 748"> <p>i Note</p> <p>All analytic privileges created and activated in the SAP HANA Web-based Development Workbench fulfill this condition. The only action supported is read access (SELECT).</p> </div> |
| An attribute restriction that includes some of the view's attributes | <p>A filter condition that applies to the view</p> <div data-bbox="847 819 1402 1111"> <p>i Note</p> <p>When the analytic privilege is created, the filter is checked immediately to ensure that it applies to the view. If it doesn't, creation will fail. However, if the view definition subsequently changes, or if a dynamically generated filter condition returns a filter string that is not executable with the view, the authorization check will fail and access is rejected.</p> </div> |

If the user has the SELECT privilege on the view but no applicable analytic privileges, the user's request is rejected with a `Not authorized` error. The same is true if the user has an applicable analytic privilege but doesn't have the SELECT privilege on the view.

- The value filters specified in the dimension restrictions (XML-based) or filter condition (SQL-based) are evaluated and the appropriate data is returned to the user. Multiple analytic privileges are combined with the logical operator OR.
For more information about how multiple attribute restrictions and/or multiple value filters in XML-based analytic privileges are combined, see *XML-Based Analytic Privileges*.

Dependent Views

The authorization check for a view that is defined on other column views is more complex. Note the following behavior.

Calculation and SQL views

- Individual views in the hierarchy are filtered according to their respective analytic privileges, which use the logical OR combination.
- The filtered result of the calculation view is derived from the filtered result of its underlying views. This corresponds to a logic AND combination of the filters generated by the analytic privileges for the individual views.

Result filtering on the view is then performed as follows:

- The user has been granted the SELECT privilege on the view or the schema that contains the view.
- The user has been granted analytic privileges that apply to the view itself **and** all the other column views in the hierarchy that are registered for a structured privilege check.

A user can access a calculation or SQL view based on other views if **both** of the following prerequisites are met:

If a user requests access to a calculation view that is dependent on another view, the behavior of the authorization check and result filtering is performed as follows:

Calculation views and SQL views can be defined by selecting data from other column views, specifically attribute views, analytic views, and other calculation views. This can lead to a complex view hierarchy that requires careful design of row-level authorization.

Analytic views

An analytic view can also be defined on attribute views, but this does **not** represent a view dependency or hierarchy with respect to authorization check and result filtering. If you reference an attribute view in an analytic view, analytic privileges defined on the attribute view are not applied.

This represents a view hierarchy for which the prerequisites described above for calculation views also apply.

- Currency or unit conversions
- Calculated attributes
- Calculated measures that use attributes, calculated attributes, or input parameters in their formulas

If an analytic view designed in the SAP HANA modeler contains one of the elements listed below, it will automatically be activated with a calculation view on top. The name of this calculation view is the name of the analytic view with the suffix /o1ap.

Troubleshooting Failed Authorization

Using the EFFECTIVE_STRUCTURED_PRIVILEGES system view, you can quickly see:

- Which analytic privileges apply to a particular view, including the dynamic filter conditions that apply (if relevant)
- Which filter is being applied to which view in the view hierarchy (for views with dependencies)
- Whether or not a particular user is authorized to access the view

Query EFFECTIVE_STRUCTURED_PRIVILEGES as follows:

```
SELECT * from "PUBLIC"."EFFECTIVE_STRUCTURED_PRIVILEGES" where ROOT_SCHEMA_NAME = '<schema>' AND ROOT_OBJECT_NAME = '<OBJECT>' AND USER_NAME = '<USER>'
```

Related Information

[Structure of XML-Based Analytic Privileges \[page 145\]](#)

8.1.4 Package Privileges

Package privileges authorize actions on individual packages in the classic SAP HANA repository.

i Note

With SAP HANA XS advanced, source code and web content are not versioned and stored in the SAP HANA database, so package privileges are not used in this context.

Privileges granted on a repository package are implicitly assigned to the design-time objects in the package, as well as to all sub-packages. Users are only allowed to maintain objects in a repository package if they have the necessary privileges for the package in which they want to perform an operation, for example to read or write to an object in that package. To be able perform operations in all packages, a user must have privileges on the root package `.REPO_PACKAGE_ROOT`.

→ Recommendation

We recommend that package privileges be granted on a single package or a small number of specific packages belonging to your organization, rather than on the complete repository.

If the user authorization check establishes that a user does not have the necessary privileges to perform the requested operation in a specific package, the authorization check is repeated on the parent package and recursively up the package hierarchy to the root level of the repository. If the user does not have the necessary privileges for any of the packages in the hierarchy chain, the authorization check fails and the user is not permitted to perform the requested operation.

In the context of repository package authorizations, there is a distinction between native packages and imported packages.

Privileges for Native Repository Packages

A native repository package is created in the current SAP HANA system and expected to be edited in the current system. To perform application-development tasks on **native** packages in the SAP HANA repository, developers typically need the privileges listed in the following table:

| Package Privilege | Description |
|--|--|
| <code>REPO.READ</code> | Read access to the selected package and design-time objects (both native and imported) |
| <code>REPO.EDIT_NATIVE_OBJECTS</code> | Authorization to modify design-time objects in packages originating in the system the user is working in |
| <code>REPO.ACTIVATE_NATIVE_OBJECTS</code> | Authorization to activate/reactivate design-time objects in packages originating in the system the user is working in |
| <code>REPO.MAINTAIN_NATIVE_PACKAGES</code> | Authorization to update or delete native packages, or create sub-packages of packages originating in the system in which the user is working |

Privileges for Imported Repository Packages

An imported repository package is created in a remote SAP HANA system and imported into the current system. To perform application-development tasks on **imported** packages in the SAP HANA repository, developers need the privileges listed in the following table:

i Note

It is not recommended to work on imported packages. Imported packages should only be modified in exceptional cases, for example, to carry out emergency repairs.

| Package Privilege | Description |
|---------------------------------|--|
| REPO.READ | Read access to the selected package and design-time objects (both native and imported) |
| REPO.EDIT_IMPORTED_OBJECTS | Authorization to modify design-time objects in packages originating in a system other than the one in which the user is currently working |
| REPO.ACTIVATE_IMPORTED_OBJECTS | Authorization to activate (or reactivate) design-time objects in packages originating in a system other than the one in which the user is currently working |
| REPO.MAINTAIN_IMPORTED_PACKAGES | Authorization to update or delete packages, or create sub-packages of packages, which originated in a system other than the one in which the user is currently working |

8.1.5 Application Privileges

In SAP HANA XS classic, application privileges define the authorization level required for access to an SAP HANA XS classic application, for example, to start the application or view particular functions and screens.

i Note

With SAP HANA XS advanced, application privileges are not used. Application-level authorization is implemented using OAuth and authorization scopes and attributes.

Application privileges can be assigned to an individual user or to a group of users, for example, in a role. The role can also be used to assign system, object, package, and analytic privileges. You can use application privileges to provide different levels of access to the same application, for example, to provide advanced maintenance functions for administrators and view-only capabilities to normal users.

If you want to define application-specific privileges, you need to understand and maintain the relevant sections in the following design-time artifacts:

- Application-privileges file (`.xsprivileges`)
- Application-access file (`.xsaccess`)
- Role-definition file (`<RoleName>.hdbrole`)

Application privileges can be assigned to users individually or by means of a user **role**, for example, with the “*application privilege*” keyword in a role-definition file (`<RoleName>.hdbrole`) as illustrated in the following

code. You store the roles as design-time artifacts within the application package structure they are intended for, for example, `acme.com.hana.xs.appl.roles`.

```
role acme.com.hana.xs.appl.roles::Display
{
  application privilege: acme.com.hana.xs.appl::Display;
  application privilege: acme.com.hana.xs.appl::View;
  catalog schema "ACME_XS_APP1": SELECT;
  package acme.com.hana.xs.appl: REPO.READ;
  package ".REPO_PACKAGE_ROOT" : REPO.READ;
  catalog sql object "_SYS_REPO"."PRODUCTS": SELECT;
  catalog sql object "_SYS_REPO"."PRODUCT_INSTANCES": SELECT;
  catalog sql object "_SYS_REPO"."DELIVERY_UNITS": SELECT;
  catalog sql object "_SYS_REPO"."PACKAGE_CATALOG": SELECT;
  catalog sql object "ACME_XS_APPL"."acme.com.hana.xs.appl.db::SYSTEM_STATE" :
  SELECT, INSERT, UPDATE, DELETE;
}
```

The application privileges referenced in the role definition (for example, `Display` and `View`) are actually defined in an application-specific `.xsprivileges` file, as illustrated in the following example, which also contains entries for additional privileges that are not explained here.

Note

The `.xsprivileges` file must reside in the package of the application to which the privileges apply.

The package where the `.xsprivileges` resides defines the scope of the application privileges; the privileges specified in the `.xsprivileges` file can only be used in the package where the `.xsprivileges` resides (or any sub-packages). This is checked during activation of the `.xsaccess` file and at runtime in the by the XS JavaScript API `$.session.(has|assert)AppPrivilege()`.

```
{
  "privileges" : [
    { "name" : "View", "description" : "View Product Details" },
    { "name" : "Configure", "description" : "Configure Product Details" },
    { "name" : "Display", "description" : "View Transport Details" },
    { "name" : "Administrator", "description" : "Configure/Run Everything" },
    { "name" : "ExecuteTransport", "description" : "Run Transports" },
    { "name" : "Transport", "description" : "Transports" }
  ]
}
```

The privileges are **authorized** for use with an application by inserting the `authorization` keyword into the corresponding `.xsaccess` file, as illustrated in the following example. Like the `.xsprivileges` file, the `.xsaccess` file must reside either in the root package of the application to which the privilege authorizations apply or the specific subpackage which requires the specified authorizations.

Note

If a privilege is inserted into the `.xsaccess` file as an authorization requirement, a user must have this privilege to access the application package where the `.xsaccess` file resides. If there is more than one privilege, the user must have at least one of these privileges to access the content of the package.

```
{
  "prevent_xsrp": true,
  "exposed": true,
  "authentication": {
    "method": "Form"
  },
}
```

```
"authorization": [
  "acme.com.hana.xs.appl::Display",
  "acme.com.hana.xs.appl::Transport"
]
```

8.2 Database Roles

A database role is a collection of privileges that can be granted to either a database user or another role in runtime.

A role typically contains the privileges required for a particular function or task, for example:

- Business end users reading reports using client tools such as Microsoft Excel
- Modelers creating models and reports
- Database administrators operating and maintaining the database and its users

Privileges can be granted directly to users of the SAP HANA database. However, roles are the standard mechanism of granting privileges as they allow you to implement complex, reusable authorization concepts that can be modeled on business roles.

Creation of Roles

Roles in the SAP HANA database can exist as runtime objects only (catalog roles), or as design-time objects that become catalog objects on deployment (database artifact with file suffix `.hdbrrole`).

In an SAP HANA XS classic environment, database roles are created in the built-in repository of the SAP HANA database using either the SAP HANA Web Workbench or the SAP HANA studio. These are also referred to as repository roles. In an SAP HANA XS advanced environment, design-time roles are created using the SAP Web IDE and deployed using SAP HANA deployment infrastructure (SAP HANA DI, or HDI).

i Note

Due to the container-based model of HDI where each container corresponds to a database schema, HDI roles, once deployed, are schema specific.

SAP HANA XS advanced has the additional concept of application roles and role collections. These are independent of database roles in SAP HANA itself. In the XS advanced context, SAP HANA database roles are used only to control access to database objects (for example, tables, views, and procedures) for XS advanced applications. For more information about the authorization concept of XS advanced, see the *SAP HANA Security Guide*.

Role Structure

A role can contain any number of the following privileges:

- **System privileges** for general system authorization, in particular administration activities
- **Object privileges** (for example, SELECT, INSERT, UPDATE) on database objects (for example, schemas, tables, views, procedures, and sequences)
- **Analytic privileges** on SAP HANA information models
- **Package privileges** on repository packages (for example, REPO.READ, REPO.EDIT_NATIVE_OBJECTS, REPO.ACTIVATE_NATIVE_OBJECTS)
- **Application privileges** for enabling access to SAP HANA-based applications developed in an SAP HANA XS classic environment

i Note

There are no HDI or XS advanced equivalents in the SAP HANA authorization concept for package privileges on repository packages and applications privileges on SAP HANA XS classic applications. For more information about the authorization concept of XS advanced, see the *SAP HANA Security Guide*.

A role can also contain other roles.

Roles Best Practices

For best performance of role operations, in particular, granting and revoking, keep the following basic rules in mind:

- Create roles with the smallest possible set of privileges for the smallest possible group of users who can share a role (principle of least privilege).
- Avoid granting object privileges at the schema level to a role if only a few objects in the schema are relevant for intended users.
- Avoid creating and maintaining all roles as a single user. Use several role administrator users instead.

Related Information

[Authorization in SAP HANA XS Advanced \[page 313\]](#)

8.2.1 Predefined Database (Catalog) Roles

Several catalog roles are available by default in the SAP HANA database.

Several predefined catalog roles are delivered with the SAP HANA database. You should not use these roles directly, but instead use them as templates for creating your own roles.

The table below lists the catalog roles delivered with the SAP HANA database.

Note

These roles do not exist in the SAP HANA repository.

| Role | Description |
|------------------------------------|---|
| CONTENT_ADMIN | <p>This role contains all the privileges required for using the information modeler in the SAP HANA studio, as well the additional authorization to grant these privileges to other users. It also contains system privileges for working with imported objects in the SAP HANA repository. You can use this role as a template for creating roles for content administrators.</p> <p>Caution</p> <p>The <code>CONTENT_ADMIN</code> role is very privileged and should not be granted to users, particularly in production systems. The <code>CONTENT_ADMIN</code> role should only be used as a template.</p> |
| MODELING | <p>This role contains all the privileges required for using the information modeler in the SAP HANA studio.</p> <p>It therefore provides a modeler with the database authorization required to create all kinds of views and analytic privileges.</p> <p>Caution</p> <p>The <code>MODELING</code> role contains the predefined analytic privilege <code>_SYS_BI_CP_ALL</code>. This analytic privilege potentially allows a user to access all the data in activated views that are protected by XML-based analytic privileges, regardless of any other analytic privileges that apply. Although the user must also have the <code>SELECT</code> object privilege on the views to actually be able to access data, the <code>_SYS_BI_CP_ALL</code> analytic privilege should not be granted to users, particularly in production systems. For this reason, the <code>MODELING</code> role should only be used as a template.</p> |
| MONITORING | <p>This role contains privileges for full read-only access to all metadata, the current system status in system and monitoring views, and the data collected by the statistics server.</p> |
| PUBLIC | <p>This role contains privileges for filtered read-only access to the system views. Only objects for which the users have access rights are visible. By default, this role is granted to every user, except restricted users.</p> |
| BI_META_DATA_CONSUMER (_SYS_BI) | <p>This role gives read access to the SAP HANA analytic catalog (BIMC views) and nothing else. As analytic clients need to query metadata from these tables at runtime, the role can be used to give end users of analytic clients the required privileges. In contrast to the role <code>MODELING</code> or <code>CONTENT_ADMIN</code>, this role doesn't give full access to the schemas <code>_SYS_BI</code> or <code>_SYS_BIC</code>. Therefore, at least <code>SELECT</code> privileges to the relevant views must also be granted.</p> |

| Role | Description |
|---|---|
| BI_META_DATA_CONSUMER_HDI (_SYS_BI) | <p>This role is similar to BI_META_DATA_CONSUMER but only for _HDI BIMC views. These are a bit faster but only read translated texts for SAP HANA deployment infrastructure (HDI) based objects. They don't include translated texts for repository objects.</p> |
| <p>i Note</p> <p>As clients will still query the original BIMC views, this role cannot be used for end users consuming analytic clients.</p> | |
| CREATE_INTERMEDIATE_CALCULATION_VIEW (_SYS_BI) | <p>This role contains only the EXECUTE privilege for a procedure that is called to create temporary calculation views used for data preview on inner nodes. Modelers who want to invoke the data preview on an inner calculation view node from the calculation view editor require this EXECUTE privilege. This privilege is also contained in role MODELING. However, since MODELING is a very powerful role that grants users too many privileges, it should not be granted to end users.</p> <p>See also SAP Note 2299622.</p> |
| RESTRICTED_USER_JDBC_ACCESS | <p>This role contains the privileges required by restricted database users to connect to SAP HANA through the JDBC client interface.</p> <p>This role is intended to be used in conjunction with application-specific roles. It is recommended that the privileges required to use an application are encapsulated within an application-specific role, which is then granted to restricted database users. By including the RESTRICTED_USER_JDBC_ACCESS role in the application-specific role, it can be ensured that application users have only those privileges that are essential to their work.</p> |
| RESTRICTED_USER_ODBC_ACCESS | <p>This role contains the privileges required by restricted database users to connect to SAP HANA through the ODBC client interface.</p> <p>This role is intended to be used in conjunction with application-specific roles. It is recommended that the privileges required to use an application are encapsulated within an application-specific role, which is then granted to restricted database users. By including the RESTRICTED_USER_ODBC_ACCESS role in the application-specific role, it can be ensured that application users have only those privileges that are essential to their work.</p> |

| Role | Description |
|--|--|
| SAP_INTERNAL_HANA_SUPPORT | <p>This role contains system privileges (for example, CATALOG READ) and object privileges (for example, SELECT on SYS schema) that allow access to certain low-level internal system views needed by SAP HANA development support in support situations. All access is read only. This role does not allow access to any customer data.</p> <p>The definition of the low-level internal system views to which this role allows access is not part of the stable end-user interface and might change from revision to revision. To avoid administrators and end users accidentally accessing these internal system views in applications or scripts, this role is therefore subject to several usage restrictions (listed below) and should be granted only to SAP HANA development support users for their support activities.</p> <p>In detail, this role contains privileges for read-only access to all metadata, the current system status, and the data of the statistics server. Additionally, it contains the privileges for accessing low-level internal system views. Without the SAP_INTERNAL_HANA_SUPPORT role, this information can be selected only by the SYSTEM user.</p> <p>To avoid accidental use of this role in day-to-day activities, the following restrictions apply to the SAP_INTERNAL_HANA_SUPPORT role:</p> <ul style="list-style-type: none"> • It cannot be granted to the SYSTEM user. • It can only be granted to a limited number of users at the same time. The maximum number of users to which the role can be granted can be configured with the parameter <code>internal_support_user_limit</code> in the authorization section of the <code>indexserver.ini</code> configuration file. The default value is 1. • It cannot be granted to another role. • It cannot be granted further object privileges. • It can be granted only further system privileges. • With every upgrade of the SAP HANA database, it is reset to its default privileges. <p>To ensure that system administrators are aware that the SAP_INTERNAL_HANA_SUPPORT role is currently granted to one or more users in a system, an information alert is issued every hour by default. This behavior can be configured with check 63.</p> |
| <ul style="list-style-type: none"> • AFL__SYS_AFL_AFLPAL_EXECUT E • AFL__SYS_AFL_AFLPAL_EXECUT E_WITH_GRANT_OPTION • AFL__SYS_AFL_AFLBFL_EXECUT E • AFL__SYS_AFL_AFLBFL_EXECUT E_WITH_GRANT_OPTION | <p>Predefined roles for application function libraries (AFL): Predictive Analysis Library (PAL) and Business Function Library (BFL)</p> <p>For more information, see the SAP HANA Predictive Analysis Library (PAL) Reference and the SAP HANA Business Function Library (BFL) Reference</p> |

| Role | Description |
|---|--|
| _SYS_DI_OO_DEFAULTS <div style="background-color: #f0f0f0; padding: 5px;"> <p>i Note</p> <p>Only exists if HDI is enabled in the database.</p> </div> | <p>This role contains the set of default privileges that are granted to all HDI container object owner users (<container>#OO users). SAP HANA Deployment Infrastructure (HDI) uses this role internally to grant default privileges instead of using the PUBLIC role. It contains only privileges to SYS views where additional security checks apply.</p> <p>The role contains SELECT privileges on the views: SYS . DUMMY, SYS . PROCEDURES, SYS . PROCEDURE_PARAMETERS, SYS . TABLES, SYS . TABLE_COLUMNS.</p> <p>This role is not intended to be granted to database users.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>i Note</p> <p>Do not extend this role in a production system.</p> </div> |

Related Information

[Predefined Repository Roles \[page 173\]](#)

[Predefined Database Roles for XS Advanced \[page 310\]](#)

[SAP Note 2299622](#) 

8.2.2 Catalog Roles and Design-Time Roles Compared

It is possible to create roles as pure runtime objects that follow classic SQL principles or as design-time objects in either an SAP HANA XS advanced or classic environment.

In SAP HANA XS classic, database roles are created in the built-in repository of the SAP HANA database using either the SAP HANA Web Workbench or the SAP HANA studio. In SAP HANA XS advanced, design-time roles are created using the SAP Web IDE and deployed using SAP HANA deployment infrastructure (SAP HANA DI, or HDI).

i Note

SAP HANA XS classic and the SAP HANA repository are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.

The following table summarizes the differences between catalog roles and design-time roles:

| Feature | Catalog Roles | Design-Time Repository Roles (XSC) | Design-Time HDI Roles (XSA) |
|--------------------|--|---|---|
| Transportability | Roles cannot be transported between systems. They can only be created in runtime by users with the system privilege ROLE ADMIN. | Roles can be transported between systems using several transport options: <ul style="list-style-type: none"> • SAP HANA Application Lifecycle Manager • The change and transport system (CTS+) of the SAP NetWeaver ABAP application server • SAP HANA Transport Container (HTC) | |
| Version management | No version management is possible. | The repository provides the basis for versioning. As repository objects, roles are stored in specific repository tables inside the database. This eliminates the need for an external version control system. | Roles are developed as design-time objects within a project stored in the GIT repository. The complete role history is therefore available in GIT. |
| Ownership | Roles are owned by the database user who creates them. If the creating user is dropped, any roles created in the user's own schema are also dropped. | The technical user <code>_SYS_REPO</code> is the owner of all roles created in the repository, not the database user who creates them. Therefore, roles are not directly associated with the creating user. To create a role, a database user needs only the privileges required to work in the repository. | Roles are owned by the object owner of the container (technical user <code><container>#00</code>) where role development is taking place. Roles are not directly associated with the creating user. To create a role, a developer needs only the authorization required to work in the relevant space using the SAP Web IDE for SAP HANA. If roles for different purposes are developed in different containers, then roles are not all owned by the same technical user (as is the case with repository roles). |

| Feature | Catalog Roles | Design-Time Repository Roles (XSC) | Design-Time HDI Roles (XSA) |
|--------------------------|---|---|--|
| Grant and revoke process | <p>Roles created in runtime are granted directly by the database user using the SQL GRANT and REVOKE statements.</p> <p>To grant privileges to a role, a user requires either the system privilege ROLE ADMIN, or all the privileges being granted to the role. In the latter case, if any of these privileges are revoked from the granting user, they are automatically revoked from the role.</p> <p>Roles can be revoked by the granting user or any user with the system privilege ROLE ADMIN.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>With the exception of roles granted by technical user <code>_SYS_REPO</code>, a user with ROLE ADMIN cannot revoke roles granted by technical users <code>SYS</code> and <code>_SYS*</code>.</p> </div> <p>If the granting user is dropped (not necessarily the role creator), all roles that he or she granted are not revoked.</p> | <p>Roles are granted and revoked using built-in procedures. Any administrator with the EXECUTE privilege on these can grant and revoke roles.</p> | <p>Roles are granted and revoked using database procedures provided in either the HDI container's or container group's API schema. Any container or container group administrator with the EXECUTE privilege on these procedures can grant and revoke roles.</p> <p>Any user with the system privilege ROLE ADMIN can also grant and revoke roles.</p> |

In general, it is recommended that you model roles as design-time objects for the following reasons:

- Unlike roles created in runtime, roles created as design-time objects can be transported between systems. This is important for application development as it means that developers can model roles as part of their application's security concept and then ship these roles or role templates with the application. Being able to transport roles is also advantageous for modelers implementing complex access control on analytic content. They can model roles in a test system and then transport them into a production system. This avoids unnecessary duplication of effort.
- Roles created as design-time objects are not directly associated with a database user. They are created by technical users and granted through the execution of stored procedures. Any user with access to these procedures can grant and revoke a role. Roles created in runtime are granted directly by the database user and can be revoked only by the granting user or a user with the system privilege ROLE ADMIN. Additionally, if the database user is deleted, all roles that he or she granted are revoked. As database users correspond

to real people, this could impact the implementation of your authorization concept, for example, if an employee leaves the organization or is on vacation.

Catalog roles make sense in scenarios where user and role provisioning is carried out solely using a higher-level application that connects to SAP HANA through a technical user such as SAP Identity Management.

Related Information

[SAP HANA DI Roles \[page 166\]](#)

[Repository Roles \[page 169\]](#)

[SAP Note 2465027](#) 

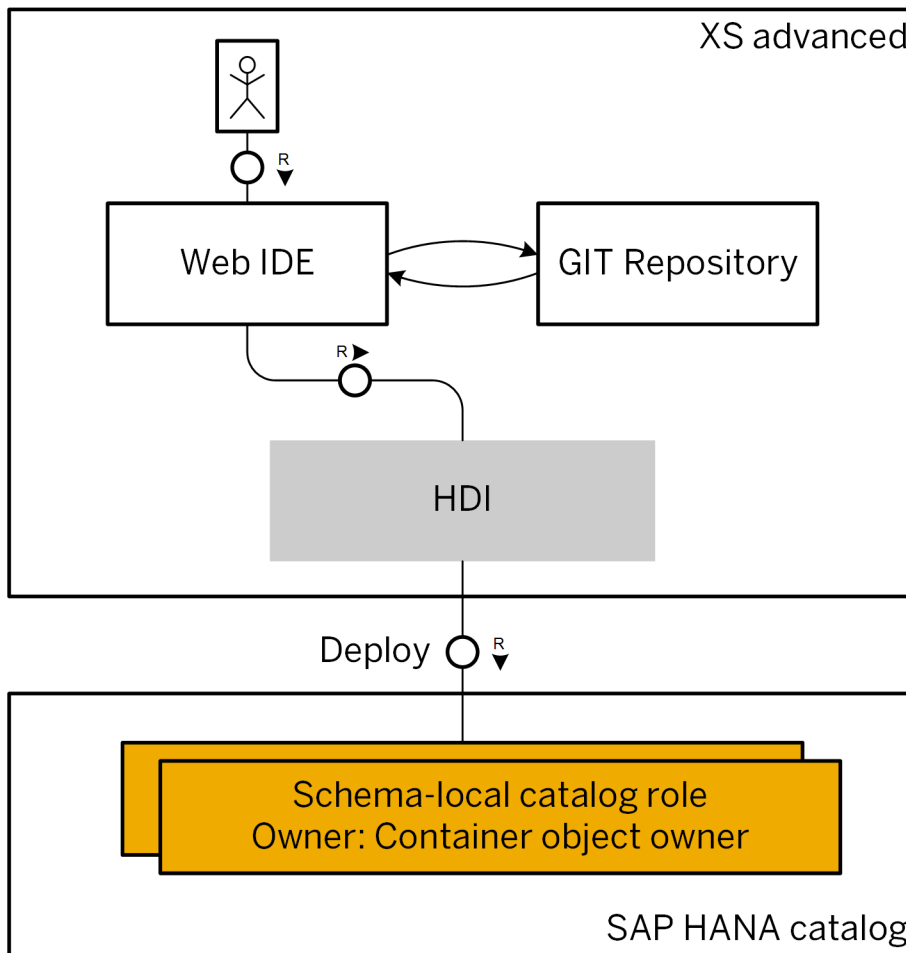
8.2.3 SAP HANA DI Roles

In an SAP HANA XS advanced environment, role developers use the SAP Web IDE for SAP HANA to create the design-time version of roles within a project stored in the GIT repository. When the developer deploys the project, a runtime version of the roles is created in an HDI container (schema) within the SAP HANA database.

- [Overview \[page 166\]](#)
- [What does the Controller-based model of HDI mean for role development? \[page 167\]](#)
- [Authorizations Needed for Role Development \[page 168\]](#)
- [Granting and Revoking HDI Roles \[page 168\]](#)
- [Dropping HDI Roles \[page 169\]](#)
- [Auditing HDI Roles \[page 169\]](#)

Overview

SAP HANA XS advanced provides a comprehensive platform for the development and execution of native data-intensive applications that run efficiently in SAP HANA. It succeeds SAP HANA XS classic as the default application programming model for SAP HANA. This changes the way in which database objects, including roles, are developed. Design-time objects are created within a project stored in the GIT repository. When the developer deploys the project, a runtime version of the objects is created in an HDI container in the SAP HANA database.



Role Development in the SAP HANA Deployment Infrastructure (HDI)

What does the Controller-based model of HDI mean for role development?

In the SAP HANA XS advanced model, development resources may be isolated by leveraging the concept of organizations and spaces. This makes it possible to keep security-related development, like the development of roles, separate from application development. In this way, application developers can't create roles with privileges for database administration for example.

Organizations and spaces also make it possible to develop roles with different purposes separately, for example, roles for database administration and roles for a specific functional area.

For more information about the Controller-based model, see the section on organizations and spaces.

Authorizations Needed for Role Development

The Container Object Owner

According to the authorization concept of the SAP HANA database, a user can only grant a privilege to a user directly or indirectly in a role if the following prerequisites are met:

- The user has the privilege him- or herself
- The user is authorized to grant the privilege to others (WITH ADMIN OPTION or WITH GRANT OPTION)

A user is also authorized to grant object privileges on objects that he or she owns.

The deployment of database objects with the HDI is based on a container model where each container corresponds roughly to a database schema. Each schema, and the database objects deployed into the schema, are owned by a dedicated technical database user called the **container object owner** (<container>#00). However, this user does not have any “external” privileges, for example system privileges or object privileges on objects in a container in a different space to the role-development container or in an external schema. These privileges must be granted to the container object owner explicitly. If the container object owner does not have all privileges, role deployment will fail with a “missing authorization” error.

There are a number of ways of granting the required privileges to the container object owner. These range from simply granting the privileges to the container object owner directly to more sophisticated methods involving a user-provided service. The various alternatives are described in detail in the document *Best Practices and Recommendations for Developing Roles in SAP HANA* (see Related Information).

i Note

The container object owner must always be granted privileges on external objects with the additional parameters WITH ADMIN OPTION or WITH GRANT OPTION.

The Role Developer

The role developer must have the roles and role collections required to access SAP Web IDE for SAP HANA and to develop in the relevant space and organization.

For more information about the authorization required for developing, see the *SAP HANA Developer Guide for XS Advanced Model*.

Granting and Revoking HDI Roles

It is not possible to grant and revoke deployed HDI roles using the GRANT and REVOKE SQL statements. Roles deployed to an HDI container are granted and revoked through the execution of the GRANT_CONTAINER_SCHEMA_ROLES and REVOKE_CONTAINER_SCHEMA_ROLES procedures of either the container's or the container group's API schema. The container administrator and the container group administrator are authorized to execute these procedures. For more information about these procedures and administrator roles, see the *SAP HANA Administration Guide*.

A role administrator (user with system privilege ROLE ADMIN) can also grant and revoke HDI roles, for example as follows: GRANT <role_schema_name>.<role_name> TO <user_name>, where <role_schema_name> is the HDI container name where the role was created.

Roles can be granted and revoked using the SAP HANA cockpit.

Dropping HDI Roles

It is not possible to drop a HDI role by dropping the runtime version of the role using the SQL statement DROP ROLE. The role must be undeployed within the container.

Auditing HDI Roles

When role objects are deployed for the first time, a runtime version of the role is created in the corresponding schema in the database using SQL. The same is true if roles are changed or undeployed. You can therefore audit activity related to HDI roles with audit actions CREATE ROLE, ALTER ROLE, and DROP ROLE.

Related Information

[Organizations and Spaces \[page 314\]](#)

[Auditing Activity in SAP HANA Systems \[page 240\]](#)

[Best Practices and Recommendations for Developing Roles in SAP HANA !\[\]\(003082e50e3009141f59bd5df831749f_img.jpg\)](#)

8.2.4 Repository Roles

In an SAP HANA XS classic environment, role developers create database roles as design-time objects in the built-in repository of the SAP HANA database using either the SAP HANA Web Workbench or the SAP HANA studio.

i Note

SAP HANA XS classic and the SAP HANA repository are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.

Roles created in the repository differ from roles created directly as runtime objects using SQL in several ways.

- [What authorization does a user need to grant privileges to a role? \[page 170\]](#)
- [What about the WITH ADMIN OPTION and WITH GRANT OPTION parameters? \[page 171\]](#)
- [How are repository roles granted and revoked? \[page 171\]](#)
- [How are repository roles dropped? \[page 171\]](#)
- [Can changes to repository roles be audited? \[page 171\]](#)

What authorization does a user need to grant privileges to a role?

According to the authorization concept of the SAP HANA database, a user can only grant a privilege to a user directly or indirectly in a role if the following prerequisites are met:

- The user has the privilege him- or herself
- The user is authorized to grant the privilege to others (WITH ADMIN OPTION or WITH GRANT OPTION)

A user is also authorized to grant object privileges on objects that he or she owns.

The technical user `_SYS_REPO` is the owner of all objects in the repository, as well as the runtime objects that are created on activation. This means that when you create a role as a repository object, you can grant the following privileges:

- Privileges that have been granted to the technical user `_SYS_REPO` and that `_SYS_REPO` can grant further
This is automatically the case for system privileges, package privileges, analytic privileges, and application privileges. Therefore, all system privileges, package privileges, analytic privileges, and application privileges can always be granted in design-time roles.
- Privileges on objects that `_SYS_REPO` owns
`_SYS_REPO` owns all activated objects. Object privileges on non-activated runtime objects must be explicitly granted to `_SYS_REPO`.

i Note

This is true even for objects belonging to schema `SYS`.

It is recommended that you use a technical user to do this to ensure that privileges are not dropped when the granting user is dropped (for example, because the person leaves the company).

The following table summarizes the situation described above:

| Privilege | Action Necessary to Grant in Repository Role |
|---|---|
| System privilege | None |
| Package privilege | None |
| Analytic privilege | None |
| Application privilege | None |
| SQL object on activated object (for example, attribute view, analytic view) | None |
| SQL object privilege on runtime object (for example, replicated table) | Grant privilege to user <code>_SYS_REPO</code> with WITH GRANT OPTION |

i Note

Technically speaking, only the user `_SYS_REPO` needs the privileges being granted in a role, not the database user who creates the role. However, users creating roles in the SAP HANA Web-based Development Workbench must at least be able to **select** the privileges they want to grant to the role. For this, they need either the system privilege `CATALOG READ` or the actual privilege to be granted.

What about the WITH ADMIN OPTION and WITH GRANT OPTION parameters?

When you create a role using SQL (that is, as a runtime object), you can grant privileges with the additional parameters WITH ADMIN OPTION or WITH GRANT OPTION. This allows a user who is granted the role to grant the privileges contained within the role to other users and roles. However, if you are implementing your authorization concept with privileges encapsulated within roles created in design time, then you do not **want** users to grant privileges using SQL statements. For this reason, it is not possible to pass the parameters WITH ADMIN OPTION or WITH GRANT OPTION with privileges when you model roles as repository objects.

Similarly, when you grant an activated role to a user, it is not possible to allow the user to grant the role further (WITH ADMIN OPTION is not available).

How are repository roles granted and revoked?

It is not possible to grant and revoke activated design-time roles using the GRANT and REVOKE SQL statements. Instead, roles are granted and revoked through the execution of the procedures GRANT_ACTIVATED_ROLE and REVOKE_ACTIVATED_ROLE. Therefore, to be able to grant or revoke a role, a user must have the object privilege EXECUTE on these procedures.

How are repository roles dropped?

It is not possible to drop the runtime version of a role created in the repository using the SQL statement DROP ROLE. To drop a repository role, you must delete it in the repository and activate the change. The activation process deletes the runtime version of the role.

Can changes to repository roles be audited?

The auditing feature of the SAP HANA database allows you to monitor and record selected actions performed in your database system. One action that is typically audited is changes to user authorization. If you are using roles created in the repository to grant privileges to users, then you audit the creation of runtime roles through activation with the audit action ACTIVATE REPOSITORY CONTENT.

Related Information

[SAP Note 2465027](#) 

8.2.4.1 Repository Roles in the Lifecycle of SAP HANA-Based Applications

Roles are an integral part of developing SAP HANA XS classic applications and their lifecycle. Developers create application-specific objects, including roles, in the repository of the development system. Content administrators transport applications as delivery units to the production system, where they are activated. User administrators grant activated roles to end users.

In application development scenarios, roles are developed like other application-specific artifacts and managed as part of overall application lifecycle management. Roles developed as part of the application encapsulate the privileges required by different user groups to use the application.

The following is a high-level overview of how applications, including application-specific roles, are developed and deployed:

1. Developers build the application by creating the required objects, including roles, in the repository of the development system.
All development objects, including roles, are stored in packages. Packages that belong to the same application are grouped together into a delivery unit (DU). DUs are the mechanism by which design-time objects in the repository are transported between two systems. They ensure that application-specific objects are transported consistently together within a system landscape.
2. Developers activate their development objects in the development system initially for testing purposes and finally to make them ready for transport.
The activation process makes the design-time objects available in runtime. In many cases, the runtime objects created are catalog objects, such as schemas, tables, views, and roles.
3. The content administrator transports the application DU from the development system to the production system. This activates the DU and creates runtime objects in the production system.
Content can be exported and imported using:
 - SAP HANA Application Lifecycle Manager
 - The change and transport system (CTS+) of the SAP NetWeaver ABAP application server
 - HANA Transport Container (HTC)The transport option used depends on the scenario. For example, CTS+ can be used to transport SAP HANA content in ABAP system landscapes where a CTS transport landscape is already in place.

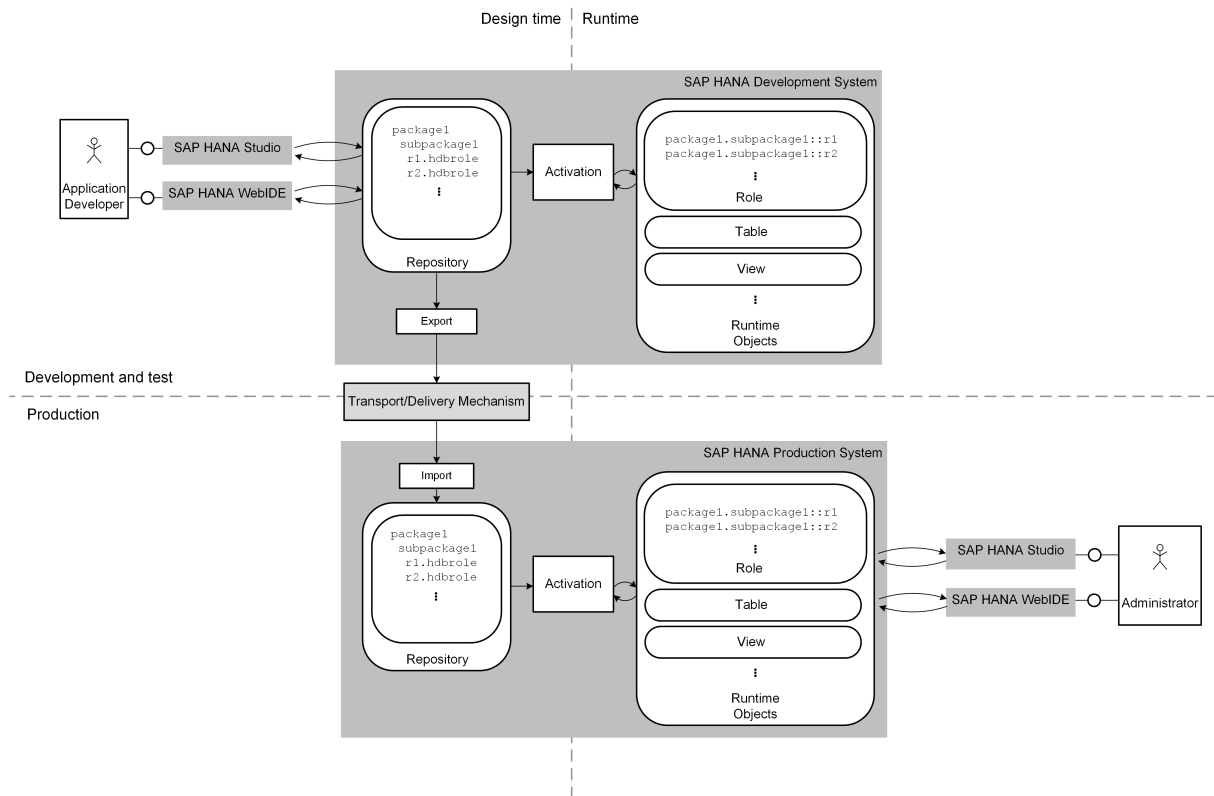
Once roles are available as runtime objects in the production system, the user administrator can grant them to end users.

Caution

The design-time version of a role in the repository and its activated runtime version should always contain the same privileges. In particular, additional privileges should not be granted to the activated runtime version of a role created in the repository. If a repository role is changed in runtime, the next time the role is activated in the repository, any changes made to the role in runtime will be reverted. It is therefore important that the activated runtime version of a role is not changed in runtime. Although it is not possible to change the activated runtime version of a repository role in the SAP HANA studio, there is no mechanism of preventing a user from doing this at the SQL level.

The following figure illustrates the process described above:

Lifecycle of Repository Roles



Related Information

[Change and Transport System \(Including CTS Plug-In\)](#)

[How to transport ABAP for SAP HANA applications with HTC](#)

8.2.4.2 Predefined Repository Roles

SAP HANA is delivered with SAP HANA content, a set of pre-installed software components implemented as SAP HANA Web applications, libraries, and configuration data, and developed on the SAP HANA XS, classic model. The privileges required to use a software component delivered as SAP HANA content are contained within repository roles delivered with the component itself.

For more information about the repository roles delivered with SAP HANA content, see *Components Delivered as SAP HANA Content*.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. 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.

i Note

No user has any predefined repository roles initially, except the user `_SYS_REPO` (as the owner of all repository content).

Related Information

[Components Delivered as SAP HANA Content \[page 363\]](#)

[Security of SAP HANA Content \[page 278\]](#)

[Package Privileges \[page 155\]](#)

8.3 Authorization in the Repository of the SAP HANA Database

The authorization concept of SAP HANA applies in the repository of the SAP HANA database.

With the SAP HANA Extended Services (SAP HANA XS) classic development model, developers of SAP HANA-based applications use the built-in repository for storing, versioning, and delivering design-time artifacts such as views, procedures, tables, roles, CDS entities, and Web content exposed via SAP HANA XS classic. The repository provides the basis for concepts like namespaces (through packages), versioning, transport in system landscapes, and software component delivery from SAP or independent software vendors to customers.

[Developer Authorization in the Repository \[page 175\]](#)

To ensure that the process of application development using the SAP HANA Extended Services (SAP HANA XS) classic model is secure, it is important that developers have access to only those repository objects that they actually need to work with.

[_SYS_REPO Authorization in the Repository \[page 176\]](#)

The technical user `_SYS_REPO` is the owner of all objects in the repository, as well as their activated runtime versions. `_SYS_REPO` must be explicitly authorized for objects that are not created in the repository but on which repository objects are modeled.

[Granting and Revoking Privileges on Activated Repository Objects \[page 177\]](#)

Only the `_SYS_REPO` user has privileges on objects in the repository. Therefore, only this user can grant privileges on them. Since no user can log on as `_SYS_REPO`, stored procedures are used to grant privileges instead.

8.3.1 Developer Authorization in the Repository

To ensure that the process of application development using the SAP HANA Extended Services (SAP HANA XS) classic model is secure, it is important that developers have access to only those repository objects that they actually need to work with.

The repository of the SAP HANA database consists of packages that contain design-time versions of various objects, such as attribute views, analytic views, calculation views, procedures, analytic privileges, roles, and so on. All repository methods that provide read or write access to content are secured with authorization checks. To allow developers to work with packages in the repository, they must have the required package, system, and object privileges.

The following table explains the privileges that developers require to work in the repository:

| Privilege Type | Privileges Required |
|--------------------|--|
| Package privileges | <p>The SAP HANA repository is structured hierarchically with packages assigned to other packages as sub-packages. If you grant privileges to a user for a package, the user is also automatically authorized for all corresponding sub-packages.</p> <p>In the SAP HANA repository, a distinction is made between native and imported packages. Native packages are packages that were created in the current system and should therefore be edited in the current system. Imported packages from another system should not be edited, except by newly imported updates. An imported package should only be manually edited in exceptional cases.</p> <p>Developers should be granted the following privileges for native packages:</p> <ul style="list-style-type: none"> • REPO.READ • REPO.EDIT_NATIVE_OBJECTS • REPO.ACTIVATE_NATIVE_OBJECTS • REPO.MAINTAIN_NATIVE_PACKAGES <p>Developers should only be granted the following privileges for imported packages in exceptional cases:</p> <ul style="list-style-type: none"> • REPO.EDIT_IMPORTED_OBJECTS • REPO.ACTIVATE_IMPORTED_OBJECTS • REPO.MAINTAIN_IMPORTED_PACKAGES |
| System privileges | <p>Developers require the following system privileges to be able to work in the repository:</p> <ul style="list-style-type: none"> • REPO.EXPORT • REPO.IMPORT • REPO.MAINTAIN_DELIVERY_UNITS • REPO.WORK_IN_FOREIGN_WORKSPACE • REPO.MODIFY_CHANGE, REPO.MODIFY_OWN_CONTRIBUTION, and REPO.MODIFY_FOREIGN_CONTRIBUTION <p>These privileges authorize the user to work with SAP HANA Change Recording, which is part of SAP HANA Application Lifecycle Management.</p> |
| Object privileges | <p>To be able to access the repository in the SAP HANA studio or another client, developers need the EXECUTE privilege on the database procedure SYS.REPOSITORY_REST.</p> |

Authorization for SAP HANA Web-based Developer Workbench

If developers are using the SAP HANA Web-based Development Workbench, the privileges required for building and testing development artifacts as well tool access are bundled into the following roles:

- `sap.hana.xs.ide.roles::EditorDeveloper`
- `sap.hana.xs.debugger::Debugger`

For more information, see *SAP HANA Web-Based Development Workbench* in the *SAP HANA Developer Guide (For Web Workbench)*.

Authorization for SAP HANA Application Lifecycle Management

SAP HANA Application Lifecycle Management is a Web-based tool that runs in SAP HANA XS classic. Application developers use this tool to create products, delivery units, packages, and basic application components, while administrators use it to set up the transport of delivery units, start and monitor transports, and upload or download delivery unit archives.

These tasks require different combinations of various privileges. Dedicated roles are available and can be granted to users based on their function (for example, `sap.hana.xs.lm.roles::Administrator`). For more information, see *SAP HANA Application Lifecycle Management*.

Related Information

[System Privileges \(Reference\) \[page 124\]](#)

8.3.2 `_SYS_REPO` Authorization in the Repository

The technical user `_SYS_REPO` is the owner of all objects in the repository, as well as their activated runtime versions. `_SYS_REPO` must be explicitly authorized for objects that are not created in the repository but on which repository objects are modeled.

The repository of the SAP HANA database consists of packages that contain design-time versions of various objects, such as attribute views, analytic views, calculation views, procedures, analytic privileges, roles, and so on. Design-time objects must be activated to become runtime objects so that they can be used by end users of SAP HANA and the SAP HANA database.

Inside the repository, only the technical user `_SYS_REPO` is used. Therefore, this user is the owner of the objects created in the repository and initially is the only user with privileges on these objects. This includes the following objects:

- All tables in the repository schema (`_SYS_REPO`)
- All activated objects such as procedures, views, analytic privileges, and roles

i Note

This does not apply in the case of objects that have been activated using the data preview on intermediate nodes in calculation models. These objects are activated and owned by the user who does the data preview.

Objects in the repository can be modeled on data objects that are not part of design time, such as tables that are used in replication scenarios. `_SYS_REPO` does not automatically have authorization to access these objects. `_SYS_REPO` must therefore be granted the `SELECT` privilege (with grant option) on all data objects behind all objects modeled in the repository. If this privilege is missing, the activated objects will be invalidated. This is true even for objects belonging to schema `SYS`.

8.3.3 Granting and Revoking Privileges on Activated Repository Objects

Only the `_SYS_REPO` user has privileges on objects in the repository. Therefore, only this user can grant privileges on them. Since no user can log on as `_SYS_REPO`, stored procedures are used to grant privileges instead.

Using stored procedures and a technical user for privilege management is beneficial for the following reasons compared to the standard SQL mechanism using the `GRANT` and `REVOKE` statements:

- To be able to grant a privilege, a user must have the privilege and be authorized to grant it further. This is not the case when procedures are used. Any user who has the `EXECUTE` privilege on the relevant grant procedure can grant privileges.

i Note

Any user with the system privilege `ROLE ADMIN` can also grant a role using the `GRANT` statement.

- If a user grants a privilege or role using the `GRANT` statement, the privilege or role is automatically revoked when the grantor is dropped or loses the granted privileges.
- Only the grantor can revoke the privilege. With the stored procedures approach, any user with the `EXECUTE` privilege on the relevant revoke procedure can revoke a granted privilege, regardless of the grantor. If the grantor is dropped, none of the privileges that he or she granted are revoked.

i Note

Any user with the system privilege `ROLE ADMIN` can revoke a role using the `REVOKE` statement regardless of the grantor.

When the SAP HANA cockpit is used for privilege management, it automatically chooses the suitable method for granting and revoking privileges and roles. So if privileges on activated objects are being granted or revoked, the procedures are used.

⚠ Caution

Users who can change and activate objects as well as grant privileges on activated objects have access to all SAP HANA content.

Related Information

[Stored Procedures Used to Grant/Revoke Privileges on Activated Repository Objects \[page 178\]](#)

8.3.3.1 Stored Procedures Used to Grant/Revoke Privileges on Activated Repository Objects

Stored procedures, which exist in the `_SYS_REPO` schema, are used to grant and revoke privileges on activated modeled objects, analytic privileges, application privileges, and roles.

i Note

Public synonyms of these procedures exist. Therefore, these procedures can be called without specifying the schema `_SYS_REPO`.

| Activated Object Type | Procedure Call for Grant and Revoke |
|--|--|
| Modeled objects, such as calculation views | <pre>CALL GRANT_PRIVILEGE_ON_ACTIVATED_CONTENT ('<object_privilege>', '<object>', '<user or role>') CALL REVOKE_PRIVILEGE_ON_ACTIVATED_CONTENT ('<object_privilege>', '<object>', '<user or role>')</pre> |
| Schema containing modeled objects | <pre>CALL GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT ('<analytic_privilege>', '<user or role>') CALL REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT ('<analytic_privilege>', '<user or role>')</pre> |
| Analytic privilege | <pre>CALL GRANT_ACTIVATED_ANALYTICAL_PRIVILEGE ('<analytic_privilege>', '<user or role>') CALL REVOKE_ACTIVATED_ANALYTICAL_PRIVILEGE ('<analytic_privilege>', '<user or role>')</pre> |
| Application privilege | <pre>CALL GRANT_APPLICATION_PRIVILEGE ('<application_privilege>', '<user or role>') CALL REVOKE_APPLICATION_PRIVILEGE ('<application_privilege>', '<user or role>')</pre> |
| Role | <pre>CALL GRANT_ACTIVATED_ROLE ('<role>', '<user or role>') CALL REVOKE_ACTIVATED_ROLE ('<role>', '<user or role>')</pre> |

i Note

Object names that are not simple identifiers must be enclosed between double quotes, for example:

```
CALL GRANT_APPLICATION_PRIVILEGE ('"com.acme.myApp::Execute"', 'User')
```

This does not apply to the procedures GRANT_ACTIVATED_ROLE and REVOKE_ACTIVATED_ROLE. The role being granted or revoked must not be enclosed in double quotes, for example:

```
CALL GRANT_ACTIVATED_ROLE ('acme.com.data::MyUserRole', 'User')
```

For all procedures, the user or role to whom/from whom a privilege or role is being granted/revoked must not be enclosed between double quotes.

8.4 Cross-Database Authorization in Tenant Databases

Read-only queries between tenant databases are possible through the association of the requesting user with a remote identity on the remote database(s). Cross-database access is not enabled by default and must be configured before such user mappings can be set up.

Every tenant database is self-contained with its own isolated set of database users and isolated database catalog. However, to support for example cross-application reporting, cross-database SELECT queries are possible. This means that database objects such as tables and views can be local to one database but be read by users from other databases in the same system.

A user in one database can run a query that references objects in another database if the user is associated with a sufficiently privileged user in the remote database. This associated user is called a remote identity. This is the user who executes the query (or part of the query) in the remote database and therefore the user whose authorization is checked.

For more information about which object types on remote databases can be accessed using this mechanism and which local object types can access remote database objects, see *Cross-Database Access* in the *SAP HANA Administration Guide*.

❁ Example

Assume that we have a system with 2 tenant databases: DB1 and DB2.

USER2 in DB2 wants to query the table SCHEMA1.TABLE1 in DB1, for example, `SELECT * FROM DB1.SCHEMA1.TABLE1`.

This can be achieved as follows:

1. The administrator of DB1 creates a user in DB1 with a remote identity in DB2:

```
CREATE USER USER1 WITH REMOTE IDENTITY USER2 AT DATABASE DB2
```

2. The administrator of DB1 grants user USER1 the privileges required to read the table SCHEMA1.TABLE1:

```
GRANT SELECT ON SCHEMA1.TABLE1 TO USER1 [WITH GRANT OPTION]
```

Now, USER2 in DB2 can select from SCHEMA1.TABLE1 in DB1.

For more information about the syntax notation, see *CREATE USER* in the *SAP HANA SQL and System Views Reference*.

Things to Note About Remote Identities

- A user can be the remote identity for only one user in another database.
- An existing user can be assigned a remote identity using the ALTER USER statement.
- The association between a user and a remote identity is unidirectional. In the above example, USER2 can access SCHEMA1.TABLE1 in DB1 as USER1, but USER1 cannot access objects in DB2 as USER2.
- Only the SELECT privileges of the user in the remote database are considered during a cross-database query. Any other privileges the remote user may have are ignored.
- Before users with remote identities can be created, an administrator must enable cross-database access for the system in the system database and specify which databases can communicate with one another. For more information, see *Enable and Configure Cross-Database Access* in the *SAP HANA Administration Guide*.
Users receive a `Not authorized` error if they attempt a cross-database operation that is not supported by the current configuration.

System Views for Monitoring Cross-Database Authorization

The following system views contain information about cross-database authorization in a tenant database:

- `USERS (SYS)`
The column `REMOTE_USER` indicates whether or not a particular user in the local database has remote identities in other databases.
- `REMOTE_USERS (SYS)`
This system view shows which local users can be used by users on other databases for cross-database query execution.

i Note

The system views `EFFECTIVE_PRIVILEGES` and `ACCESSIBLE_VIEWS` **do not** include privileges that a user has through a remote identity.

8.5 LDAP Group Authorization

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing directory services. If you use an LDAP-compliant directory server to manage users and their access to resources, you can leverage LDAP group membership to authorize users.

LDAP groups can be mapped to SAP HANA roles. This allows SAP HANA to determine which roles to assign to users based on their membership in one or more LDAP groups, either directly or indirectly through nested groups. Users' access to requested resources is then determined by the privileges defined in the SAP HANA roles.

LDAP group membership can be used to authorize existing SAP HANA users. If you're also using the LDAP server for user authentication in SAP HANA, the required user can be created automatically. For more information, see the section on LDAP user authentication.

Implementation Considerations

- LDAP group authorization is not supported for cross-database queries between tenant databases. The remote identity of a user must be authorized using the standard mechanism.
- LDAP group authorization is not supported for SAP HANA Extended Application Services, advanced users, that is application users, including those that are created in SAP HANA when SAP HANA is used as identity provider. In addition, the use of SAP HANA XS advanced roles and role collections is not affected by LDAP authorization.
- In an active/active (read enabled) scenario, users are always authenticated on the primary system even if they are connecting to the secondary system, either directly or on the basis of statement routing. This means that roles are granted to the connecting user (based on LDAP group membership) on the primary system. The role assignment is then propagated on the secondary system. However, the user may not immediately have all roles on the secondary system due to a lag in propagation.
- LDAP group authorization can be disabled in tenant databases if it is not required. If any existing users are configured for LDAP group authorization when the feature is disabled, they won't be able to log on. However, the authorization mode of these users can be changed to the standard mechanism.

Related Information

[LDAP Group Authorization for Existing Users \[page 181\]](#)

[Restricted Features in Tenant Databases \[page 358\]](#)

[Auditing Activity in SAP HANA Systems \[page 240\]](#)

[LDAP User Authentication \[page 114\]](#)

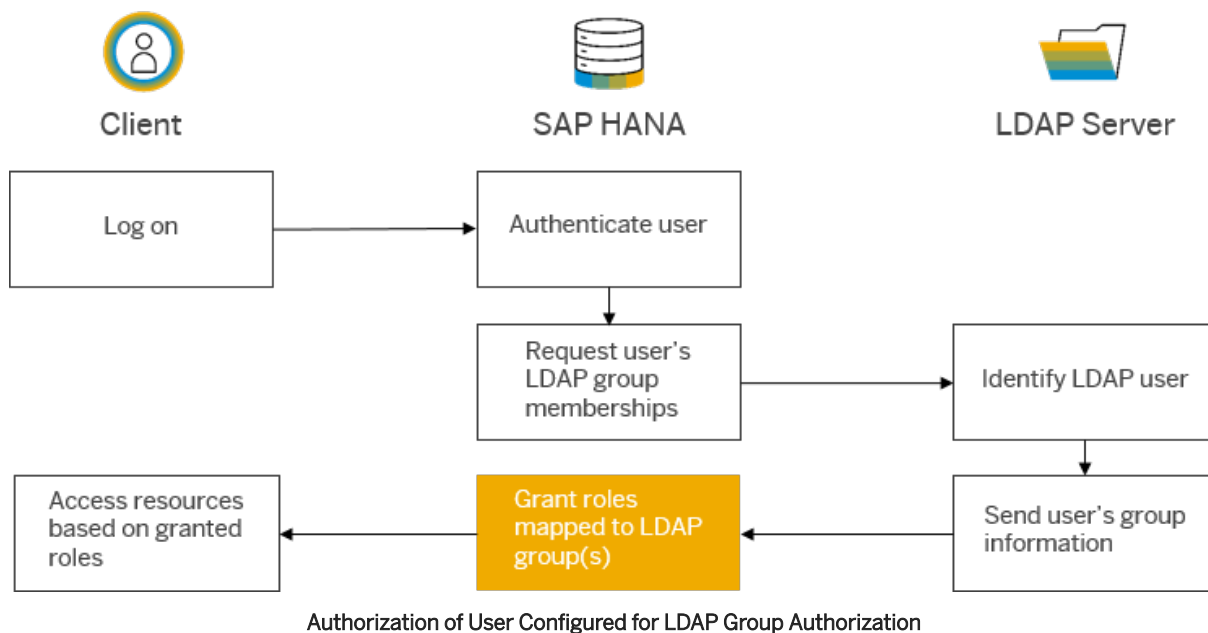
8.5.1 LDAP Group Authorization for Existing Users

Use LDAP group membership to authorize existing SAP HANA database users.

- [Overview \[page 181\]](#)
- [Implementing LDAP Group Authorization \[page 182\]](#)
- [Role Reuse Duration \[page 184\]](#)

Overview

The following figures illustrates how an SAP HANA user is authorized on the basis of his or her LDAP group membership.



i Note

If the user is not a member of any LDAP groups, or the groups of which he or she is a member are not mapped to any SAP HANA roles, user logon fails even if authentication was successful.

i Note

SAP HANA supports several user authentication mechanisms, including LDAP authentication. When LDAP authentication is used, the required database user can be created automatically and authorized based on LDAP group membership. For more information about how this works, see the section on LDAP user authentication.

Implementing LDAP Group Authorization

To implement LDAP group authorization, the following high-level steps are required.

Map LDAP groups to SAP HANA roles

LDAP groups can be mapped to SAP HANA catalog roles using the `CREATE ROLE` or `ALTER ROLE` statements. A role that has an LDAP group mapping can be granted to users and other roles as usual. If the role is deleted, it is also revoked from users as usual. Mappings of LDAP groups to this role are also deleted.

❖ Example

```
CREATE ROLE Securities_DBA LDAP GROUP
"cn=Securities_DBA,OU=Application,OU=Groups,ou=DatabaseAdmins,cn=Users,o=verylarg
ebank.com"
```

i Note

LDAP group mappings cannot be defined in the definition of design-time roles (database artifact with file suffix `.hdbrole`). This includes both roles created in the built-in repository of the SAP HANA database (repository roles), and those created using the SAP Web IDE and deployed using SAP HANA deployment infrastructure (SAP HANA DI). However, for roles deployed using SAP HANA DI, it is possible to map LDAP groups to the activated catalog role.

Configure connection to LDAP server in SAP HANA

You set up the LDAP server connection by creating an LDAP provider in the SAP HANA database with the `CREATE LDAP PROVIDER` or `ALTER LDAP PROVIDER` statements.

Access to the LDAP server takes place using an LDAP server user with permission to perform searches as specified by the user look-up URL. The credential of this user is stored in the secure internal credential store.

Communication between SAP HANA and the LDAP server can be secured using the TLS/SSL protocol or Secure LDAP protocol (LDAPS). For more information, see the section on secure communication between SAP HANA and an LDAP directory server.

Configure SAP HANA users for LDAP group authorization

To be granted roles on the basis of LDAP group membership, a user must be configured for authorization mode `LDAP`. You can configure the authorization mode for a user in the SAP HANA cockpit or using the `CREATE USER` or `ALTER USER` statement:

❁ Example

```
CREATE USER USER1 PASSWORD enSMRia3s4hS AUTHORIZATION LDAP
```

If the LDAP server is also being used for user authentication, the LDAP provider in SAP HANA can be configured for automatic user creation. In this way, the required database user is automatically created with authorization mode `LDAP`.

A user with authorization mode `LDAP` is granted roles exclusively based on their LDAP group membership. It is not possible to grant such a user other roles or privileges directly. The default user authorization mode is `LOCAL`. This means that the user must be granted roles and privileges by a user administrator.

→ Tip

To see which authorization mode is configured for a user, refer to the `AUTHORIZATION_MODE` column of the `USERS` system view.

When the authorization mode of a user is switched, the following changes happen:

From LOCAL to LDAP

- Roles based on LDAP group membership are determined and granted the next time the user logs on.
- Previously granted roles and privileges are revoked, except roles that were granted by user `SYS`, for example the `PUBLIC` role, as well as the `CREATE ANY` privilege on user's own schema.

From LDAP to LOCAL

- Roles that are granted based on LDAP group membership are revoked.
- The time of the last successful LDAP authorization refresh is set to `NULL` in the system view `LDAP_USERS`.

For more information about configuring LDAP authorization, see the *SAP HANA Administration Guide*.

Role Reuse Duration

To avoid users' LDAP group membership having to be re-evaluated and roles assigned every time they log on, role assignments are stored during the first connection and reused for subsequent connections for a specified period of time, referred to as reuse duration.

When this reuse duration expires, LDAP group membership is re-evaluated and roles are reassigned during a subsequent user connection. In this way, high-frequency short-duration connections are efficiently handled.

If the user is logged on in multiple concurrent sessions when the reuse duration expires, LDAP group membership is re-evaluated and roles reassigned for a subsequent user session. Remaining user sessions reuse the role information obtained by the new session.

The default reuse duration is 240 minutes (4 hours) and can be configured using the system property `[authorization] ldap_authorization_role_reuse_duration` in the `indexserver.ini` configuration file.

- If `ldap_authorization_role_reuse_duration` is set to 0, users' LDAP group membership is re-evaluated and roles are granted on every logon.
- If `ldap_authorization_role_reuse_duration` is set to -1, users' LDAP group membership is never re-evaluated, initially granted roles are never revoked.

❁ Example

If the reuse duration is set to 60 minutes and the user logs on for the first time at 9:00 am, any new sessions for the user do not have to contact the LDAP server to re-evaluate role assignments until 10:00 am.

Related Information

[LDAP User Authentication \[page 114\]](#)

[Certificate Management in SAP HANA \[page 261\]](#)

[Secure Internal Credential Store \[page 217\]](#)

[Secure Communication Between SAP HANA and an LDAP Directory Server \[page 57\]](#)

8.6 Shared Business Authorizations in SAP HANA

The basic layer of authorization for ABAP-based SAP applications such as S/4 HANA is provided by "authorization objects" in the SAP NetWeaver Application Server for ABAP. It is possible to create analytic privileges in SAP HANA that reuse these authorizations for read access.

Analytic privileges that reuse ABAP authorization objects can be used to secure access to data in SAP HANA that is accessed either directly or through SAP smart data access. When SAP HANA reuses ABAP

authorization objects, it is directly accessing user data in the tables maintained by the ABAP system. Therefore, you can also use the mechanism described here for replication scenarios in which you replicate data from an ABAP system (including user and authorization information) into an SAP HANA system.

i Note

When you reuse ABAP authorization objects in SAP HANA, access to views in SAP HANA is based on the authorizations as they are maintained in the ABAP system. This means that the names of the database user in SAP HANA executing the query and the name of the user on the ABAP server have to match.

Being able to create analytic privileges based on ABAP authorizations means that you can provide consistent access to your SAP business data from both SAP applications and new applications built using the SAP HANA extended application services, advanced model. It also reduces the effort of maintaining your authorization model.

Securing a View with Shared Business Authorizations

To secure a view using an analytic privilege that reuses ABAP authorization objects, take the following high-level steps.

1. In the ABAP system, determine the relevant ABAP authorization objects for your scenario.

→ Tip

Objects are bundled into roles. You could look at the roles relevant for your scenario and then see what objects are in them.

2. In SAP HANA, create a view registered for an authorization check based on analytic privileges (STRUCTURED PRIVILEGE CHECK parameter).

≡ Sample Code

```
CREATE VIEW TABLEOWNER.SALES_VIEW as (select * from SALES_DATA) WITH  
STRUCTURED PRIVILEGE CHECK;
```

3. Create a helper procedure that calls the built-in procedure `SYS.GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION`. The input parameters passed to this procedure generate the filter condition for restricting access based on the specified authorization objects for the current user and client. For more information, see `GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION (SYS)`.

i Note

The procedure must be executable by `_SYS_REPO`.

≡ Sample Code

```
CREATE PROCEDURE TABLEOWNER.SALES_CONDITION_PROVIDER(OUT result_condition  
NCLOB) LANGUAGE SQLSCRIPT SQL SECURITY DEFINER READS SQL DATA AS  
BEGIN  
CALL GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION  
(
```

```
'A_TEST_SCHEMA',
'CHECKID1 AND NOT CHECKID2',
'{
  "data":
  {
    "CHECKID1":
    {
      "authobj": "OBJ1",
      "filter": [{"key": "ACTVT", "valueList": ["03"]}],
      "mappings": [{"fieldName": "TEST01",
"mappedName": "LIFECYCLE_STATUS"}]
    },
    "CHECKID2":
    {
      "authobj": "OBJ2",
      "filter": [{"key": "ACTVT", "valueList": ["03"]}],
      "mappings": [{"fieldName": "TEST02",
"mappedName": "BILLING_STATUS"}]
    }
  }
}',
result_condition);
END;
GRANT EXECUTE ON TABLEOWNER.SALES_CONDITION_PROVIDER_NAME TO _SYS_REPO;
```

4. Create an SQL-based analytic privilege, using the helper procedure to specify the filter condition (CONDITION PROVIDER clause).

Sample Code

```
CREATE STRUCTURED PRIVILEGE SALES_AP FOR SELECT ON TABLEOWNER.SALES_VIEW
CONDITION PROVIDER TABLEOWNER.SALES_CONDITION_PROVIDER;
```

5. Include the analytic privilege in a role and assign the role to end users.

Result

When a user requests access to data stored in the views, an authorization check based on the analytic privilege is performed and the data returned to the user is filtered according to their ABAP authorizations. The ABAP authorizations to be used to authorize access are determined by the value of the XS_APPLICATIONUSER session context variable. Therefore, the value of XS_APPLICATIONUSER must match a user name in the ABAP authorization data. By default, XS_APPLICATIONUSER is set to the name of the database user that has opened the connection to SAP HANA. In SAP HANA XS advanced scenarios, XS_APPLICATIONUSER is automatically set to the name of the business user who has logged in to the XSA application.

Related Information

[ABAP Authorization Concept](#)

[GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION \(SYS\) \[page 187\]](#)

[Structure of SQL-Based Analytic Privileges \[page 137\]](#)

8.6.1 GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION (SYS)

SQLScript procedure to generate a filter condition based on ABAP authorization objects that can be used in the CONDITION PROVIDER clause of an analytic privilege

Procedure Call

```
CALL SYS.GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION(<parameter_list>)
```

Input Parameters

- Schema in which the SAP authorization tables UST12 and USRBF2 reside
- An expression that contains at least one CHECKID

i Note

Multiple CHECKIDs may be concatenated. Expression can contain AND, OR and NOT Boolean operators.

- A JSON-formatted string that contains the information required to translate each ABAP authorization object referenced. For each CHECKID, the following must be specified:
 - `authobj`
This specifies the name of the ABAP authorization object to which the CHECKID is mapped. This may contain an empty value if the CHECKID itself is the name of the ABAP authorization object.
 - `filter`
This allows you to specify fixed conditions that the user's ABAP authorizations need to match. It is a list of required fields (`key`) and their values (`values`).
 - `mappings`
This specifies the mapping of FIELD value (`fieldName`) in the ABAP authorization object to column name (`mappedName`) of the target view to be protected.

Output Parameters

An NCLOB that contains the SQL filter condition

i Note

The procedure returns 1>1 if the user has no permissions and 1=1 if the user has full access.

Examples

❖ Example

Input:

```
CALL SYS.GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION (
'A_TEST_SCHEMA',
'CHECKID1',
'{"data":
  {
    "CHECKID1":
    {
      "authobj":"OBJ1",
      "filter":[{"key":"ACTVT","valueList":["03"]}],
      "mappings":[{"fieldName":"SACMTSOID", "mappedName":"SO_ID"},
{"fieldName":"SACMTSOLCS", "mappedName":"LIFECYCLE_STATUS"}]
    }
  }
}',
?)
```

Related Information

[Shared Business Authorizations in SAP HANA \[page 184\]](#)

8.7 Data Masking

Data masking represents an additional layer of access control that can be applied to tables and views. A column mask protects sensitive or confidential data in a particular column of a table or view by transforming the data in such a way that it is only visible partially or rendered completely meaningless for an unprivileged user, while still appearing real and consistent.

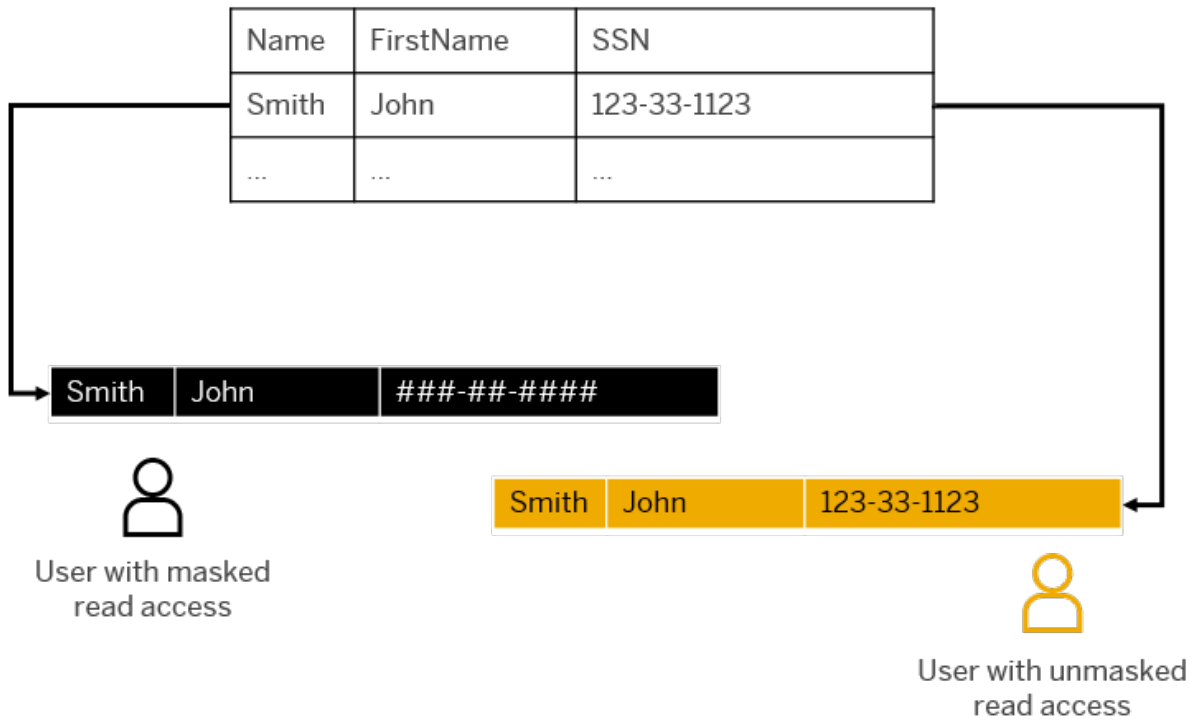
Unlike the all-or-nothing approach of authorization in which an unprivileged user gets a "not authorized" error, a user who is not privileged to see masked data does not get an error but rather sees only the masked data.

i Note

A user still needs the SELECT privilege on a table or view with data masking. Only a user who also has the UNMASKED object privilege for the table or view is able to see the original data.

Masking is useful in situations where strict data protection regulations apply and a second layer of protection is required on top of authorization to safeguard specific data. For example, a highly-privileged administrator has access to company employee data but instead of seeing the actual content of the column containing social security numbers, he sees masked values instead (for example, ###-##-####).

Employee Data



Supported Objects

You can specify a mask for columns in rowstore and columnstore tables, as well as in SQL and calculations views.

Masking supports the following string data types:

- VARCHAR
- NVARCHAR
- CHAR
- STRING
- NSTRING
- SHORTEXT
- ALPHANUM

It is also possible to nest views containing masked columns.

Related Information

[Masks and Authorization in Masked Tables and Views \[page 190\]](#)

8.7.1 Masks and Authorization in Masked Tables and Views

A mask specifies how the data in a column should be returned. It is specified directly in the table or view definition. The object privilege `UNMASKED` controls the visibility of unmasked data. Access to masked data in view hierarchies is determined by the authorization of the underlying object owner (definer mode).

Mask Expression

A mask specifies how the data in a column should be returned. It is specified directly in the object definition using the `CREATE TABLE` and `CREATE VIEW` statements, for example:

```
CREATE VIEW <view_name> (<column_name_list>) AS <subquery> WITH MASK
(<column_name> USING '<mask_expression>');
```

The mask expression can be specified using constants or built-in functions. However, it must not alter the data type or length of the original data. This ensures that masked data appears real and consistent.

❁ Example

Create new SQL view with column mask as constant expression:

```
CREATE VIEW V(FirstName, Name, MaskedSSN, MaskedCreditcard) AS SELECT
FirstName, Name, SSN, Creditcard FROM T
WITH MASK (MaskedSSN USING 'XXXX-XX-' || RIGHT(MaskedSSN, 4),
MaskedCreditcard USING 'XXXX-XXXX-XXXX-XXXX');
```

The `ALTER VIEW` or `ALTER TABLE` statement can be used to add, remove, or change a column mask in an existing table or view.

For more information about the syntax of the `MASK` clause, see the relevant statements in the *SAP HANA SQL and Systems View Reference*.

→ Tip

You can add a mask to a column when modeling calculation views using the SAP Web IDE for SAP HANA.

Authorization in Masked Tables and Views

Simple Access

Access to tables or views containing masked columns is controlled by the usual object privileges, for example, `SELECT` for read access. Data in masked columns is returned in line with the user's privileges, but it is always masked, regardless of query type.

The ability to see the real, unmasked data is controlled by the object privilege `UNMASKED`.

Note

Not having the `UNMASKED` privilege does not result in an authorization error, assuming the user is otherwise sufficiently authorized.

As an object privilege, `UNMASKED` can initially be granted only by the object owner or the owner of the schema that contains the object. For more information, see the section on object privileges.

The following table illustrates how different combinations of the `SELECT` and `UNMASKED` control what a user sees on simple read access to a masked object:

| | | SELECT privilege | |
|--------------------|-------------|------------------|-------------|
| | | Not granted | Granted |
| UNMASKED privilege | Not granted | Not authorized | ###-##-#### |
| | Granted | Not authorized | 123-33-1123 |

Authorization for Simple Read Access

Access to Views with Dependencies

If a view is dependent on another view, it is not only the accessing user's privileges that are considered. The privileges of underlying view owners are also evaluated. Definer mode applies.

Assume user A creates a view, V1, as follows:

```
CREATE VIEW V1 (FirstName,
Name,
MaskedCreditcard1) AS SELECT
FirstName,
Name,
Creditcard
FROM T WITH MASK ( MaskedCreditcard1 USING mask_expression1 );
```

User B creates a view, V2, on top of V1 as follows:

```
CREATE VIEW V2 (FirstName,
Name,
MaskedCreditcard2) AS SELECT
FirstName,
Name,
MaskedCreditcard1
FROM V1 with MASK ( MaskedCreditcard2 USING mask_expression2 );
```

Now, if an end user queries the top-level view (V2), the masking of returned data depends on the authorization of both the owner of V2 (user B) and the end user.

Assume that the end user has read access, in other words: the end user has `SELECT` on V2 and user B has `SELECT WITH GRANT OPTION` on V1. How data in the masked column is returned to the end user depends on who has or doesn't have the `UNMASKED` privilege as follows:

| | V2 owner | End user | | Result to End User |
|---------------------------|-------------------|-------------------|---|--|
| UNMASKED Privilege | Not granted on V1 | Not granted on V2 | → | Data from V1 is first masked with V1 mask expression, and then masked again with V2 mask expression. |
| | Granted on V1 | Not granted on V2 | | Data from V1 is masked with V2 mask expression. |
| | Not granted on V1 | Granted on V2 | | Data from V1 is masked with V1 mask expression. |
| | Granted on V1 | Granted on V2 | | Data is not masked. |

Authorization for Views with Dependencies

Access with Analytic Privileges

If a view containing masked columns is also secured with analytic privileges, the user must have an applicable analytic privilege in addition to the `SELECT` privilege. If this is the case, the authorization check first filters the data based on the conditions specified in the analytic privilege, and then masks any data in masked columns. This means that a user who also has the `UNMASKED` privilege sees unmasked data, but still only the data that he is authorized to see as specified in the analytic privilege.

It is possible to build views containing masked columns and analytic privileges on top of each other. Data is masked or filtered in accordance with the masks and filter conditions of the underlying view. Definer mode applies.

Related Information

- [Example: Masking Data Using a Built-In Procedure \[page 193\]](#)
- [Example: Masking Data in a View with Structured Privilege Check \[page 194\]](#)
- [Example: Masking Data in View Hierarchy with Structured Privilege Check \[page 195\]](#)

8.7.2 Example: Masking Data Using a Built-In Procedure

This example shows how to mask data in a view using a mask expression based on a built-in procedure.

Procedure

1. User `data_owner` creates a table (`credit_tab`) containing credit card data:

```
CREATE TABLE credit_tab (Name varchar(20), CREDIT_CARD varchar(19));
INSERT INTO credit_tab values ('John', '1111-1111-1111-1111');
INSERT INTO credit_tab values ('James', '2222-2222-2222-2222');
```

2. User `mask_owner` defines a mask as a built-in function and grants user `data_owner` authorization to execute:

```
CREATE FUNCTION credit_mask(input varchar(19)) RETURNS output VARCHAR(19)
LANGUAGE SQLSCRIPT
AS
BEGIN
    output = LEFT(:input,4) || '-XXXX-XXXX-' || RIGHT(:input,4);
END;
GRANT EXECUTE ON credit_mask TO data_owner;
```

3. User `data_owner` creates a view on the credit card table (`credit_view`), masking the credit card number column using the built-in function:

```
CREATE VIEW credit_view AS SELECT * FROM credit_tab
WITH MASK (CREDIT_CARD USING mask_owner.credit_mask(credit_card));
```

4. User `data_owner` grants user `end_user` access to the view:

```
GRANT SELECT ON credit_view TO end_user;
```

Results

When user `end_user` selects from the credit card view, the data is masked as follows:

```
;NAME ;CREDIT_CARD
1;John ;1111-XXXX-XXXX-1111
2;James;2222-XXXX-XXXX-2222
```

8.7.3 Example: Masking Data in a View with Structured Privilege Check

This example shows the behavior of a view that contains a masked column and is also secured with an analytic privilege.

Prerequisites

User `mask_owner` has the system privilege `STRUCTUREDPRIVILEGE ADMIN`.

Procedure

1. User `data_owner` creates a table (`credit_tab`) containing credit card data:

```
CREATE TABLE credit_tab (Name varchar(20), CREDIT_CARD varchar(19));
INSERT INTO credit_tab values ('John', '1111-1111-1111-1111');
INSERT INTO credit_tab values ('James', '2222-2222-2222-2222');
```

2. User `mask_owner` defines a mask as a built-in function and grants user `data_owner` authorization to execute:

```
CREATE FUNCTION credit_mask(input varchar(19)) RETURNS output VARCHAR(19)
LANGUAGE SQLSCRIPT
AS
BEGIN
    output = LEFT(:input,4) || '-XXXX-XXXX-' || RIGHT(:input,4);
END;
GRANT EXECUTE ON credit_mask TO data_owner;
```

3. User `data_owner` creates a view (`credit_view`) on the credit card table, masking the credit card number column using the built-in function and further securing the view with a structured privilege check:

```
CREATE VIEW credit_view AS SELECT * FROM credit_tab
WITH MASK (CREDIT_CARD USING mask_owner.credit_mask(credit_card))
STRUCTURED PRIVILEGE CHECK;
```

4. User `data_owner` grants select authorization to user `end_user`:

```
GRANT SELECT ON credit_view TO end_user;
```

5. User `mask_owner` creates an analytic privilege for `credit_view` that allows access to the row containing the credit number `1111-1111-1111-1111`:

```
CREATE STRUCTURED PRIVILEGE credit_ap for SELECT ON data_owner.credit_view
WHERE CREDIT_CARD = '1111-1111-1111-1111';
```

6. User `mask_owner` grants the analytic privilege `credit_ap` to user `end_user`:

```
GRANT STRUCTURED PRIVILEGE credit_ap TO end_user;
```

Results

When user `end_user` selects from the credit card view, the data is first filtered in line with the analytic privilege and then masked:

```
;NAME ;CREDIT_CARD  
1;John ;1111-XXXX-XXXX-1111
```

8.7.4 Example: Masking Data in View Hierarchy with Structured Privilege Check

This example shows the behavior of a view secured with an analytic privilege built on top of a view with a masked column.

Prerequisites

User `mask_owner` has the system privilege `STRUCTUREDPRIVILEGE ADMIN`.

Procedure

1. User `data_owner1` creates a table (`credit_tab`) containing credit card data:

```
CREATE TABLE credit_tab (Name varchar(20), CREDIT_CARD varchar(19));  
INSERT INTO credit_tab values ('John', '1111-1111-1111-1111');  
INSERT INTO credit_tab values ('James', '2222-2222-2222-2222');
```

2. User `mask_owner` defines a mask as a built-in function and grants user `data_owner1` authorization to execute:

```
CREATE FUNCTION credit_mask(input varchar(19)) RETURNS output VARCHAR(19)  
LANGUAGE SQLSCRIPT  
AS  
BEGIN  
    output = LEFT(:input,4) || '-XXXX-XXXX-' || RIGHT(:input,4);  
END;  
GRANT EXECUTE ON credit_mask TO data_owner1;
```

3. User `data_owner1` creates a view on the credit card table (`credit_view_base`), masking the credit card number column using the built-in function:

```
CREATE VIEW credit_view_base AS SELECT * FROM credit_tab  
WITH MASK (CREDIT_CARD USING mask_owner.credit_mask(credit_card));
```

4. User `data_owner1` grants select authorization on `credit_view_base` to user `data_owner2`:

```
GRANT SELECT ON credit_view_base TO data_owner2;
```

5. User `data_owner2` creates a top-level view on the base view, securing it with a structured privilege check:

```
CREATE VIEW credit_view_top AS SELECT * from data_owner1.credit_view_base  
WITH STRUCTURED PRIVILEGE CHECK;
```

i Note

User `data_owner2` does not have the UNMASKED privilege on `credit_view_base`.

6. User `data_owner2` grants select authorization to user `end_user`:

```
GRANT SELECT ON credit_view_top TO end_user;
```

7. User `mask_owner` creates an analytic privilege for the top-level view `credit_view_top` that allows access to the row containing the credit number 1111-1111-1111-1111:

```
CREATE STRUCTURED PRIVILEGE credit_ap for SELECT ON  
data_owner.credit_view_top WHERE CREDIT_CARD = '1111-1111-1111-1111';
```

8. User `mask_owner` grants the analytic privilege `credit_ap` to user `end_user`:

```
GRANT STRUCTURED PRIVILEGE credit_ap TO end_user;
```

Results

When user `end_user` selects from the credit card view, the data is masked first and then filtered in line with the analytic privilege. This returns a empty result set.

9 SAP HANA Data Anonymization

Anonymization methods available in SAP HANA allow you to gain statistically valid insights from your data while protecting the privacy of individuals.

Why anonymize?

In a data-driven world, a growing amount of business data contains personal or sensitive information. If this data is to be used by applications for statistical analysis, it must be protected to ensure privacy. Trivial modifications to the data like replacing information that directly identifies an individual such as name or social security number (pseudonymization) or simply removing the information is not enough. Re-identification is still possible, for example if additional information is obtained (referred to as a linkage attack).

Unlike masking and pseudonymization, anonymization methods (also called privacy-enhancing methods) provide a more structured approach to modifying data for privacy protection. The quality of such anonymized or privacy-enhanced data is still sufficient for meaningful analysis. Several anonymization methods exist.

SAP HANA supports the methods **k-anonymity** and **differential privacy**. Which method provides the most appropriate level of privacy depends on your data and the potential attack scenarios and attackers.

Anonymizing Data in SAP HANA

To enable analytics on data while still protecting the privacy of individuals, data anonymization capabilities are integrated into SAP HANA calculation views.

A data controller – that is someone who determines when and how personal data is accessed and processed – defines a calculation view and configures the parameters of the chosen anonymization method to meet the required privacy level. Access to the anonymized view can then be granted to users using standard SAP HANA authorization mechanisms. A list of all calculation views that have one or more anonymization node views configured is available in the SAP HANA cockpit (for documentation purposes for example).

For more information about anonymizing data using calculation views, see the *SAP HANA Modeling Guide for XS Advanced Model*.

i Note

Data anonymization requires the script server to be running in the SAP HANA database, which is not the case by default. To add a script server to the database, execute the following SQL statement on the system database: `ALTER DATABASE <database_name> ADD 'scriptserver'`. For more information, see the section on adding a service to a tenant database in the *SAP HANA Administration Guide*.

[k-anonymity \[page 198\]](#)

k-anonymity is an intuitive and widely used method for modifying data for privacy protection. k-anonymity anonymizes data by hiding the individual record in a group of similar records, thus significantly reducing the possibility that the individual can be identified.

How can quasi-identifying fields be generalized into hierarchical groups?

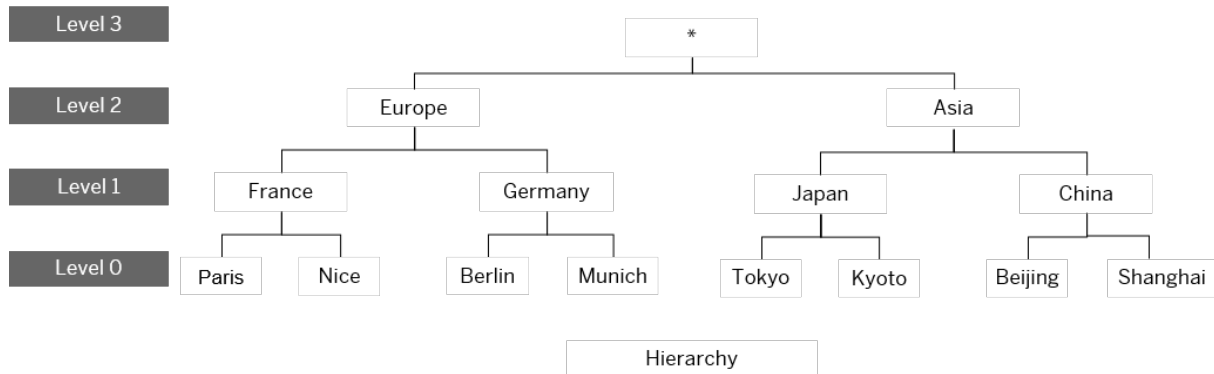
Removing identifiers is not enough to protect privacy as records can be re-identified by quasi-identifiers. Hierarchical groups are used to make quasi-identifying information in individual records less specific, thus reducing the scope for re-identification.

A hierarchy describes a generalization scheme for the information in a quasi-identifier column. The attributes in each row will be replaced by a higher-level group until each group in the data set contains a minimum of k members.

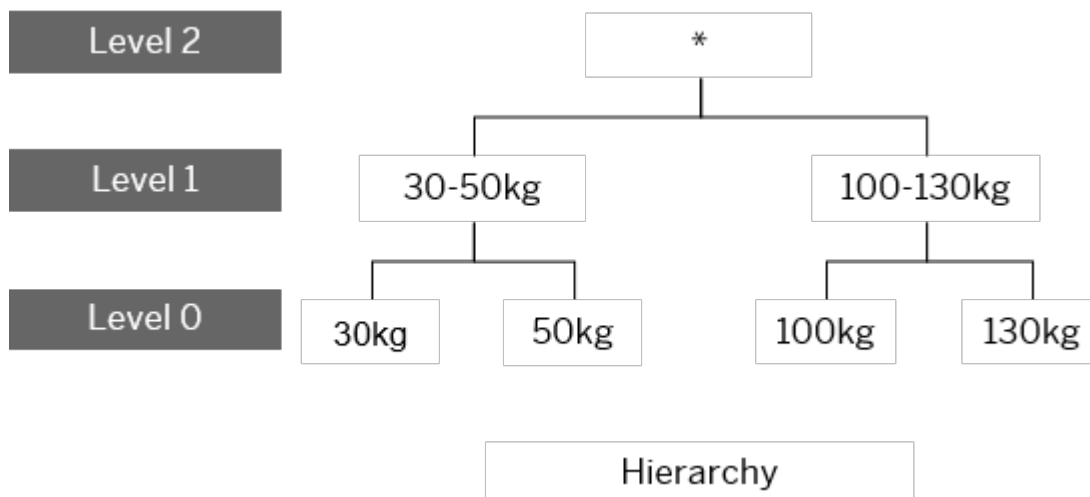
Note

A **global recoding scheme** is used. This means if that an attribute in one row is replaced by a higher-level group, the same attribute in all other rows is also replaced. For instance, in the example below, if "Paris" is replaced by "Europe", the anonymized data set will not contain "Paris" at all, even though this would not be necessary to achieve k -anonymity with the provided k value. (For more about k , see below.)

As shown in the examples below, categorical attributes can be grouped into more general categories while numerical attributes can be grouped into ranges or averages.



Hierarchy of categorical attributes

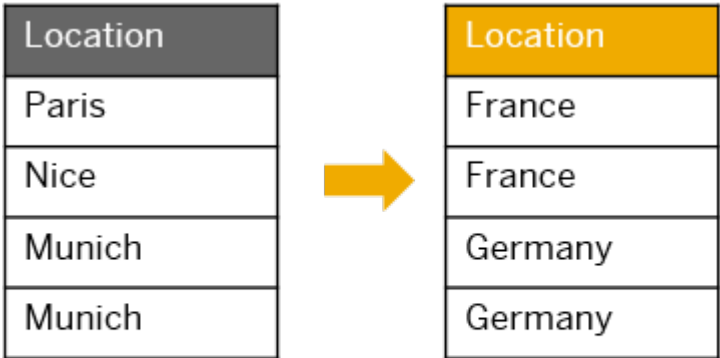


Hierarchy of numerical attributes

How many individuals have to be in a crowd for it to be considered anonymous?

This is the value k . A data set is considered k -anonymous if every individual is indistinguishable to $k - 1$ others with respect to the information in the quasi-identifying columns. In the following example, choosing $k=2$ means

at least two rows must have the exact same combination of quasi-identifying information. If this is not possible, the records are grouped into the next higher-level category.



Generalization of data if k=2

Generalization of categorical attributes along defined hierarchy

9.2 Differential Privacy

Differential privacy anonymizes data by randomizing sensitive information but in a way that regardless of whether an individual record is included in the data set or not, the outcome of statistical queries remains approximately the same. Differential privacy provides formal statistical privacy guarantees.

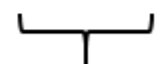
The differentially private approach to anonymizing data is typically applied to numerical data in statistical databases. It works by adding noise to the sensitive values to protect privacy, while maximizing the accuracy of queries.

To configure differential privacy, you need to consider the following questions.

What questions do you want the data to answer?

Knowing which queries will be executed on the data determines which columns to include in the data set. For example, to average salaries grouped by gender, region, start year and level, the data table to be anonymized could look like the table below. Direct identifiers and any other unrelated columns are removed.

| Gender | Region | Hire Year | Level | Salary |
|--------|--------|-----------|-------|--------|
| m | APJ | 1998 | L2 | 20000 |
| f | EMEA | 1990 | L5 | 50000 |
| m | NA | 2016 | L2 | 20000 |
| f | NA | 2005 | L3 | 30000 |


 Sensitive
data

Original data to be anonymized using differential privacy

How much of an impact can an individual have on the outcome of queries?

The aim of differential privacy is to ensure that regardless of whether an individual record is included in the data or not, a query on the data returns approximately the same result. Therefore, we need to know what the maximum impact of an individual record could be. This will be determined by the highest possible value and the lowest possible value in the data set and is referred to as the sensitivity of the data. The higher the sensitivity, the more noise needs to be applied.

In the above example, the column containing salary information needs to be protected. The maximum impact of an individual value would be the maximum possible salary minus the minimum possible salary. If we know that the maximum possible salary is 80,000 EUR, then the sensitivity value is also 80,000 (maximum salary minus minimum salary, which is 0).


For more technical information about sensitivity, see the section below.

What is the acceptable probability that the outcome of queries changes before it is considered a privacy breach?

This is the value epsilon (ϵ). Typical values are 0.1 or 0.01. However, for some use cases, setting epsilon to a value larger than 1, for example 5, is fine as well. e^ϵ is the maximum multiplicative impact on the probability of any outcome. The lower the value of epsilon, the greater the privacy required and the more noise is applied.

For more technical information about epsilon, see the section *Differential Privacy – Additional Technical Information* below.

| Gender | Region | Hire Year | Level | Salary |
|--------|--------|-----------|-------|---------------|
| m | APJ | 1998 | L2 | 20000 + X_1 |
| f | EMEA | 1990 | L5 | 50000 + X_2 |
| m | NA | 2016 | L2 | 20000 + X_3 |
| f | NA | 2005 | L3 | 30000 + X_4 |


 Noised data

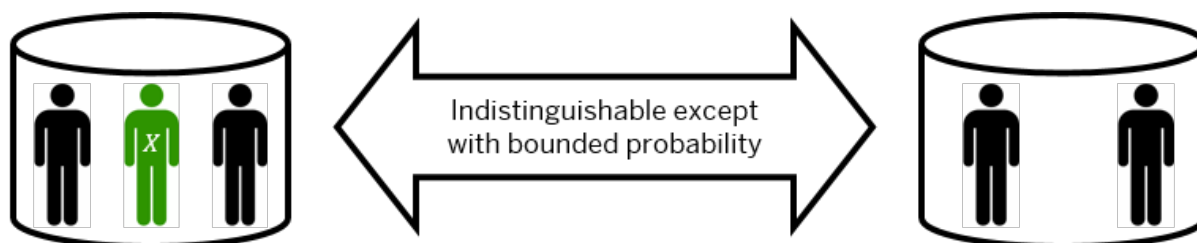
Data anonymized using differential privacy

9.2.1 Differential Privacy – Additional Technical Information

The following section provides further technical explanation of differential privacy. This will help you understand how to optimize the utility of anonymized data.

In general, differential privacy is a definition, not a method. If a data set is differentially private, the inclusion or the exclusion of a specific person will only lead to a limited multiplicative impact (e^ϵ) on the probability of each query result on that data set

In the figure below $San(DB)$ represents the differentially private representation of the data set and X the included/excluded person.



$$\frac{\Pr[San(DB \cup \{X\}) = y]}{\Pr[San(DB \setminus \{X\}) = y]} \leq e^\epsilon \approx 1 \pm \epsilon$$

Differential Privacy Definition

Differential privacy can be achieved in many ways. SAP HANA uses the Laplace mechanism. It draws noise from a Laplace distribution such that the multiplicative guarantee holds, and requires the definition of the sensitivity, that is the maximum impact an individual can have on the data set with respect to the query results. This depends on the application. For instance, if you want to publish salary data, the minimum possible value one could think of is 0, and the maximum value is the highest possible salary in the company. For other scenarios, defining sensitivity is much easier. For example, consider a data set in which individuals answer a survey in a given range of values.

Choosing the correct sensitivity is necessary to guarantee differential privacy. Setting the sensitivity higher than necessary will reduce the quality of the anonymized data.

Utility Tuning

ϵ and sensitivity impact the utility of a data set after anonymization. Since ϵ defines the privacy guarantee and is motivated by external privacy requirements (for example, regulations), it should usually not be changed. However, sensitivity depends on the data set and can be influenced.

Assume you want to publish a data set containing salaries. Usually, there is a broad range of average salaries and a very low number of very high salaries in the data set. The sensitivity is at least the difference between 0 and the maximum salary (in the data set). However, if you guarantee that no salary larger than a certain upper limit will be in the data set (by filtering the incoming data), you can set the sensitivity much lower. In this way, by filtering and pre-processing the data, you can keep the sensitivity at a certain level and therefore increase the utility of the anonymized data.

10 Data Storage Security in SAP HANA

Several mechanisms can be used to protect security-relevant data used by the SAP HANA database from unauthorized access.

[Data Security in the File System \[page 204\]](#)

Data in the SAP HANA database (including configuration data) is stored in the file system of the operating system and protected by operating system permissions.

[Server-Side Data Encryption Services \[page 205\]](#)

SAP HANA features encryption services for encrypting data at rest, as well as an internal encryption service available to applications with data encryption requirements. SAP HANA uses the secure store in the file system functionality to protect all encryption root keys. All passwords on the SAP HANA database server are stored securely.

[Client-Side Data Security \[page 222\]](#)

The client user store and client-side data encryption allow SAP HANA clients to protect security-relevant data from unauthorized access.

[Cryptographic Service Provider \[page 238\]](#)

All encryption services used in SAP HANA require the availability of a cryptographic service provider on the SAP HANA server and the SAP HANA client.

10.1 Data Security in the File System

Data in the SAP HANA database (including configuration data) is stored in the file system of the operating system and protected by operating system permissions.

You configure the data path during installation. For more information about the recommended file system layout, see the *SAP HANA Server Installation and Update Guide*. The file permissions of the operating system are strictly configured. Therefore, do not change them after installation.

For more information see SAP Note 1730999 and SAP Note 1731000.

Hardware-Based Encryption with Persistent Memory

Persistent memory, also referred to as non-volatile RAM (NVRAM) or storage class memory, is supported in SAP HANA as a persistent storage type. If you are using persistent memory, you can use hardware-based encryption to protect your data on disk. Please contact your hardware vendor for the latest information about availability of the hardware and support for this feature.

For more information about persistent memory, see the *SAP HANA Administration Guide*.

Related Information

[SAP Note 1730999](#)

[SAP Note 1731000](#)

10.2 Server-Side Data Encryption Services

SAP HANA features encryption services for encrypting data at rest, as well as an internal encryption service available to applications with data encryption requirements. SAP HANA uses the secure store in the file system functionality to protect all encryption root keys. All passwords on the SAP HANA database server are stored securely.

- [Passwords \[page 205\]](#)
- [Data-at-Rest Encryption \[page 205\]](#)
- [Security-Relevant Application Data \[page 206\]](#)
- [Secure Store in the File System \(SSFS\) \[page 206\]](#)
- [Encryption Services and Keys \[page 206\]](#)

Passwords

On the SAP HANA database server, all passwords are stored securely:

- Operating system user passwords are protected by the standard operating system mechanism, `/etc/passwd` file.
- All database user passwords are stored in hashed salted form using PBKDF2 (Password-Based Key Derivation Function 2) and, for downward compatibility, secure hash algorithm SHA-256. The SAP HANA implementation of PBKDF2 uses the SHA-256 secure hash algorithm and 15,000 iterations.
- Credentials required by SAP HANA applications for outbound connections are securely stored in a database-internal credential store. This internal credential store is in turn secured using the internal application encryption service. For example, in an SAP HANA smart data access scenario, credentials required to access a remote source are protected using the internal application encryption service.

Data-at-Rest Encryption

To protect data saved to disk from unauthorized access at operating system level, the SAP HANA database supports data encryption in the persistence layer for the following types of data:

- Data in data volumes
- Redo logs in log volumes

Data and log backups can also be encrypted.

Security-Relevant Application Data

An internal encryption service is used to encrypt sensitive application data. This includes credentials required by SAP HANA for outbound connections, private keys of the SAP HANA server stored in the database, and data in secure stores defined by developers of SAP HANA XS applications.

Secure Store in the File System (SSFS)

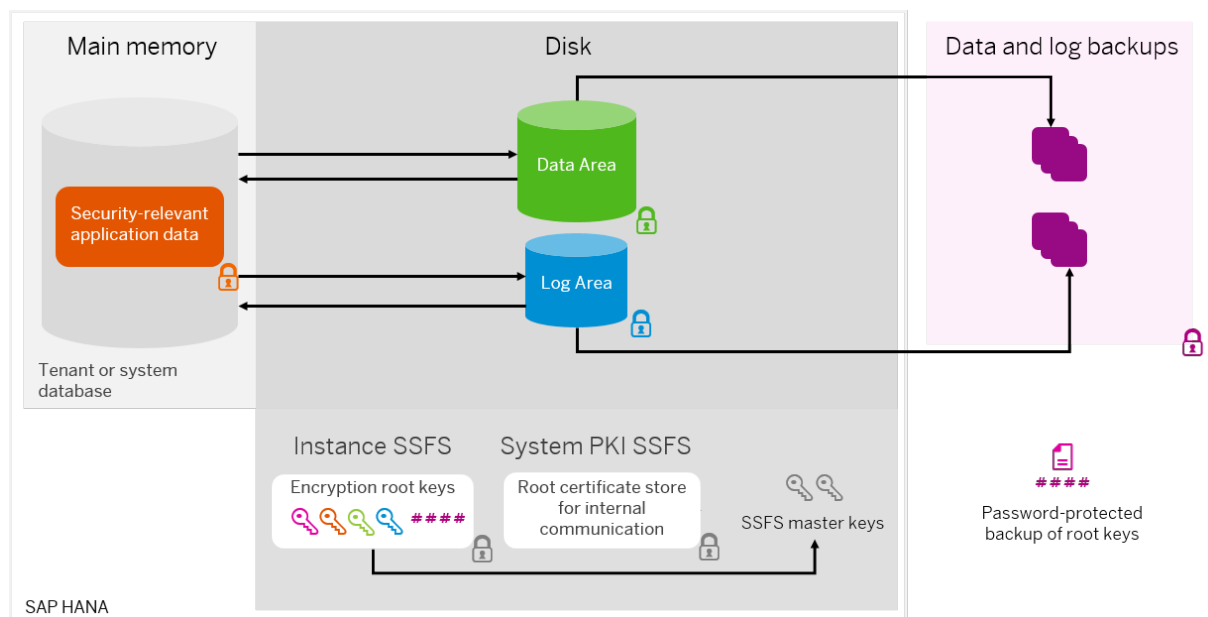
SAP HANA uses two secure stores in the file system: the **instance SSFS** and the **system PKI SSFS**. The instance SSFS protects the root keys used for all data-at-rest encryption services and the internal application encryption service. The system PKI SSFS protects system-internal root certificates required for secure internal communication.

Encryption Services and Keys

The following diagram provides an overview of which data in SAP HANA can be encrypted using a dedicated encryption service, and how all associated encryption root keys are stored in the secure store in the file system of the SAP HANA instance (instance SSFS).

Note

The following diagram shows only one database. However, a system always has a system database and any number of tenant databases. Every database in the system has its own encryption root keys for each of the available encryption services. The root keys of all databases are stored in the instance SSFS.



Encryption Services and Keys

Related Information

- [Encryption Key Management \[page 207\]](#)
- [Data and Log Volume Encryption \[page 209\]](#)
- [Backup Encryption \[page 212\]](#)
- [Internal Application Encryption Service \[page 216\]](#)
- [Root Key Backup \[page 219\]](#)
- [Secure Stores in the File System \(SSFS\) \[page 220\]](#)

10.2.1 Encryption Key Management

SAP HANA generates unique keys on installation and database creation. However, if you received SAP HANA from a hardware or hosting vendor, you might want to change them to ensure they are not known outside your organization. We recommend that you do this immediately after handover. All keys can be changed using the SAP HANA cockpit.

The following master and root keys exist and can be changed:

- Instance SSFS master key
- System PKI SSFS master key
- Data volume encryption root key
- Redo log encryption root key
- Backup encryption root key
- Application encryption service root key

i Note

Unlike the root keys for the internal application encryption service and data volume encryption, the root keys for redo log encryption and backup encryption do not encrypt any other encryption keys.

Reinstalling your system will change all master and root keys. You can also change keys manually and individually.

The following sections explain how and when you can safely change master and root keys. More detailed instructions, as well as information on the overall configuration workflow for server-side encryption are available in the *SAP HANA Administration Guide*.

How and When to Change the SSFS Master Keys

How to Change

You change the SSFS master keys using the command line tool `rsecssfx`. This requires operating system access (<sid>adm user).

When to Change

Unique master keys are generated during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change them immediately after

handover to ensure that they are not known outside of your organization. You can also change the master keys any time later.

i Note

In a system-replication configuration, you change the instance SSFS master key on the primary system. To trigger replication of the new key to the secondary system, you must subsequently restart the secondary system. In multi-tier system replication scenarios involving three systems, restart the tier-2 secondary system first, then the tier-3 secondary system. If a secondary system takes over from its replication source before the new master key has been replicated, all systems registered will use the old key from the former secondary system instead.

Changing the Encryption Root Keys

Encryption root keys are stored in the instance SSFS. Before you change these root keys, it is important to understand how the key change process impacts the recoverability of the database. In some cases, a backup of encryption root keys must be available.

| | |
|--|--|
| Data volume encryption root key | Changing the data volume encryption root key does not require a root key backup as a data backup will always contain all the required information to recover that backup successfully. For snapshot-based backups, the data volume encryption root key is part of the data backup itself. For backups other than snapshot-based backups, the data volume encryption root key is not required to recover the system at all. |
| Redo log encryption root key | SAP HANA does not log the key change operation of the redo log encryption root key in the SAP HANA redo log. This means that the redo log encryption root key must be backed up after a new one is created, but before it is actually used to encrypt data. In this way, it is ensured that even in the case of a disaster recovery, all encrypted data can be recovered successfully. |
| Backup encryption root key | SAP HANA does not log the key change operation of the backup encryption root key in the SAP HANA redo log. This means that the backup encryption root key must be backed up after a new one is created, but before it is actually used to encrypt data. In this way, it is ensured that even in the case of a disaster recovery, all encrypted data can be recovered successfully. |
| Application encryption service root key | SAP HANA does not log the key change operation of the application service encryption root key in the SAP HANA redo log. This means that the application service encryption root key must be backed up after a new one is created, but before it is actually used to encrypt data |

The recommended approach is to always backup all root keys.

How to Change Encryption Root Keys

Use the SAP HANA cockpit to change the data volume encryption root key, redo log encryption root key, the backup encryption root key, and the internal application encryption service root key. The key change process is as follows:

1. Generate new keys.

2. Back up new keys.
3. Activate new keys.
4. Back up activated keys.

→ Recommendation

SAP recommends that you always adhere to this process for all key types. Establishing this process for all root key changes in your organization will ensure that a backup of all root keys is created even when an administrator changes a root key that does not require a backup. Using the SAP HANA cockpit will help you to adhere to the recommended process.

For more information, see the *SAP HANA Administration Guide*.

When to Change Encryption Root Keys

Unique root keys are generated during installation or database creation. However, if you received SAP HANA from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization.

You can also change them any time later.

i Note

In a system-replication configuration, change root keys in the primary system only. New keys will be propagated to all secondary systems. The secondary systems must be running and replicating.

Related Information

[SAP Note 2097613](#)

10.2.2 Data and Log Volume Encryption

To protect data saved to disk from unauthorized access at operating system level, the SAP HANA database supports data encryption in the persistence layer. Data volume encryption protects the data area on disk, while redo log encryption protects the log area on disk.

The SAP HANA database holds the bulk of its data in memory for maximum performance, but it still uses persistent disk storage to provide a fallback in case of failure. During normal operation, data is automatically saved from memory to disk at regular savepoints. Additionally, all data changes are captured in redo log entries. A redo log entry is written to disk with each committed database transaction. After a power failure, SAP HANA can be restarted like any disk-based database and returns to its last consistent state by replaying the redo log entries since the last savepoint.

Both data and redo log entries can be encrypted:

- **Data volume encryption**
If data volumes are encrypted, all pages that reside in the data area on disk are encrypted using the AES-256-CBC algorithm. Pages are transparently decrypted as part of the load process into memory.

When pages reside in memory they are therefore not encrypted and there is no performance overhead for in-memory page accesses. When changes to data are persisted to disk, the relevant pages are automatically encrypted as part of the write operation.

Pages are encrypted and decrypted using 256-bit page encryption keys. Page keys are valid for a certain range of savepoints and can be changed by executing SQL statements. After data volume encryption has been enabled, an initial page key is automatically generated. Page keys are never readable in plain text, but are encrypted themselves using a dedicated data volume encryption root key, which is generated randomly during installation.

- **Redo log encryption**

If redo logs are encrypted, log entries are encrypted using the AES-256-CBC algorithm before they are written to disk. Log entries are encrypted and decrypted using a 256-bit long root key, which is generated randomly during installation.

During start-up, administrator interaction is not required. The data volume encryption and redo log root keys are stored using the secure storage in the file system functionality of the instance (instance SSFS) and are automatically retrieved from there.

i Note

SAP HANA uses the instance SSFS to protect the encryption root keys that are used to protect encryption keys or persistent data in the SAP HANA system from unauthorized access. All root keys are encrypted using the SSFS master key.

→ Recommendation

Although SAP HANA provides you with the flexibility to encrypt data volumes, redo logs, and backups independently of each other, if you require full protection in the persistence layer, we recommend that you enable all services.

Enabling encryption does not increase data size.

Data Not Encrypted

- Database backups

The data volume encryption feature does **not** encrypt the contents of data and log backups. To encrypt data and log backups, backup encryption must be enabled.

If backup encryption is **not** enabled, only data that has been encrypted internally in the database (that is, independently of the data volume encryption feature) is encrypted in backups. For example, data stored in the secure internal credential store is encrypted in backups

i Note

Unlike data backups, data in storage snapshots **is** encrypted as part of data volume encryption. This is because a storage snapshot captures the content of the data area exactly as it is at a particular point in time.

i Note

To ensure that all data restored during the data and log recovery phases is encrypted, data volume encryption must be enabled before the recovery is started.

- Database traces
The data volume encryption feature does **not** encrypt database traces.
For security reasons, we recommend that you do not run the database with extended tracing for more than short-term analysis since tracing might expose security-relevant data that would be encrypted in the persistence layer, but not in the trace. Therefore, you should not keep such trace files on disk beyond the respective analysis task.

Enabling Data and Log Volume Encryption in Tenant Databases

Enabling Data and Log Encryption in a New Database

Ideally, encryption is enabled in the database immediately on creation. You can ensure that this is the case with the following parameters in the `database_initial_encryption` section of the `global.ini` configuration file in the system database:

- `persistence_encryption`
- `log_encryption`

To ensure that any subsequently created tenant databases automatically have encryption enabled, set the value of these parameters to **on**. If a particular tenant database does not require encryption, it can be switched off. By default these parameters are set to **off**.

Who can enable or disable data and log volume encryption (as well as backup encryption) in the tenant database initially depends on how the parameter `[database_initial_encryption] encryption_config_control` is configured:

- If the value of this parameter is `local_database` (default), then only the tenant database administrator can enable or disable encryption from the tenant database using the ALTER SYSTEM statement.
- If it is `system_database`, then only the system database administrator can enable or disable encryption from the system database using the ALTER DATABASE statement.

i Note

If the tenant database administrator has control over encryption configuration and later wants to hand over this control to the system administrator, the tenant database administrator must execute the statement `ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY SYSTEM DATABASE`. If the system administrator has control and wants to hand it over to the tenant database administrator, the system administrator must execute the statement `ALTER DATABASE <database_name> ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL DATABASE`. For simplicity, the system administrator can hand over control to all tenants instead of one by one by executing statement `ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL DATABASES`.

i Note

In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Enabling Data and Log Encryption in a Running Database

It is possible to enable data volume encryption in a tenant database that already exists and is already in operation. However, only the pages in use within the data volumes are initially encrypted. Pages in the data

volumes that are not in use may still contain old content and will only be overwritten and encrypted over time. Furthermore, if you enable redo log encryption after the database has been in operation, only future redo log entries are encrypted. Existing redo log entries remain unencrypted until they are overwritten. Log segment files are cyclically overwritten once they have been backed up. This means that data in data and log volumes will only be fully encrypted after some delay.

To attain complete protection for an existing tenant database, the overall process is therefore:

1. Back up the tenant database.
2. Drop the tenant database.
All volumes are removed.
3. Create the tenant database again.

i Note

In a system-replication configuration, all secondary systems must be now be running and replicating before the next steps can be performed.

4. Set the password for the root key backup.

i Note

In a system-replication configuration, set the root key backup password in the primary system only. The password will be propagated to all secondary systems. The secondary systems must be running and replicating.

5. Back up root keys.
6. Enable data volume encryption and redo log encryption (if not already enabled).
Who can initially enable encryption in the tenant database depends on how the parameter `[database_initial_encryption] encryption_config_control` is configured as described above.

i Note

In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

7. Perform a data recovery.

For more information about encryption configuration, including recommended key change management, see the *SAP HANA Administration Guide*.

Related Information

[Backup Encryption \[page 212\]](#)

10.2.3 Backup Encryption

SAP HANA supports native backup encryption.

Backup encryption safeguards the privacy of the SAP HANA business data by preventing unauthorized parties from reading the content of backups.

i Note

As an alternative to native SAP HANA encryption, many third-party backup tools and storage tools offer support for backup encryption. If you are using a third-party backup tool, consult your tool vendor for more information.

Which Backup Types Can Be Encrypted?

Backup encryption can be enabled for all backup types.

For more information, see *Enable and Disable Encryption of Data and Log Backups* in *SAP HANA Administration Guide (Encryption)*.

i Note

To encrypt data snapshots, additional steps are necessary.

For more information, see *Encryption of Data Snapshots* in *SAP HANA Administration Guide (SAP HANA Database Backup and Recovery)*.

Considerations for Backup Encryption

- It takes longer to create encrypted backups than unencrypted backups.
- It takes longer to recover a database using encrypted backups than from unencrypted backups.
- If backup encryption is enabled, both data backups and log backups are encrypted.

i Note

If you enable encryption (either for backup, log, or data volume) in the system database immediately after installation of SAP HANA, encryption is automatically enabled for any subsequently created tenant databases. If a particular tenant database does not require encryption, the administrator for that tenant database can disable encryption for it.

For more information, see *Enable Data and Log Volume Encryption in a New SAP HANA Database* in *SAP HANA Administration Guide (Encryption)*.

- The same backup encryption root key is used for both data backups and log backups.

i Note

Data snapshots are not encrypted using the backup encryption root key.

For more information, see *Encryption of Data Snapshots* in *SAP HANA Administration Guide (SAP HANA Database Backup and Recovery)*.

- It is currently not possible to enable encryption for an individual data backup.
- The size of encrypted SAP HANA backups is the same as unencrypted backups (except for the checksum).
- The backup catalog is not encrypted.
The backup catalog shows which backup encryption root keys were used to encrypt the backups.

Enabling Backup Encryption in Tenant Databases

You can enable backup encryption at any time. By default, backup encryption is not enabled in a new tenant database. Backup encryption is controlled by the parameter `backup_encryption` in the `database_initial_encryption` section of the `global.ini` configuration file of the system database.

If you want to ensure that new tenant databases automatically have encryption enabled, set the value of this parameter to **on**. If a particular tenant database subsequently does not require encryption, it can be switched off. By default these parameters are set to **off**.

Who can enable or disable backup encryption (as well as data and log volume encryption) in the tenant database initially depends on how the parameter `[database_initial_encryption] encryption_config_control` is configured:

- If the value of this parameter is `local_database` (default), then only the tenant database administrator can enable or disable encryption from the tenant database using the ALTER SYSTEM statement.
- If it is `system_database`, then only the system database administrator can enable or disable encryption from the system database using the ALTER DATABASE statement.

i Note

If the tenant database administrator has control over encryption configuration and later wants to hand over this control to the system administrator, the tenant database administrator must execute the statement `ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY SYSTEM DATABASE`. If the system administrator has control and wants to hand it over to the tenant database administrator, the system administrator must execute the statement `ALTER DATABASE <database_name> ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL DATABASE`. For simplicity, the system administrator can hand over control to all tenants instead of one by one by executing statement `ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL DATABASES`.

i Note

In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Backup Encryption Root Keys

- If you enable encryption (either for backup, log, or data volume) in the SAP HANA backups are encrypted and decrypted using backup encryption root keys.
The backup encryption root keys are encrypted and stored in the secure store in the file system (instance SSFS) together with other encryption root keys. For example, application encryption root keys.

- A new backup encryption root key is generated for every tenant database when the tenant database is created.
- If backup encryption is enabled, a database administrator must ensure that the backup encryption root keys are backed up.

⚠ Caution

Whenever the backup encryption root keys are changed, you must back them up. Without a current backup of the backup encryption root keys, some data will be lost after a recovery.

For more information, see *Root Key Backup* in the *SAP HANA Security Guide*.

i Note

The block-level integrity of encrypted backups can still be checked without access to the backup encryption root keys.

For more information, see *Manually Checking Whether a Recovery is Possible*.

For more information about working with encryption root keys, see *Encryption Key Management* in the *SAP HANA Security Guide* and *Changing Encryption Root Keys* in the *SAP HANA Administration Guide (Encryption)*.

Backing Up and Recovering Backup Encryption Root Keys

The backup encryption root keys are stored in the instance SSFS.

The instance SSFS has to be backed up and recovered independently of the SAP HANA database.

To recover SAP HANA from encrypted backups, you need to first recover (import) the backup encryption root keys into the instance SSFS.

For more information, see *Import Backed-up Root Keys* in the *SAP HANA Administration Guide (Encryption)*.

i Note

Ensure that the backup encryption root keys are recovered to the database with the correct database ID. For a recovery or a database copy, the database ID may be different.

For more information about working with backup encryption root keys, see *Encryption Key Management* in the *SAP HANA Security Guide* and *Changing Encryption Root Keys* in the *SAP HANA Administration Guide (Encryption)*.

Change the Backup Encryption Root Key

Depending on your security policy, it may be necessary to change the backup encryption root key at regular intervals.

After a new backup encryption root key is created, **but before it is activated**, the backup encryption root keys must be backed up. If the backup encryption root keys were changed, several backup encryption root keys may be needed to decrypt backups.

For more information, see the *ALTER SYSTEM BACKUP ENCRYPTION Statement* in the *SQL Reference Guide*.

i Note

At any one time, only one backup encryption root key can be active.

Related Information

SAP HANA Security Guide

[Data Storage Security in SAP HANA \[page 204\]](#)

[Secure Stores in the File System \(SSFS\) \[page 220\]](#)

[Root Key Backup \[page 219\]](#)

[Data and Log Volume Encryption \[page 209\]](#)

[Encryption Key Management \[page 207\]](#)

[SAP HANA Administration Guide \(Encryption\)](#)

[SAP HANA Administration Guide \(SAP HANA Database Backup and Recovery\)](#)

[SQL Reference Guide](#)

10.2.4 Internal Application Encryption Service

The internal encryption service is used internally by applications requiring data encryption.

i Note

In the SAP HANA 1.0 documentation, the internal application encryption service was referred to as the internal data encryption service.

The internal application encryption service is used in the following contexts:

- **Secure internal credential store**
This service stores credentials required by SAP HANA for outbound connections. It is used, for example, when data is retrieved from remote data sources using SAP HANA smart data access. It is also used during HTTP destination calls from SAP HANA XS classic applications.
For more information, see the section on the secure internal credential store.
- **Application data requiring encryption**
Application developers can maintain values in the SAP HANA secure store for SAP HANA XS advanced applications or define secure stores using the SAP HANA XS classic `$.security.store` API.
For more information, see:
 - SAP HANA XS advanced: *Maintain Values in the SAP HANA Secure Store* in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*
 - SAP HANA XS classic: *Using the Server-Side JavaScript APIs* in the *SAP HANA Developer Guide (For SAP HANA Studio)* and *Class:Store* in the *SAP HANA XS JavaScript API Reference*.
- **Private key store**
This service stores the private keys of the SAP HANA server required for secure client-server communication, if the relevant personal security environment (PSE) is stored in the database. PSEs stored in the database are called certificate collections.

For more information, see the section on SSL configuration on the SAP HANA server and certificate management in SAP HANA.

Every consumer of the service has its own system-internal application encryption key. These keys are generated as follows:

- The application key for the internal credential store is generated randomly during the first startup.
- Application keys for XS secure stores are created at the same time as the XS secure store.
- The application key for the private key store is created when the first private key is set for a certificate collection.

Application encryption keys are encrypted with the application encryption service root key.

SAP HANA generates unique root keys on installation or database creation. However, if you received SAP HANA from a hardware or hosting partner, you might want to change the root key of the internal application encryption service to ensure it is not known outside your organization. We recommend that you do this immediately after system installation or handover from your hardware or hosting partner.

i Note

In a system-replication configuration, change root keys in the primary system only. New keys will be propagated to all secondary systems. The secondary systems must be running and replicating.

The system database and all tenant database have their own individual application encryption service root key.

Related Information

[Secure Internal Credential Store \[page 217\]](#)

[Certificate Management in SAP HANA \[page 261\]](#)

[Class: Store](#)

10.2.4.1 Secure Internal Credential Store

A database-internal credential store is available that allows you to securely store in the SAP HANA database the credentials required by SAP HANA applications for outbound connections. For example, in an SAP HANA smart data access scenario, credentials required to access a remote source are protected using the internal application encryption service.

Credentials can be created and updated by users and privileged administrators using the SQL interface. However, access to credentials in unencrypted form is only available to native SAP HANA applications via an internal API.

Users can create and modify their own credentials. A user with the system privilege CREDENTIAL ADMIN can manage credentials for other users. Credentials are also created implicitly during the creation of remote data sources (SAP HANA smart data access scenario) and HTTP destinations for SAP HANA XS classic applications.

Credentials are created using the SQL statement CREATE CREDENTIAL as follows.

```
CREATE CREDENTIAL FOR USER <user_name> COMPONENT '<application>' PURPOSE
'<credential_purpose>' TYPE '<credential_type>' USING '<credential>'
```

A credential consists of the following elements:

| Element | Description |
|-----------|---|
| User | <p>The database user for which the credential is stored</p> <p>If no user name is specified, the supplied credential serves as a general entry that can be used by the application if no explicit mapping for a database user is possible. For example, in an SAP HANA smart data access scenario, the connection to a data source may always be established using the same technical user.</p> |
| Component | <p>The application for which the credential is stored</p> <p>The value of the 'component' element is defined by the application, for example, in an SAP HANA smart data access scenario, the component is 'SAPHANAFEDERATION'.</p> |
| Purpose | <p>The purpose for which the application is storing this credential</p> <p>The value of the 'purpose' element is defined by the application, for example, in an SAP HANA smart data access scenario, the purpose is the name of the remote data source.</p> |
| Type | <p>The type of credential being stored, for example PASSWORD or X509</p> <p>The supported values for the this element are specific to the application.</p> |
| Using | <p>The actual credential, for example user name and password for a credential of type PASSWORD</p> |

i Note

You can only set credentials using SQL. It not possible to view them. The unencrypted value of the credential is only available to the application via an internal interface.

Example

```
CREATE CREDENTIAL FOR USER TESTUSER COMPONENT 'SAPHANAFEDERATION' PURPOSE 'ASE'
TYPE 'PASSWORD' USING 'user="remotedbuser";password="abc123"'
```

Credentials can be changed and dropped using the ALTER CREDENTIAL and DROP CREDENTIAL statements respectively.

The system view CREDENTIALS contains information about stored credentials.

i Note

Credentials stored using the credential store remain encrypted even in backups. To allow for the reconstruction of credential data in the case of database recovery, the encryption key used is also part of the backup. To avoid unauthorized access to the encrypted credentials, backups should be stored in a safe and secure place.

i Note

The credential store uses the internal application encryption service of the SAP HANA database. The encryption root key of the application encryption service is stored in the instance secure store in the file system (SSFS) along with the encryption root key used for data volume encryption (if activated). During a recovery, the encryption root key of the application encryption service is restored to the target system's instance SSFS without interfering with the encryption root key of data volume encryption of the target system.

Related Information

[Server-Side Data Encryption Services \[page 205\]](#)

[Encryption Key Management \[page 207\]](#)

10.2.5 Root Key Backup

A backup of encryption root keys must be available at an external location to ensure recovery is possible in certain scenarios.

Encryption root keys are stored in the instance secure storage in the file system (SSFS) and can be retrieved from there automatically during operation.

However, a backup of encryption root keys must be stored in a root key backup file (*.rkb) at an external location accessible to an SAP HANA administrator. The instance SSFS must always be restored from this backup before a database recovery, unless:

- You have never changed any of the encryption root keys.
- You are performing a recovery into the same database from which the backup was taken, and the database's SSFS is intact and contains the latest root key changes.

Root keys must be backed up every time they are changed, that is every time new keys are generated or activated. The root key backup is secured using a single password, which must be set before a backup is created. The password is securely stored and all subsequent root key backups taken are protected by this password.

For more information about setting the root key backup password and validating the password in the instance SSFS, as well as changing encryption root keys, see the *SAP HANA Administration Guide*.

⚠ Caution

Store both the root key backup and the password required to read it in a secure location. Losing the backup or the password may result in the database being unrecoverable.

i Note

In a system-replication configuration, set the root key backup password in the primary system only. The password will be propagated to all secondary systems. The secondary systems must be running and replicating.

In a disaster-recovery situation, the root keys must first be extracted from the backup into the database before the recovery is started. For more information about how to do this, see the *SAP HANA Administration Guide*.

10.2.6 Secure Stores in the File System (SSFS)

SAP HANA uses two secure stores in the file system: the **instance SSFS** and the **system PKI SSFS**. The instance SSFS protects the root keys used for all data-at-rest encryption services and the internal application encryption service. The system PKI SSFS protects system-internal root certificates required for secure internal communication.

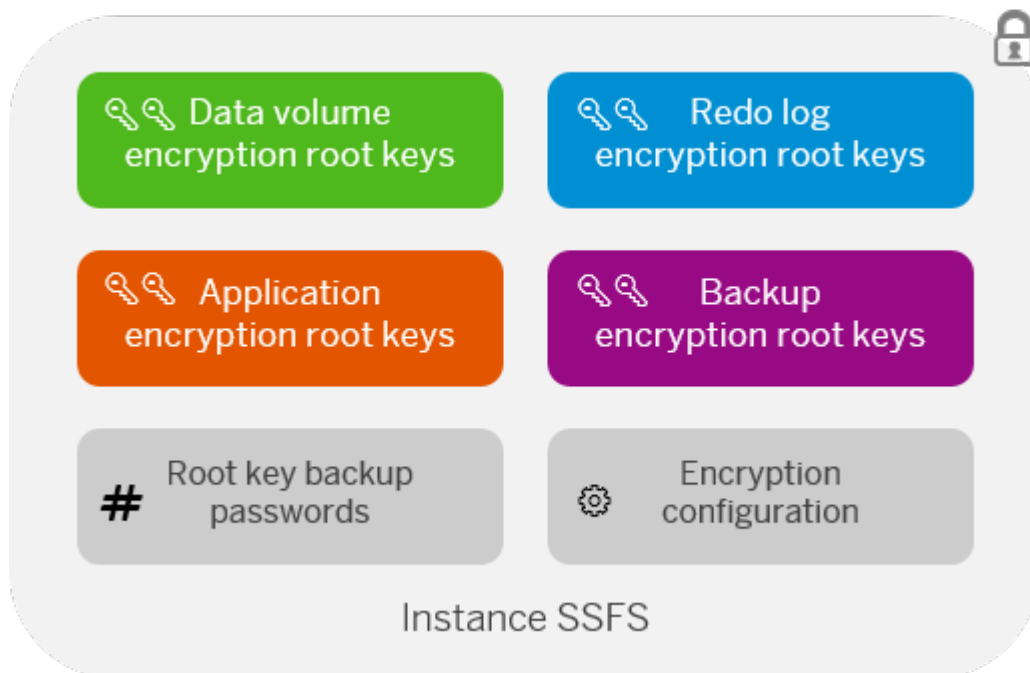
Instance SSFS

SAP HANA uses the instance SSFS to protect the following encryption root keys:

- The root keys used for:
 - Data volume encryption
 - Redo log encryption
 - Data and log backup encryption
 - Internal application encryption service of the database
- The password of the root key backup
- Encryption configuration information

These root keys protect all encryption keys (and data) used in the SAP HANA database from unauthorized access.

The system database and all tenant databases have their own encryption root keys.



Instance SSFS Contents

To prevent data encrypted in the SAP HANA database from becoming inaccessible, the content of the instance SSFS and key information in the database must remain consistent. The database detects if this is not the case, for example if the instance SSFS becomes corrupted, and issues an alert (check 57). It is recommended that you contact SAP Support to resolve the issue.

System PKI SSFS

The system PKI SSFS protects the X.509 certificate infrastructure that is used to secure internal SSL/TLS-based communication between hosts in a multiple-host system or between processes of the individual databases in a system.

The master key of the system PKI SSFS encrypts all system-internal root certificates required for secure internal communication.

Master Key Change

The contents of both the instance SSFS and the system PKI SSFS are protected by individual master key files in the file system.

Unique master keys are generated during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization. You can also change the master keys any time later.

i Note

The default path of the key file of the instance SSFS is `/usr/sap/<sid>/SYS/global/hdb/security/ssfs`. If you change the default path, you may need to reconfigure it in the event of a system rename.

For more information, see the *SAP HANA Administration Guide*.

Related Information

[Secure Internal Communication \[page 60\]](#)

10.3 Client-Side Data Security

The client user store and client-side data encryption allow SAP HANA clients to protect security-relevant data from unauthorized access.

Secure User Store

The user store of the SAP HANA client (`hdbuserstore`) can be used to store user logon information for connecting to an SAP HANA system. This allows client applications to connect to the database without having to enter a user's password explicitly. It is typically used by scripts connecting to SAP HANA. For more information, see the section on the secure user store.

Client-side Data Encryption

To protect sensitive data from unauthorized access on the server, column data can be encrypted by the client. Encryption keys accessible only to the client are used to encrypt and decrypt data directly on the client. For more information, see the section on client-side data encryption.

Information on Specific Client Applications

SAP HANA Cockpit

See the section on security aspects of the SAP HANA cockpit.

SAP Web IDE for SAP HANA

See the section on security aspects of the SAP Web IDE for SAP HANA.

SAP HANA Studio

- For users using the SAP HANA studio to connect to an SAP HANA system, the Eclipse secure storage can be used to store passwords. If this is not desired, the feature can be disabled for the SAP HANA studio. For more information, see *Disable Password Storage in Eclipse Secure Store* in the *SAP HANA Administration Guide*. The Eclipse secure storage can be used to store user passwords in the SAP HANA studio.
- Data copied to local workspaces requires additional protection. For more information, see *Protection of Data in SAP HANA Studio Workspaces*.

Microsoft Excel

Microsoft Excel is an end-user client for SAP HANA. In Microsoft Excel, you can connect to an SAP HANA system as an external data source, and then create a PivotTable to analyze that data. Connections to SAP HANA use the SAP HANA ODBO driver, which is installed with the SAP HANA client. When you are creating a connection to an SAP HANA system, you must specify a database user and password in the connection wizard.

⚠ Caution

Although you can choose to save the password in the connection file, we recommend that you do **not** since the saved password is not encrypted.

Related Information

[Client-Side Data Encryption \[page 230\]](#)

[Secure User Store \(hdbuserstore\) \[page 223\]](#)

[Security Aspects of SAP HANA Cockpit \[page 282\]](#)

[Security Aspects of SAP Web IDE for SAP HANA \[page 348\]](#)

[Protection of Data in SAP HANA Studio Workspaces \[page 237\]](#)

10.3.1 Secure User Store (hdbuserstore)

The secure user store (`hdbuserstore`) is a tool installed with the SAP HANA client. Use it to store connection information to SAP HANA systems securely on the client so that client applications can connect to SAP HANA without users having to enter this information. It is typically used by scripts connecting to SAP HANA.

The secure user store allows you to store SAP HANA connection information, including user passwords, securely on clients. In this way, client applications can connect to SAP HANA without the user having to enter host name or logon credentials. You can also use the secure store to configure failover support for application servers in a 3-tier scenario (for example, SAP Business Warehouse) by storing a list of all the hosts that the application server can connect to.

i Note

The secure user store can any be used for all supported clients. The SAP HANA studio however does not use the SAP HANA secure user store, but the Eclipse secure storage. For more information, see the Eclipse documentation.

The secure user store is installed with the SAP HANA client package. After you install the SAP HANA client, the `hdbuserstore` program is located in one of the following directories:

- `/usr/sap/hdbclient` (Linux/UNIX)
- `%SystemDrive%\Program Files\sap\hdbclient` (Microsoft Windows)

The secure store runs on all platforms supported by SAP HANA client interfaces and SAP BASIS 7.20 EXT.

To access the secure store using JDBC, there are two connect options: `key` and `virtualHostName`. `key` is the `hdbuserstore` key that you use to connect to SAP HANA, while `virtualHostName` specifies the virtual host name. This option allows you to change where the `hdbuserstore` searches for the data and key files.

To connect, define the `hdbuserstore` key using the `key` connect option. JDBC only supports reading the key and data files for existing keys and using those keys to connect to SAP HANA.

Managing the Secure Store

Connection information stored in the secure store is saved in the secure store file `SSFS_HDB.DAT`.

| Operating system | Default location |
|-------------------|---|
| Microsoft Windows | <code><%ALLUSERSPROFILE%>\.hdb\<host-identity>\<security-identifier(SID)>\SSFS_HDB.DAT</code> |
| UNIX | <code><home-directory-of-effective-user-ID>/.hdb/<host-identity>/SSFS_HDB.DAT</code> |

For Microsoft Windows systems:

- `<%ALLUSERSPROFILE%>` stores a path to profile settings for all users on the system. On Microsoft Windows Vista and later, the default value for this environment variable is `C:\ProgramData`.
- `<host-identity>` is determined as follows:
 - If `%HDB_USE_IDENT%` is set, its value determines `<host-identity>`
 - If `<computer-name>` (as returned by `GetComputerName()`) is set, its value determines `<host-identity>`
- `<security-identifier(SID)>` is the Windows Security Identifier of the user that created the Secure User Store.

For UNIX systems:

- `<host-identity>` is determined as follows:
 - If `$HDB_USE_IDENT` is set, its value determines `<host-identity>`
 - If `<virtual-hostname>` is set (by using the `-H` flag, where available), its value determines `<host-identity>`
 - The computer's `<hostname>` determines `<host-identity>`
 - If `$SAPLOCALHOST` is set, its value determines `<host-identity>`
 - If `VIRTUALHOST` connection property is set, its value determines `<host-identity>`
 - On Linux only, if `<virtual-hostname>` is specified on the client install command line (by using the `-H` flag with `hdbinst`), its value determines `<host-identity>`

Use the HDB_USE_IDENT environment variable to store your hdbuserstore in one environment so that users on all hosts can connect to the SAP HANA database using the same secure store information.

If the SSFS_HDB.DAT path does not already exist, then the hdbuserstore program creates it.

The secure store's content is platform-dependent. You cannot copy the secure store from one platform to another.

Managing Connection Information

Use the hdbuserstore program to store and manage connection information in the secure store. For more information about the available commands, see *hdbuserstore Commands*.

The secure user store is user specific, so only the operating system user who owns the corresponding secure store file can access the secure store. However, it is possible, with the appropriate operating system privileges, to manage another user's secure store. This behavior is needed, for example, to manage the connection details for ABAP on Microsoft Windows since the application server is running under a different user (SAPService<SAPSID> instead of <SAPSID>adm).

Using Stored Connection Information

When the secure store is accessed in the context of the correct operating system user, you can open it with a user key.

| Client | How to Connect Using a Stored User Key |
|---------|--|
| HDBSQL | Specify the key to be used with the -U connection option: <code>hdbsql -U <KEY></code> |
| ODBC | Specify the user store key with the @ sign in your data source: <code>servernode=@<KEY></code> |
| ABAP | Specifies the key DEFAULT by default |
| JDBC | Specify the <KEY> and <VIRTUALHOSTNAME> properties to read the secure store directly |
| Node.js | Specify the user store key with the @ sign in your data source: <code>servernode=@<KEY></code> |
| Python | Specify the <USERKEY> (case sensitive) or the <KEY> (case insensitive) in the dbapi.connect function. For example, <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> <pre>dbapi.connect(address=<LOCALHOST>, port=30015, <USERKEY>=<KEY>)</pre> </div> |
| Go | Specify the <KEY>. For example, <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> <pre>hdb://user:password@localhost:30015?<KEY>=<VALIDKEY></pre> </div> |

Managing the Encryption Key

The initial default encryption key of the secure user store is automatically changed when the first entry is created.

In older revisions, password information contained in the secure user store may have been encrypted using the initial default encryption key. As of revision 102, this key is automatically changed the first time the `SET` or `DELETE` commands are executed. You can also change it explicitly by using the `CHANGEKEY` command. The `SET` and `DELETE` commands implicitly execute the `CHANGEKEY` command. For more information, see *Change the Secure User Store Encryption Key* in the *SAP HANA Administration Guide*.

If a user forgets the stored password, then you cannot recover that password because the system does not display passwords in a human-readable form.

Connecting to a Requested Tenant Database

You can associate a key with tenant database information for use in a connection attempt. The server keeps track of which tenant databases are assigned to which ports for a host in the system database. You should only supply the host name/port pair for the system database that you plan to connect to, with each host specification being one of the three master nameservers associated with the system. The database name, when supplied in a connection attempt, is used to query a system database that runs on a well-defined port.

A failover may occur if one or more hosts in the connection list is down. Only one database may be supplied to a port. Therefore, whichever database fails over first is the database that is assigned to the port.

The following example sets the key for a tenant database:

```
set new-key host-name:30013@Tenant-DB-Name myusername mypassword
```

Related Information

[hdbuserstore Commands \[page 226\]](#)

10.3.1.1 hdbuserstore Commands

Several commands are available for managing connection information stored in the secure user store of the SAP HANA client (`hdbuserstore`).

You store and manage connection information in the user store with the `hdbuserstore` program. Execute commands using the following syntax:

```
hdbuserstore [OPTION]... COMMAND [PARAMETER]... }
```

Returned Codes

Every command returns 0 if successful.

If an error occurs, every command returns a positive value, which:

- Has a value of 1 under exceptional circumstances (usually, this is the error).
- Has a value of 100 (SQLDBC_NO_DATA_FOUND) if you use a command that references a `KEY` that is supposed to be found in the store, but the tool cannot find it. This applies to the `LIST` and `DELETE` commands.

i Note

This value is in addition to the possible return value of 1 for any other exceptional store circumstance (for example, a store read/write error).

Command Options

| Option | Description |
|-----------|--|
| -h | Displays a help message |
| -H <HOST> | Assumes host name <HOST> |
| -i | Enables interactive mode |
| -u <USER> | Execute command on the user store of user <USER> |
| -v | Executes command in verbose mode |

i Note

You must have administrator privileges to work on the store of a different user.

Commands

| Command | Parameter | Description |
|---------|-----------|--|
| HELP | - | Displays a help message. |
| LIST | KEY | <p>Lists entries with the specified key, even without any arguments.</p> <p>Passwords are not displayed.</p> <p>Returns a value of 100 (SQLDBC_NO_DATA_FOUND) if the tool cannot find the <code>KEY</code> in the store.</p> |

i Note

This value is in addition to the possible return value of 1 for any other exceptional store circumstance (for example, a store read/write error).

| Command | Parameter | Description |
|-----------|-----------|---|
| DELETE | KEY | <p>Deletes entries with the specified key.</p> <p>Returns a value of 100 (SQLDBC_NO_DATA_FOUND) if the tool cannot find the KEY in the store.</p> <div data-bbox="1007 510 1394 712" style="background-color: #f0f0f0; padding: 5px;"> <p>i Note</p> <p>This value is in addition to the possible return value of 1 for any other exceptional store circumstance (for example, a store read/write error).</p> </div> |
| SET | KEY | <p>Sets the entry key.</p> <div data-bbox="1007 779 1394 1014" style="background-color: #f0f0f0; padding: 5px;"> <p>i Note</p> <p>Underscores, hyphens, periods, and alphanumerical characters (0-9, a-z, A-Z) are allowed. The resulting key is always upper case (0-9, A-Z).</p> </div> |
| | ENV | <p>Sets the connection environment in the format:</p> <ul style="list-style-type: none"> • <host:port> • <host:port@tenant_database>, where port is the SQL port number of the system database |
| | USERNAME | Sets the user name for the profile. |
| | PASSWORD | Sets the password for the profile. |
| | | <div data-bbox="1007 1357 1394 1648" style="background-color: #f0f0f0; padding: 5px;"> <p>i Note</p> <p>We recommend executing the SET command in interactive mode so that you are prompted to enter the password. If you enter the password directly in the command, it is stored in your shell's command history.</p> </div> |
| CHANGEKEY | - | Randomly generates a new master encryption key and re-encrypts password of all keys with the new master key. |

| Command | Parameter | Description |
|-------------|-----------|---|
| ADDFROMDIR | DIR | <p>Adds entries from a store specified by the DIR parameter to the secure user store without overwriting existing keys. You must have read permission for both the SSFS_HDB.DAT and SSFS_HDB.KEY files in the specified directory.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p>i Note</p> <p>This command does not copy trace configurations stored in the SSFS_HDB.DAT file.</p> </div> <p>If this command reads a key name in the store and the same key name is already in use, then the existing key remains unchanged. If this command is interrupted, then the keys that have already been added are not automatically rolled back. An intermittent store backup file is created for manual recovery. The backup file is created before each key is added and is automatically removed after each successful add single key operation.</p> <p>Returns SQLDBC_NO_DATA_FOUND if the store located in the directory DIR has no keys in it.</p> |
| LISTFROMDIR | DIR | <p>Lists entries from a store in the store directory. You must have read permission for both the SSFS_HDB.DAT and SSFS_HDB.KEY files.</p> <p>Returns SQLDBC_NO_DATA_FOUND if the store located in the directory DIR has no keys in it.</p> |

Examples

| Action | Command | Example |
|-------------------------------------|---|--|
| Create a user key in the user store | <pre>hdbuserstore SET <KEY> <host:port> <USERNAME> <PASSWORD></pre> | <pre>hdbuserstore SET millerj "localhost:30115" JohnMiller 2wsx\$RFV</pre> |

| Action | Command | Example |
|--|--|--|
| Create a user key in the user store for a tenant database | <pre>hdbuserstore SET <KEY> <host:port@tenant_database > <USERNAME> <PASSWORD></pre> | <pre>hdbuserstore SET millerj "localhost:30113@DB1" JohnMiller 2wsx\$RFV</pre> |
| List all available user keys (passwords are not displayed) | <pre>hdbuserstore LIST <KEY></pre> | <pre>hdbuserstore LIST millerj</pre> <p>The following information is displayed:</p> <ul style="list-style-type: none"> • KEY: millerj • ENV: localhost:30115 • USER: JohnMiller <p>or (if tenant database was specified):</p> <ul style="list-style-type: none"> • KEY: millerj • ENV: localhost:30113 • USER: JohnMiller • DATABASE: DB1 |
| Configure failover support for application server by specifying a list of host/ports pairs that the server can connect to | <pre>hdbuserstore SET DEFAULT "<hostname_node1>: 3<inst>15; ... ;<hostname_node(n)>: 3<inst>15" <sapsid> <password></pre> | <pre>hdbuserstore SET default "ld9490:33315;ld9491:33315 ;ld9492:33315;ld9493:33315 " SAPP20 <Password></pre> |
| Configure failover support for client running on a distributed tenant database by specifying a list of host/ports pairs that the client can connect to | <pre>hdbuserstore SET <keyname> <hostname>:<System_DB_SQL_ port>[,<hostname2>:<System_ DB_SQL_port>, ...]&@<database_name> <user> <password></pre> | <pre>hdbuserstore SET DB1 "host1:30013;host2:30013;h ost3:30013@DB1" JohnMiller <password></pre> |

→ Remember

The port number is the SQL port of the name server of the system database.

10.3.2 Client-Side Data Encryption

With client-side data encryption, columns that contain sensitive data, such as credit card numbers or social security numbers, are encrypted by using an encryption key accessible only by the client. Client-side encryption makes encryption transparent to applications and column data is encrypted and decrypted on the client-driver, allowing the application to read and write data in cleartext form.

Cleartext data and keys are never available to the SAP HANA server, nor are the cleartext values sent over the network between the client and server. As a result, client-side encryption provides a separation between those who own the data (and can view it) and those who manage the data (but should have no access). Client-side encryption delivers built-in protection of sensitive data from database administrators, cloud administrators, and users who do not need access to cleartext data. Client-side encryption uses both symmetric and asymmetric encryption. Sensitive column data is encrypted with a symmetric column encryption key (CEK) that is encrypted using a client key pair (CKP). A CKP consists of a private key and a public key. The public key

is stored on the SAP HANA server and in the hdbkeystore on the client's local machine. The private key is stored only in the hdbkeystore on the client's local machine.

To access the encrypted data, an application must use a client driver that supports client-side encryption and the client must have access to the CEK that encrypts the column. To distribute these CEKs to clients that need them, key pairs are generated by the clients. The key administrator can then grant access to the CEK, and thus the encrypted data, by decrypting the CEK with their private key, which is stored in their local hdbkeystore, and creating a copy of the CEK. The CEK copy is encrypted with the public key of the CKP of the user who needs access. The CEK copy for the user is stored in the SAP HANA database. When a user requests a CEK from the server, the server sends the copy encrypted with that user's public key and the client-driver decrypts it using the corresponding private key stored locally.

When writing or reading encrypted data to/from the server, the application must use a prepared statement. When an application issues a parameterized query, the client-driver transparently collaborates with the server to encrypt or decrypt the column data using the key information stored in the SAP HANA server and hdbkeystore. To decrypt the data, the client-driver uses the locally stored private key to decrypt the CEK copy of the user and uses the CEK to access the encrypted column data. The driver encrypts bound parameter values using the CEK (obtained as explained above) before passing the data to the SAP HANA server.

Selecting Deterministic Versus Randomized Encryption

The SAP HANA server never operates on cleartext data stored in encrypted columns. However, depending on the encryption type of the column, some queries on encrypted data are supported. There are two types of client-side encryption: non-deterministic (or randomized) encryption and deterministic encryption.

With deterministic encryption, cleartext is always encrypted into the same ciphertext for a given cleartext and CEK, even over separate executions of the encryption algorithm. This consistency permits operations such as equality and inequality comparisons on encrypted data. Deterministic encryption can result in attackers matching cleartext to encrypted values for a column, especially for columns storing low-cardinality data like status flags, boolean values, or major classifications such as gender.

The key administrator must create their key pair in a secure location, so that no other entities have access to their private key. The key administrator must never operate in the same cloud environment that is hosting the SAP HANA server to avoid having the keys or other sensitive data accessible by the server environment.

When a column is encrypted using the non-deterministic algorithm, the same cleartext yields different ciphertexts, even when encrypting the cleartext several times with the same CEK. Due to the randomness of the ciphertext, non-deterministic encryption is stronger, but limits the types of operations that can be performed with the encrypted data. No operations are possible except for basic inserts, updates and fetches.

Choose the encryption algorithm based on the intended use of the data. Use deterministic encryption for columns that are used as search parameters, for example SSN. Use randomized encryption for data that is used for basic inserts, updates, and fetches.

Feature Details and Considerations

Creating and Altering Tables with Encrypted Columns

- Only columns in row and column tables can be altered to encrypt data.
- Tables with encrypted columns must have a primary key, and the primary key cannot be encrypted.
- Column encryption keys must be located in the same schema as the tables whose columns they are encrypting.
- You cannot alter the table type (row to column, or vice versa) of tables with encrypted columns.
- Only encryption-related column alterations are supported for encrypted columns; you can run any alter column operations on unencrypted columns as usual, without affecting encrypted columns.
- You can only alter the encryption status or encryption key for one column at a time.
- You cannot rename an encrypted column.
- Indexes, foreign keys, partition keys, and check constraints are not supported on encrypted columns
- You cannot include an encrypted column in the RESET BY query of the CREATE | ALTER SEQUENCE statements.
- Encrypted columns are not supported in SQLScript, UDFs, and stored procedures.

Querying Tables with Encrypted Columns

- The application must use prepared statements.
- Queries can perform equality look-ups on columns using deterministic encryption, but cannot perform other operations, such as equality joins, greater than/less than, pattern matching using the LIKE operator, or arithmetical operators.
- Only basic inserts, updates, and fetches are possible on columns using random encryption.
For a detailed description of DML limitations, see *DML Limitations With Client-Side Encryption*.

Supported Column Data Types

You can only encrypt the following column data types:

- BOOLEAN
- DATE, TIME, SECONDDATE, TIMESTAMP
- TINYINT, SMALLINT, INTEGER, BIGINT, REAL, DOUBLE
- VARBINARY
- VARCHAR, NVARCHAR

Data Masking

You cannot configure data masking for client-side encrypted columns.

Calculation Views, OLAP Views, and Join Views

You cannot use encrypted columns in calculation views, OLAP views, or join views.

Related Information

[Getting Started With Client-Side Encryption \[page 233\]](#)

10.3.2.1 Getting Started With Client-Side Encryption

Setting up client-side encryption in an SAP HANA database involves deciding which sensitive columns need to be encrypted and encrypting the data in those columns.

Prerequisites

Before setting up client-side encryption, perform the following tasks:

- Use threat modeling to identify sensitive data.
- Decide whether to use deterministic or randomized encryption for your data. For more information on the types of encryption, see *Client-Side Data Encryption*.
- Inform application users about how client-side encryption affects your business solution. For example, share the information described in the "Feature Details and Considerations" section of *Client-Side Data Encryption*.

Context

When using client-side encryption, adhere to the following best practices:

- Be selective about what you encrypt as excessive data encryption can affect functionality and performance.
- Always back up and archive your client key pairs (CKPs) and data and store them in a safe place.
- Assign key administrator privileges to trusted users only.
- Ensure that the key administrator creates their client key pair in a secure location that is different than the cloud environment that houses the SAP HANA server.
- Change your client key pairs (CKPs) and column encryption keys (CEKs) regularly.

Key provisioning is handled in the client, not by the SAP HANA database. Also, when using the client-side encryption feature, all encryption and decryption operations are performed on the client, not on the SAP HANA server.

Procedure

1. Create a key administrator who can create client key pairs and column encryption keys.
2. Create a client key pair. Back up the client key pair and store it in a safe place. If the client machine crashes, then the CKP can be restored to a different machine so that encrypted data can be accessed.
3. Create a column encryption key that is encrypted with the client key pair. Export the column encryption key and store in a safe place.
4. Create new tables with encrypted columns or encrypt existing data in specified database columns.
5. Enable users to access encrypted data by creating key copies of column encryption keys that are encrypted with the user's client key pair.

6. Rotate CEKs regularly and re-encrypt your data using the most current CEK. Key copies for the new CEK must be created for users who need access to data.

Results

You have set up client-side encryption in an SAP HANA database.

Related Information

[Client-Side Data Encryption \[page 230\]](#)

10.3.2.2 Column Encryption Keys

Column encryption keys (CEK) are used to encrypt and decrypt sensitive column data.

Columns with sensitive data can be encrypted by using a column encryption key (CEK) that only clients can access. The server never sees the cleartext value of the encrypted columns or CEKs, and the cleartext values are never sent between the client and server. CEKs are symmetric and are used by the client driver to encrypt and decrypt the column values, allowing the application to read and write data in cleartext form. Each encrypted column can have its own CEK, or CEKs can be shared by multiple encrypted columns.

CEKs are encrypted by using the public part of the client key pair (CKP), and are stored persistently on the server in HANA metadata, along with other encryption information.

CEKs are decrypted on the client, using the corresponding private part of the CKP that is stored in the hdbkeystore of the client's local computer. There can be multiple copies of each CEK in the server metadata, each encrypted using a different client key pair. You can export or import CEKs used to encrypt column data.

CEKs are not stored persistently on the client, but can be cached in memory to improve performance. Any attempt to delete all copies of a column encryption key that is still in use will fail.

Key Administrators

A key administrator is a user with the CLIENTSIDE ENCRYPTION COLUMN KEY ADMIN privilege. This privilege allows the key administrator to create and administer column encryption key copies. When a key administrator creates a new column encryption key (CEK) and encrypts it with their public key, no other system user, including database administrators, can access it unless the key administrator explicitly creates a copy of the CEK for them.

The key administrator grants access to a CEK by creating a copy of the CEK encrypted with the CKP of the user.

While the key administrator creates the encryption key used to protect sensitive data, they do not need access to the sensitive data itself.

If the key administrator creates a copy of the CEK for a user who also has the CLIENTSIDE ENCRYPTION COLUMN KEY ADMIN privilege, or for a user who is able to themselves assign that privilege, then that user can also create CEK copies for other users.

10.3.2.3 Client Key Pairs

Client key pairs (CKPs) are asymmetric keys used to distribute column encryption keys (CEKs) to clients.

CKPs generated by the client-driver, consist of both a public and private key, and are stored in the hdbkeystore on the local computer, along with their name and UUID. The hdbkeystore itself is encrypted and requires a password to be provided from either a user or application.

The key administrator must create their CKP in a secure location so that no other entities have access to their private key. The key administrator must never operate in the same cloud environment that is hosting the HANA server.

Clients register their CKPs with the server, where they are considered database level objects, rather than schema level objects. Registration of a CKP with the server stores the public key of the CKP in the server catalog.

CKPs are not associated with a user; the client driver uses any key pair found in its hdbkeystore. CKPs are not shared between different database systems, so if a client access multiple databases that support client-side encryption, then it needs a unique CKP for each database.

A CKP from an existing client can be exported from its hdbkeystore and imported into a new client's hdbkeystore. The new client can access encrypted columns immediately, providing the existing CKP has been approved and provided with a CEK copy. Export the CKPs to a safe place in case the client computer crashes so that the CKPs can be imported after the computer is recovered.

Since CKPs are identified by UUID, there is no impact to clients and no changes are required for clients when a database system is moved, replicated, or renamed, as long as the new system has the same CEKs, CEK copies, and public keys as the original. When connecting to the new system, CKPs are still identified by the same unique identifier.

The following privileges are required to create or delete a CKP: CREATE CLIENTSIDE ENCRYPTION KEYPAIR privilege and DROP CLIENTSIDE ENCRYPTION KEYPAIR privilege, respectively.

10.3.2.4 Encrypted Columns

Client-side encryption encrypts columns with a column encryption key (CEK). Each encrypted column can have its own CEK, or CEKs can be shared by multiple encrypted columns.

Encrypted columns are represented as a VARBINARY data type. Metadata associated with these columns includes a link to the encrypted CEK, the mode of encryption (random or deterministic), and the encryption status.

There are two types of encryption algorithms used to encrypt columns:

- Deterministic** Using the deterministic algorithm encrypts the given cleartext into the same ciphertext, even over separate executions of the encryption algorithm. This consistency allows operations like equality comparisons. However, this can let users guess information about encrypted data by examining patterns in the data, especially in low cardinality data.
- Random** Using the random algorithm ensures that all plain text values are encrypted into different values. For example, a NULL value in two different records are encrypted into different values. This inconsistency ensures stronger encryption but limits the types of operations that can be performed with the encrypted data.

Related Information

[Client-Side Data Encryption \[page 230\]](#)

[Column Encryption Keys \[page 234\]](#)

10.3.2.5 Secure Key Store (hdbkeystore)

The secure key store (hdbkeystore) is a tool installed with the SAP HANA client. It stores client key pairs for client-side encryption.

The secure key store uses the UUID of the key pair and is encoded as a series of name-value attribute pairs.

Several commands are available for managing information stored in the secure key store. Use the hdbkeystore program to execute commands by using the following syntax:

```
hdbkeystore [OPTION]...COMMAND [PARAMETER]
```

| Option | Description |
|-----------|---|
| -h | Displays a help message |
| -p | Specify the password used to decrypt an encrypted key store |
| -v | Executes the command in verbose mode |
| --version | Prints out the version information for the hdbkeystore. |

| Command | Parameter | Description |
|---------|-----------|--------------------|
| HELP | | Print help message |

| Command | Parameter | Description |
|---------|--------------------------------|---|
| EXPORT | <NAME> <FILENAME> [<PASSWORD>] | Export the specified key or keys to a file. <NAME> The name of the key to be exported; '*' and '?' wildcards can be used to specify multiple keys. <FILENAME> The file to export the key to. <PASSWORD> Optional password to encrypt the private key in the exported file. |
| IMPORT | <FILENAME> [<PASSWORD>] | Import one or more keys from a file. <FILENAME> The file that contains the key to be imported. <PASSWORD> The password that the private key in the import file was encrypted with. |
| LIST | | List all the keys in the store |
| REMOVE | <NAME> | Remove the specified key from the key store. <NAME> can contain * and ? wildcards to remove all keys matching the pattern. |

10.3.3 Protection of Data in SAP HANA Studio Workspaces

When users are working in the SAP HANA studio, data is copied to workspaces on their local disk for editing. This data requires additional protection.

In the SAP HANA studio, data is copied to the following workspaces on the local disks of users:

- Eclipse workspace
When the SAP HANA studio is installed, a local workspace is created by default in the user's home directory in the `hdbstudio` sub-directory. This workspace contains for example, the connection details of SAP HANA systems that the user adds in the SAP HANA studio, as well as other configuration data. It is possible to change the location of this directory using the standard Eclipse *Switch Workplace* feature.
- SAP HANA repository workspaces
In the *SAP HANA Development* perspective of the SAP HANA studio, content and application developers create repository workspaces in a local directory. This allows them to work on local copies of design-time objects from an SAP HANA repository.

To ensure that only the user can access the data in workspaces, workspaces must be created in the user's home directory. In addition, it is recommended that users encrypt the data on their hard drives using an encryption tool.

Users must delete their workspaces when they uninstall the SAP HANA studio.

10.4 Cryptographic Service Provider

All encryption services used in SAP HANA require the availability of a cryptographic service provider on the SAP HANA server and the SAP HANA client.

Server

The SAP HANA server supports the following cryptographic libraries:

- The SAP Cryptographic Library, CommonCryptoLib (default)
CommonCryptoLib (`libsapcrypto.so`) is installed by default as part of SAP HANA server installation at `$DIR_EXECUTABLE`.
CommonCryptoLib supports a FIPS 140-2 compliant cryptographic kernel module. If this is required, you must enable it with the parameter `[cryptography] ccl_fips_enabled` in the `global.ini` file (restart required). For more information, see SAP Note 2117112 and the *SAP HANA Administration Guide*.

i Note

Encryption features available as of SAP HANA 1.0 SPS 09 require CommonCryptoLib.

- OpenSSL
The OpenSSL library is installed by default as part of the operating system installation.

i Note

Deprecated: OpenSSL is deprecated. You must migrate to CommonCryptoLib. For more information, see SAP Note 2093286.

Client

If you are using secure client communication, then the connection property `ENCRYPT=1` must be specified, along with any other relevant TLS/SSL options. Additionally, the `sslKeyStore` must point to a valid private key file.

Related Information

[Client-Side TLS/SSL Connection Properties \(ODBC\) \[page 51\]](#)

[SAP Note 1848999](#)

[SAP Note 2093286](#)

[SAP Note 2117112](#)

11 Auditing Activity in SAP HANA Systems

Auditing provides you with visibility on who did what in the SAP HANA database (or tried to do what) and when. This allows you, for example, to log and monitor read access to sensitive data.

Auditing allows you to monitor and record selected actions performed in the SAP HANA database. Auditing can be enabled individually and independently for every database in the system. Although auditing does not directly increase your system's security, if wisely designed, it can help you achieve greater security in the following ways:

- Uncover security holes if too many privileges were granted to some user
- Show attempts to breach security
- Protect the system owner against accusations of security violations and data misuse
- Allow the system owner to meet security standards

The following actions are typically audited:

- Changes to user authorization
- Creation or deletion of database objects
- Authentication of users
- Changes to system configuration
- Access to or changing of sensitive information

Unauditable Events

Only actions that take place inside the database engine can be audited. If the database engine is not online when an action occurs, it cannot be detected and therefore cannot be audited.

This is important to bear in mind in the following cases:

- Upgrade of an SAP HANA system instance
Upgrade is triggered when the instance is offline. When it becomes available online again, it is not possible to determine which user triggered the upgrade and when.
- Direct changes to system configuration files using operating system commands
Only changes that are made using SQL are visible to the database engine. It is also possible to change configuration files when the system is offline.
- Changing the password of the SYSTEM user of the system database by starting the name server in emergency mode.

We recommend enabling audit logging in the operating system. For more information, see your operating system documentation.

SAP Host Agent

The SAP Host Agent is a tool used to perform several administration tasks in SAP systems. It is installed by default on all hosts.

The SAP HANA database lifecycle manager (HDBLCM) and the SAP HANA cockpit for offline administration rely on the SAP Host Agent to execute tasks as the system administrator user `<sid>adm`.

To audit operations performed in these SAP HANA tools, you must enable the audit logging feature of the SAP Host Agent. For more information about how to do this, see the SAP Host Agent documentation on SAP Help Portal. See also SAP Note 1907566.

The SAP Host agent audit log is written to the Linux syslog.

[Audit Policies \[page 241\]](#)

An audit policy defines the actions to be audited, as well as the conditions under which the action must be performed to be relevant for auditing. When an action occurs, the policy is triggered and an audit event is written to the audit trail. Audit policies are database specific.

[Audit Trails \[page 246\]](#)

When an audit policy is triggered, that is, when an action in the policy occurs under the conditions defined in the policy, an audit entry is created in one or more audit trails.

[Auditing Configuration and Audit Policy Management \[page 254\]](#)

To audit database activity, auditing must first be enabled in the database, and if necessary audit trails configured. It is then possible to create and activate the required audit policies. Audit policies can also be deactivated and reactivated later, or deleted altogether.

[Best Practices and Recommendations for Creating Audit Policies \[page 258\]](#)

Related Information

[SAP Note 1907566](#)

11.1 Audit Policies

An audit policy defines the actions to be audited, as well as the conditions under which the action must be performed to be relevant for auditing. When an action occurs, the policy is triggered and an audit event is written to the audit trail. Audit policies are database specific.

Audited Actions

An action corresponds to the execution of an action in the database by SQL statement. For example, you want to track user provisioning in your system, so you create an audit policy that audits the execution of the SQL statements CREATE USER and DROP USER.

Although most actions correspond to the execution of a single SQL statement, some actions can cover the execution of multiple SQL statements. For example, the action GRANT ANY will audit the granting of multiple entities on the basis of the SQL statements GRANT PRIVILEGE, GRANT ROLE, GRANT STRUCTURED PRIVILEGE, and GRANT APPLICATION PRIVILEGE.

An audit policy can specify any number of actions to be audited, but not all actions can be combined together in the same policy. Actions can be grouped in the following main ways:

- All auditable actions
You can include all actions performed by a specific user in a single policy. This covers not only all other actions that can be audited individually but also actions that cannot otherwise be audited. Such a policy is referred to as a firefighter policy and is useful if you want to audit the actions of a particularly privileged user.

⚠ Caution

The actions that are audited are limited to those that take place inside the database engine while it is running. Therefore, system restart and system recovery will not be audited.

⚠ Caution

Create a firefighter policy only in exceptional circumstances, for example, to check whether a certain user is being used for everyday work or if a support user has been given access to the system. Firefighter policies may create large amounts of audit data and significantly impact performance if they are used for high-load users.

- Data manipulation actions (DML)
You can include any actions that involve data manipulation together in a single policy, for example actions that audit SELECT, INSERT, UPDATE, DELETE, and EXECUTE statements on database objects. A policy that includes these actions requires at least one target object that allows the actions in question. This type of policy is useful if you want to audit a particularly critical or sensitive database object.
- Data definition actions (DDL)
Other action types, for example actions that involve data definition, can only be combined together in a single policy if they are compatible. For example, the action GRANT PRIVILEGE can be combined with REVOKE PRIVILEGE but not with CREATE USER. The action CREATE USER can be combined with DROP USER.

For a full list of all actions that can be audited, see the documentation for SQL access control statement CREATE AUDIT POLICY in the *SAP HANA SQL and Systems View Reference*.

Audit Policy Parameters

In addition to the actions to be audited, an audit policy specifies parameters that further narrow the number of events actually audited.

- Audited action status
For each audit policy, it must be specified when the actions in the policy are to be audited:
 - On successful execution
 - On unsuccessful execution
 - On both successful and unsuccessful execution

i Note

An unsuccessful attempt to execute an action means that the user was not authorized to execute the action. If another error occurs (for example, misspellings in user or object names and syntax errors), the action is generally not audited. In the case of actions that involve data manipulation (that is, INSERT, SELECT, UPDATE, DELETE, and EXECUTE statements), additional errors (for example, invalidated views) are audited.

- Target object(s)

Actions that involve data manipulation require at least one target object. The following target object types are possible:

- Schemas (and all objects contained within)
- Tables
- Views
- Procedures

Target objects are specified at the level of audit policy, so if an audit policy contains several data manipulation actions, the target object must be valid for all actions in the policy. In the case of the action EXECUTE, the only valid target object is procedure. The reverse is also true: the only valid action for procedures is EXECUTE. This means that the action EXECUTE cannot be combined with any other actions. An object does not have to exist before it can be named as the target object of an audit policy. However, if the object does not exist, it cannot be audited by the audit policy. When an object with the specified name is subsequently created, the audit policy will apply for the object, assuming it is of a type that can be audited and the audited action applies to that object type. For example, if the audited action is EXECUTE, the subsequently created object must be a procedure.

i Note

It is not possible to specify a target object for DDL actions.

- Audited user(s)

It is possible to specify that the actions in the policy be audited only when performed by a particular user or users. Alternatively, you can specify that the actions in the policy be audited when performed by all users **except** a particular user or users. In the case of a policy that contains all auditable actions, a user must be specified.

Users do not have to exist before they can be named in an audit policy. However, if a specified user does not exist, it cannot be audited by the audit policy. When the user is subsequently created, the audit policy will apply for the user.

- Audit level

Each audit policy must be assigned one of the following levels:

- EMERGENCY
- ALERT
- CRITICAL
- WARNING
- INFO

When the audit policy is triggered, an audit entry of the corresponding level is written to the audit trail. This allows tools checking audited actions to find the most important information, for example.

Policy-Specific Audit Trail Target(s)

You can optionally configure one or more policy-specific audit trail targets. If you do not configure a policy-specific audit trail target, audit entries generated by the policy are written to the audit trail target for the audit level of the policy if configured, or the audit trail target configured for the system.

If an action is audited by multiple audit policies and these audit policies have different audit trail targets, the audit entry is written to all trail targets.

i Note

Policy-specific audit trails are not possible in tenant databases. The audit trail targets configured for the database or audit level apply, by default internal database table. A system administrator may change the audit trail targets for tenant databases by changing the relevant system property (`[auditing configuration] *_audit_trail_type`) in the `global.ini` file. However, this is not recommended. For more information, see *System Properties for Configuring Auditing*.

Related Information

[Audit Trails \[page 246\]](#)

[System Properties for Configuring Auditing \[page 255\]](#)

[Best Practices and Recommendations for Creating Audit Policies \[page 258\]](#)

11.1.1 Actions Audited by Default Audit Policy

If auditing is active, certain actions are always audited and are therefore not available for inclusion in user-defined audit policies. These actions are audited by the internal audit policy `MandatoryAuditPolicy`.

The actions listed below are always audited and result in audit entries with the audit level CRITICAL. Audit entries are written to the audit trail configured for this audit level. If no audit trail is configured for audit level CRITICAL, entries are written to the audit trail configured for the database.

| Action | Description |
|--|--|
| <ul style="list-style-type: none">CREATE AUDIT POLICYALTER AUDIT POLICYDROP AUDIT POLICY | Creation, modification, or deletion of audit policies |
| <code>ALTER SYSTEM CLEAR AUDIT LOG UNTIL <timestamp></code> | Deletion of audit entries from the audit trail This only applies to the audit trail written to an internal database table. It is not possible to delete audit entries from the syslog audit trail target. |

| Action | Description |
|---|--|
| <ul style="list-style-type: none"> • ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') set ('auditing configuration','global_auditing_state') = <value> with reconfigure; • ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') set ('auditing configuration','default_audit_trail_type') = '<audit_trail_type>' with reconfigure; • ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') set ('auditing configuration','default_audit_trail_path') = '<path>' with reconfigure; • ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') set ('auditing configuration','audit_statement_length') = '<value in bytes>' with reconfigure; • ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') set ('authentication','authentication_methods')= '<methods>' with reconfigure; • ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') unset ('authentication','authentication_methods') with reconfigure; | <p>Changes to auditing configuration, that is:</p> <ul style="list-style-type: none"> • Enabling or disabling auditing • Changing the audit trail target • Changing the location of the audit trail target if it is a CSV text file • Changing the maximum length of a statement that is audited completely • Changing enabled authentication methods |
| <hr/> ALTER DATABASE <database_name> SYSTEM USER PASSWORD <password> | <p>Changing the password of the SYSTEM user of a tenant database from the system database</p> <p>An audit entry is written to the audit trail of both the system database and the tenant database.</p> <hr/> |

11.2 Audit Trails

When an audit policy is triggered, that is, when an action in the policy occurs under the conditions defined in the policy, an audit entry is created in one or more audit trails.

Audit Trail Targets

The following audit trail targets are supported for production systems:

| Audit Trail Target | Description |
|---|---|
| Internal database table | <p>Using an SAP HANA database table as the target for the audit trail makes it possible to query and analyze auditing information quickly. It also provides a secure and tamper-proof storage location. Audit entries are only accessible through the public system views <code>AUDIT_LOG</code>, <code>XSA_AUDIT_LOG</code>, and the union of these two views <code>ALL_AUDIT_LOG</code>. Only <code>SELECT</code> operations can be performed on this view by users with the system privilege <code>AUDIT OPERATOR</code> or <code>AUDIT ADMIN</code>.</p> <p>To avoid the audit table growing indefinitely, it is possible to delete old audit entries by truncating the table. You can do in the SAP HANA cockpit or with the SQL statement <code>ALTER SYSTEM CLEAR AUDIT LOG</code>. The system monitors the size of the table with respect to the overall memory allocation limit of the system and issues an alert when it reaches defined values (by default 5%, 7%, 9%, and 11% of the allocation limit). This behavior can be configured with check 64 ("Total memory usage of table-based audit log"). Only users with the system privilege <code>AUDIT OPERATOR</code> can truncate the audit table.</p> |
| Logging system of the Linux operating system (syslog) | <p>The syslog is a secure storage location for the audit trail because not even the database administrator can access or change it. There are also numerous storage possibilities for the syslog, including storing it on other systems. In addition, the syslog is the default log daemon in UNIX systems. The syslog therefore provides a high degree of flexibility and security, as well as integration into a larger system landscape. For more information about how to configure syslog, refer to the documentation of your operating system.</p> |

⚠ Caution

If the syslog daemon cannot write the audit trail to its destination, you will not be informed. To avoid a situation in which audited actions are occurring but audit entries are not being written to the audit trail, ensure that the syslog is properly configured and that the audit trail target is accessible and has sufficient space available.

| Audit Trail Target | Description |
|-----------------------|--|
| SAP HANA kernel trace | <p>The audit log can be written to a kernel trace file (*.ltc) in the trace directory (/usr/sap/<sid>/<instance>/<host>/trace).</p> <p>The kernel trace output is not human-readable. It must be converted into a CSV-formatted files using the command-line tool <code>hdbtracediag</code> and then loaded into relational tables for SQL analysis.</p> <p><code>hdbtracediag</code> is available on the SAP HANA server at /usr/sap/<sid>/HDB<instance>/exe.</p> |

Additionally, the option exists to store the audit trail in a CSV text file. This should only be used for test purposes in non-production systems. A separate CSV file is created for every service that executes SQL.

⚠ Caution

You must not use a CSV text file for a production system as it has severe restrictions.

Firstly, it is not sufficiently secure. By default, the file is written to the same directory as trace files (/usr/sap/<sid>/<instance>/<host>/trace). This means that database users with the system privilege DATA ADMIN, CATALOG READ, TRACE ADMIN, or INIFILE ADMIN can access it. In the SAP HANA database explorer, it is listed under *Database Diagnostics Files*, and at operating system level, any user in the SAPSYS group can access it.

Secondly, audit trails are created for each server in a distributed database system. This makes it more difficult to trace audit events that were executed across multiple servers (distributed execution).

Multiple Audit Trails

Separate audit trail targets may be configured for the severity of the action being audited, that is the audit level.

Audit entries from audit policies with the audit level EMERGENCY, CRITICAL, or ALERT are written to the audit trail target(s) specified for the audit level in question. If no audit trail target is configured for an audit level, entries are written to the audit trail target configured for the database.

Audit policy-specific targets are also possible in the system database. In this case, audit entries from a particular policy are written to the specified audit trail target(s). If no audit trail target is configured for an audit policy, entries are written to the audit trail target for the audit level if configured, or the audit trail target configured for the database. Several audit trail targets are configurable for each individual policy.

Audit Trails for Tenant Databases

By default, tenant database administrators **cannot** configure audit trail targets independently for their database since the underlying system properties are in the default configuration change blacklist

(`multidb.ini`). The default target for all audit trails in tenant databases is internal database table. Although not recommended, it is possible to change the audit trail target of a tenant database in the following ways:

- The system administrator changes the audit trail targets for individual tenant databases directly by configuring the relevant system property (`[auditing configuration] *_audit_trail_type`) in the `global.ini` file. For more information about the system properties for configuring audit trail targets and the configuration change blacklist in the *SAP HANA Security Guide*.
- The system administrator removes the relevant system property (`[auditing configuration] *_audit_trail_type`) from the configuration change blacklist, thus enabling the tenant database administrator to change the audit trail target.

⚠ Caution

To ensure the privacy of tenant database audit trails, it is recommended that you do **not** change the default audit trail target (internal database table) of tenant databases.

Audit Trail in Active/Active (Read Enabled) Scenario

Active/Active (read enabled) is a feature that enables SAP HANA system replication to support read access on the secondary system. In this scenario, write access to the secondary system is not possible. If internal database table is configured as an audit trail target in the primary system, audit entries are written to `syslog` in the secondary system instead.

i Note

It is possible to configure CSV text file as the alternative audit trail target in the secondary system using the `global.ini` parameter `[auditing configuration] sr_audit_trail_type_cstable_override`. However, this is not recommended for the reasons mentioned above.

Audit Entries

For each occurrence of an audited action, one or more audit entries are created and written to the configured audit trail(s).

❁ Example

If an action that involves data manipulation was executed implicitly by a procedure, the call to this procedure is audited together with the audited action. If the action does not involve data manipulation, then an implicitly executed procedure is not audited. For example, if there is an active audit policy that audits the action of creating users, the execution of `CREATE USER` statements within procedures will be audited but not the procedures themselves.

The layout of audit entries varies depending on the audit trail type.

11.2.1 Audit Trail Layout for Trail Target CSV and SYSLOG

For each occurrence of an audited action, one or more audit entries are created and written to the audit trail. The layout of audit entries varies depending on the audit trail type.

The following table describes the layout of the audit trail when either the syslog or a CSV text file is the trail target:

| Field | Description | Example |
|--------------------|---|---|
| Event Timestamp | Local system time of event occurrence | 2012-09-19 15:44:53 |
| Service Name | Name of the service where the action occurred | Indexserver |
| Hostname | Name of the host where the action occurred | myhanablade23.customer.corp |
| SID | System ID | HAN |
| Instance Number | Instance number | 23 |
| Port Number | Port number | 32303 |
| Database Name | The name of the tenant database | SYSTEMDB or the name of the tenant database |
| | <div style="background-color: #e0e0e0; padding: 5px;"> <p>i Note This field is available only in the syslog audit trail.</p> </div> | |
| Client IP Address | IP address of the client application | 127.0.0.2 |
| Client Name | Name of the client machine | lu241511 |
| Client Process ID | Process ID of the client process | 19504 |
| Client Port Number | Port of the client process | 47273 |
| Policy Name | Audit policy that was triggered | AUDIT_GRANT, MandatoryAuditPolicy |
| Audit Level | Severity of audited action | CRITICAL |
| Audit Action | Action that was audited and thus triggered the policy | GRANT PRIVILEGE |
| Session User | User who is connected to the session | MYADMIN |
| Target Schema | Name of the schema where the action occurred, for example, a privilege was granted on a schema, or a statement was executed on object in a schema | PRIVATE |
| Target Object | Name of the object on which an action was performed, for example, a privilege was granted | HAXXOR |

| Field | Description | Example |
|-----------------------|---|---|
| Privilege Name | Name of the privilege that was granted or revoked | SELECT |
| Grantable | Indication of whether the privilege or role was granted with or without GRANT/ADMIN OPTION | NON GRANTABLE |
| Role Name | Name of the role that was granted or revoked | MONITORING |
| Target Principal | Name of the target user of the action, for example, grantee in a GRANT statement | HAXXOR |
| Action Status | Execution status of the statement | SUCCESSFUL |
| Component | Name of the configuration file in which a parameter value was changed | indexserver.ini |
| Section | Name of the configuration file section in which a parameter value was changed | auditing_configuration |
| Parameter | Name of the configuration parameter whose value was changed | global_auditing status |
| Old Value | Previous value of the parameter | CSVTEXTFILE |
| New Value | New parameter value | CSTABLE |
| Comment | Additional information about failed user connection attempts | user is locked Currently in case of failed logon attempts, the reason for failure appears in this field. |
| Executed Statement | Statement that was executed | GRANT SELECT ON SCHEMA PRIVATE TO HAXXOR |
| Session ID | ID of the session in which the statement was executed | 400006 |
| Application User Name | Application user name | A099999 |
| Role Schema Name | Name of the schema in which a role was created/dropped, or the schema of a granted/revoked role | MYSHEMA |
| Grantee Schema Name | Name of the schema of a granted or revoked role | MYSHEMA |

⚠ Caution

Treat this information with caution. It comes from the application and SAP HANA has no way of verifying its authenticity.

| Field | Description | Example |
|--------------------------|--|--------------------------------------|
| Origin Database Name | Name of the tenant database in which the query originated; relevant for cross-database queries between tenant databases | DB1 |
| Origin User Name | Name of the database user who executed the query in the origin tenant database; relevant for cross-database queries between tenant databases | MYADMIN |
| XS Application User Name | XS application user name | XSA_ADMIN |
| Application Name | Name of the application | sap.hana.cons |
| Statement User Name | Name of the user who executed the statement | DEMO |
| Create Time | Currently only filled for XSA events: Time of event occurrence at client side | 01.05.2018 13:13:21.273 |
| XSA_MESSAGE_IP | Currently only filled for XSA events: IP address of event occurrence | 127.0.0.1 |
| XSA_TENANT | Currently only filled for XSA events: XSA tenant GUID | testAuditlogServiceTenant |
| XSA_UUID | Currently only filled for XSA events: Unique audit log message ID generated by the audit service | A48E7693FEFE0089CA0A776BDAB7F745 |
| XSA_CHANNEL | Currently only filled for XSA events: Communication protocol (for example, http, JCO, websockets) that was used when the audit event was triggered | UI |
| XSA_ATTACHMENT_ID | Currently only filled for XSA events: ID of the attachment that triggered the event | id:456 |
| XSA_ATTACHMENT_NAME | Currently only filled for XSA events: Name of the attachment that triggered the event | File.text |
| XSA_ORGANIZATION_ID | Currently only filled for XSA events: Application organization GUID | 48a341df-5869-4b80-ba5e-dc9102d06e70 |
| XSA_SPACE_ID | Currently only filled for XSA events: Application space GUID | 87c9f8ff-511b-4a81-a6b8-d81b701839d7 |
| XSA_INSTANCE_ID | Currently only filled for XSA events: GUID of the used auditlog service instance | 74f3d921-55b9-434a-bbf1-13487e424828 |
| XSA_BINDING_ID | Currently only filled for XSA events: Application binding GUID in regards to the specific auditlog service instance that is being used | 358ca146-490e-4fb6-bc18-693720f6f089 |

| Field | Description | Example |
|------------------|--|--|
| XSA_OBJECT | Currently only filled for XSA events: Object containing the accessed personal data | {"type":"type","id": {"key1":"value1","key2":"value2","key3":"value3"}}} |
| XSA_DATA_SUBJECT | Currently only filled for XSA events: Owner of the accessed personal data | {"type":"type","role":"role","id": {"key1":"value1","key2":"value2","key3":"value3"}}} |

Example

```
2013-11-30 13:04:54;indexserver;myhanablade23.customer.corp;HAN;
01;30103;10.29.14.177;lu306309;6776;58060;Alter User Policy;INFO;ALTER
USER;SYSTEM;;;;;ADAMS;SUCCESSFUL;;;;;alter user ADAMS VAXXXXXXXXXXXXXX;
434597;
```

11.2.2 Audit Trail Layout for Trail Target Database Table

For each occurrence of an audited action, one or more audit entries are created and written to the audit trail. The layout of audit entries varies depending on the audit trail type.

When database table is the trail target, audit entries are accessible through the public system views AUDIT_LOG and XSA_AUDIT_LOG, as well as the union of these views ALL_AUDIT_LOG. The table below describes the layout of the full audit trail, that is ALL_AUDIT_LOG.

Note

Only SELECT operations can be performed on these views by users with the system privilege AUDIT OPERATOR or AUDIT ADMIN.

| Column name | Data type | Description |
|---------------|---------------|--|
| TIMESTAMP | TIMESTAMP | Specifies the time that the event occurred. |
| HOST | VARCHAR(64) | Specifies the name of the host where the event occurred. |
| PORT | INTEGER | Specifies the port number. |
| SERVICE_NAME | VARCHAR(32) | Specifies the name of the service. |
| CONNECTION_ID | INTEGER | Specifies the connection ID. |
| CLIENT_HOST | NVARCHAR(256) | Specifies the IP of the client host. |
| CLIENT_IP | VARCHAR(45) | Specifies the IP of the client application. |

| Column name | Data type | Description |
|-----------------------|----------------|--|
| CLIENT_PID | BIGINT | Specifies the PID of the client process. |
| CLIENT_PORT | INTEGER | Specifies the port of the client process. |
| USER_NAME | NVARCHAR(256) | Specifies the name of the user that is connected to the database. |
| STATEMENT_USER_NAME | NVARCHAR(256) | Specifies the name of the user who executed the statement. |
| APPLICATION_NAME | NVARCHAR(256) | Specifies the name of the application. |
| APPLICATION_USER_NAME | NVARCHAR(256) | Specifies the name of the application user. |
| AUDIT_POLICY_NAME | NVARCHAR(256) | Specifies the name of the Audit Policy hit. |
| EVENT_STATUS | VARCHAR(32) | Specifies whether the event was successful or not. |
| EVENT_LEVEL | VARCHAR(16) | Specifies the severity level of the event. |
| EVENT_ACTION | VARCHAR(64) | Specifies the action performed by the audit event. |
| KEY | NVARCHAR(2000) | Specifies the attribute that was changed. |
| PREV_VALUE | NVARCHAR(5000) | Specifies the old value of the attribute. |
| VALUE | NVARCHAR(5000) | Specifies the new value of the attribute. |
| STATEMENT_STRING | NCLOB | Specifies the SQL statement that caused the event. |
| COMMENT | VARCHAR(5000) | Specifies extra information about the event. |
| CREATE_TIME | TIMESTAMP | (Applies to XSA-events) Specifies the time of the event occurrence at the client side. |
| XSA_MESSAGE_IP | VARCHAR(45) | (Applies to XSA-events) Specifies the IP address of the event occurrence. |
| XSA_TENANT | VARCHAR(36) | (Applies to XSA-events) Specifies the XSA tenant GUID. |

| Column name | Data type | Description |
|---------------------|----------------|---|
| XSA_UUID | VARCHAR(256) | (Applies to XSA-events) Specifies the unique audit log message ID generated by the audit service. |
| XSA_CHANNEL | VARCHAR(16) | (Applies to XSA-events) Specifies the communication protocol that was used when the audit event was triggered. |
| XSA_ATTACHMENT_ID | NVARCHAR(256) | (Applies to XSA-events) Specifies the ID of the attachment that triggered the event. |
| XSA_ATTACHMENT_NAME | NVARCHAR(256) | (Applies to XSA-events) Specifies the name of the attachment that triggered the event. |
| XSA_ORGANIZATION_ID | VARCHAR(36) | (Applies to XSA-events) Specifies the application organization GUID. |
| XSA_SPACE_ID | VARCHAR(36) | (Applies to XSA-events) Specifies the application space GUID. |
| XSA_INSTANCE_ID | VARCHAR(36) | (Applies to XSA-events) Specifies the GUID of the used auditlog service instance. |
| XSA_BINDING_ID | VARCHAR(36) | (Applies to XSA-events) Specifies the application binding GUID in regards to the specific auditlog service instance that is being used. |
| XSA_OBJECT | NVARCHAR(5000) | (Applies to XSA-events) Specifies the object containing the accessed personal data. |
| XSA_DATA_SUBJECT | NVARCHAR(5000) | (Applies to XSA-events) Specifies the owner of the accessed personal data. |

11.3 Auditing Configuration and Audit Policy Management

To audit database activity, auditing must first be enabled in the database, and if necessary audit trails configured. It is then possible to create and activate the required audit policies. Audit policies can also be deactivated and reactivated later, or deleted altogether.

System properties in the `global.ini` file (tenant databases) and the `nameserver.ini` file (system database) control the configuration of auditing and audit trail targets. However, we recommend that you

configure auditing and manage audit policies on the [Auditing](#) page of the SAP HANA cockpit or the [Security](#) editor of the SAP HANA studio.

Related Information

[System Properties for Configuring Auditing \[page 255\]](#)

11.3.1 System Properties for Configuring Auditing

The system properties for configuring auditing are in the `auditing` configuration section of the `global.ini` system properties file (tenant databases) or `nameserver.ini` file (system database).

The following system properties are used to configure auditing. It is recommended that you not edit these properties directly, but use the [Auditing](#) page of the SAP HANA cockpit or the [Security](#) editor of the SAP HANA studio instead.

i Note

All `*_audit_trail_type` properties and the property `default_audit_trail_type` are in the default configuration change blacklist (`multidb.ini`). This means that they cannot initially be changed in tenant databases. They must be changed from the system database. If appropriate for your scenario, you can remove these properties from the change blacklist. SAP HANA deployment options are described in the *SAP HANA Master Guide*. For more information about how to edit the change blacklist, see the *SAP HANA Administration Guide*.

| System Property | Value | Default Value | Description |
|-------------------------------------|-----------------|---------------|------------------------------------|
| <code>global_auditing_status</code> | <Boolean value> | false | Status of auditing in the database |

i Note

In the system database, this property can only be in the `nameserver.ini` file, not `global.ini`. This makes it possible to enable auditing for the system database independently of tenant databases.

| System Property | Value | Default Value | Description |
|----------------------------|---|--|--|
| default_audit_trail_type | {SYSLOGPROTOCOL CSTORE CSVTEXTFILE KERNELTRACE} | <ul style="list-style-type: none"> SYSLOGPROTOCOL if global_auditing_state is true (system database) CSTORE if global_auditing_state is true (tenant database) | Overall audit trail target of the database |
| default_audit_trail_path | <file> | /usr/sap/<sid>/<instance>/<host>/trace if default_audit_trail_type is CSVTEXTFILE (system database) | The file path of audit trail target CSVTEXTFILE |
| emergency_audit_trail_type | {SYSLOGPROTOCOL CSTORE CSVTEXTFILE KERNELTRACE} | -- | Audit trail target to which audit entries from audit policies with the audit level EMERGENCY are written |
| alert_audit_trail_type | {SYSLOGPROTOCOL CSTORE CSVTEXTFILE KERNELTRACE} | -- | Audit trail target to which audit entries from audit policies with the audit level ALERT are written |
| critical_audit_trail_type | {SYSLOGPROTOCOL CSTORE CSVTEXTFILE KERNELTRACE} | -- | Audit trail target to which audit entries from audit policies with the audit level CRITICAL are written |

| System Property | Value | Default Value | Description |
|------------------------|--------------------------------|----------------|---|
| audit_statement_length | <Value in bytes> | -1 | <p>The maximum length of a statement that is audited completely</p> <p>Statements that exceed the maximum length are truncated once the limit is reached.</p> <p>The default value sets no limit. The complete statement is written to the audit trail.</p> |
| sr_audit_trail_type | {SYSLOGPROTOCOL CSVTEXTFILE} | SYSLOGPROTOCOL | <p>In an active/active (read enabled) scenario, the audit trail target to which audit entries are written in the secondary system if CSTABLE is configured as a trail type in the primary system.</p> <p>This is necessary because it is not possible to write to an internal database table in the secondary system.</p> |
| _cstable_override | | | |

⚠ Caution

Limiting the length of the audit statement string output might compromise your audit log. For example, an attacker who knows about this limitation can simply prefix sensitive statements with the corresponding number of whitespace characters to prevent the actual statement string being written to the audit trail.

❁ Example

- Enable auditing in system database: `ALTER SYSTEM ALTER CONFIGURATION ('nameserver.ini', 'system') set ('auditing configuration', 'global_auditing_state') = 'true' ;`
- Enable auditing in tenant database from that database: `ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'system') set ('auditing configuration', 'global_auditing_state') = 'true' ;`

11.4 Best Practices and Recommendations for Creating Audit Policies

General Best Practices

To reduce the performance impact of auditing, some basic guidelines for creating audit policies apply.

- Create as few audit policies as possible. It's usually better to have one complex policy than several simple ones.

→ Remember

Some audit actions can't be combined in the same policy.

- Use audit actions that combine other actions where possible.

❖ Example

Audit the `GRANT ANY` action instead of the `GRANT PRIVILEGE` and the `GRANT STRUCTURED PRIVILEGE` actions.

- Create audit policies for DML actions only if required. Auditing DML actions impacts performance more than auditing DDL actions.
- Don't create audit policies for actions that are automatically audited, for example `CREATE AUDIT POLICY`. For a list of actions that are always audited, see the section on the default audit policy in the *SAP HANA Security Guide*.
- Don't create audit policies for database-internal tables that are involved in administration actions. Create policies for the administration actions themselves.

❖ Example

`P_USER_PASSWORD` is an internal database tables that cannot be accessed by any user, not even `SYSTEM`. Changes in these tables are carried out by internal mechanisms, and not by DML operations. Don't include these tables in an audit policy. Instead create an audit policy for changes to users (`ALTER USER` action) instead.

- Create a firefighter policy (that is, a policy that audits all actions for a user) only in exceptional circumstances, for example, to check whether a certain user is being used for everyday work or if a support user has been given access to the system. Firefighter policies may create large amounts of audit data and significantly impact performance if they are used for high-load users.

Recommended Audit Policies

Once auditing is active in the database, certain actions are always audited in the internal audit policy `MandatoryAuditPolicy`. In addition, consider the following recommendations.

Audit policies for administrative activities

At a minimum, we recommend that you create audit policies in development and production systems to audit the following additional administrative activities:

- Changes to SAP HANA configuration files (*.ini files). The relevant audit action is `SYSTEM CONFIGURATION CHANGE`.

Sample Code

```
CREATE AUDIT POLICY "configuration changes" AUDITING SUCCESSFUL SYSTEM
CONFIGURATION CHANGE LEVEL WARNING;
ALTER AUDIT POLICY "configuration changes" ENABLE;
```

- Changes to users. The relevant audit actions are:
 - `CREATE USER`
 - `ALTER USER`
 - `DROP USER`

Sample Code

```
CREATE AUDIT POLICY "user administration" AUDITING SUCCESSFUL CREATE USER,
ALTER USER, DROP USER LEVEL INFO;
ALTER AUDIT POLICY "user administration" ENABLE;
```

- Changes to authorization. The relevant audit actions are:
 - `GRANT ANY`
 - `REVOKE ANY`

Sample Code

```
CREATE AUDIT POLICY "authorizations" AUDITING SUCCESSFUL GRANT ANY, REVOKE
ANY LEVEL INFO;
ALTER AUDIT POLICY "authorizations" ENABLE;
```

If design-time roles and authorizations are used, also audit the execution of the grant/revoke of design-time roles and privileges.

Sample Code

```
CREATE AUDIT POLICY "designtime privileges" AUDITING SUCCESSFUL
EXECUTE on _SYS_REPO.GRANT_ACTIVATED_ANALYTICAL_PRIVILEGE,
_SYS_REPO.GRANT_ACTIVATED_ROLE,
_SYS_REPO.GRANT_APPLICATION_PRIVILEGE,
_SYS_REPO.GRANT_PRIVILEGE_ON_ACTIVATED_CONTENT,
_SYS_REPO.GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT,
_SYS_REPO.REVOKE_ACTIVATED_ANALYTICAL_PRIVILEGE,
_SYS_REPO.REVOKE_ACTIVATED_ROLE,
_SYS_REPO.REVOKE_APPLICATION_PRIVILEGE,
_SYS_REPO.REVOKE_PRIVILEGE_ON_ACTIVATED_CONTENT,
_SYS_REPO.REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT
LEVEL INFO;
ALTER AUDIT POLICY "designtime privileges" ENABLE;
```

Additional policies in production systems

In production systems, additional audit policies are usually required to log further activities as defined by IT policy and to meet governance and legal requirements such as SOX compliance.

We also recommend auditing not only successful events but unsuccessful events by defining the audit action status `ALL`. Knowing about unsuccessful events might be a prerequisite to discovering an attack on your system.

Caution

SAP HANA audit policies are defined at the database level and cannot cover all requirements for data protection and privacy. The business semantics of data are part of the application definition and implementation. It is therefore the application that "knows", for example, which tables in the database contain sensitive personal data, or how business level objects, such as sales orders, are mapped to technical objects in the database.

Related Information

[Actions Audited by Default Audit Policy \[page 244\]](#)

12 Certificate Management in SAP HANA

SAP HANA uses X.509 client certificates as the basis for securing internal and external communication channels, as well as for several user authentication mechanisms. Certificates can be stored and managed in files in the file system and in some cases directly in the SAP HANA database.

Certificate Management in the Database

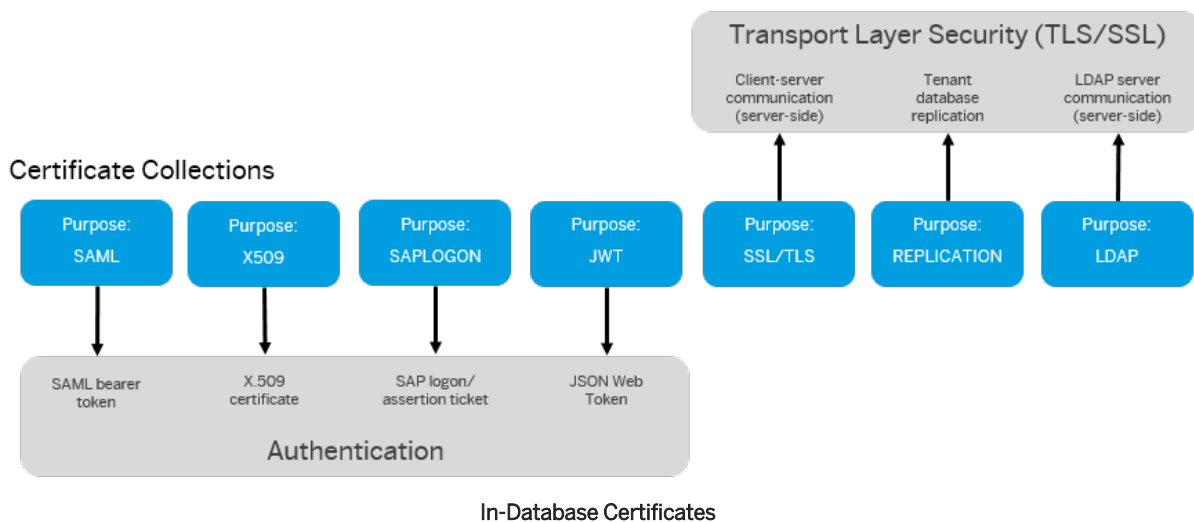
All certificate-based user authentication mechanisms in SAP HANA, as well as secure communication between SAP HANA and clients that access the SQL interface of the database rely on X.509 client certificates for authentication and verifying digital signatures. For ease of management, it's possible to store these certificates and configure their usage directly in the SAP HANA database.

In addition, in-database certificates must be used to secure communication during the process of copying or moving a tenant database between two systems, and to secure communication between SAP HANA and an LDAP server being used for user authentication and authorization.

The following figure shows for which purposes in-database certificates stored in certificate collections can be used. In-database certificates and certificate collections can be fully managed in the SAP HANA cockpit.

i Note

Although we recommend creating and managing both certificates and certificate collections in the database, files containing certificates may also be stored in the file system.



Certificate Management in the File System

Although we recommend using in-database storage where possible, you can store and manage certificates in trust and key stores located in the file system, in so-called personal security environments or PSEs.

⚠ Caution

By default, the same PSE in the file system is shared by all databases for all external communication channels (including HTTP) and certificate-based authentication. Different PSEs must be explicitly configured for tenant databases.

→ Recommendation

You can migrate certificates from file-system based storage to in-database storage. If you do migrate certificates in the file system to the database, delete all related files from the file system to avoid any potential conflicts. For more information, see SAP Note 2175664.

However, not all certificates can be stored in the database, in particular the certificates required to secure internal communication channels using the system public key infrastructure (system PKI), and HTTP client access using SAP Web Dispatcher. These certificates are contained in PSE files located in the file system.

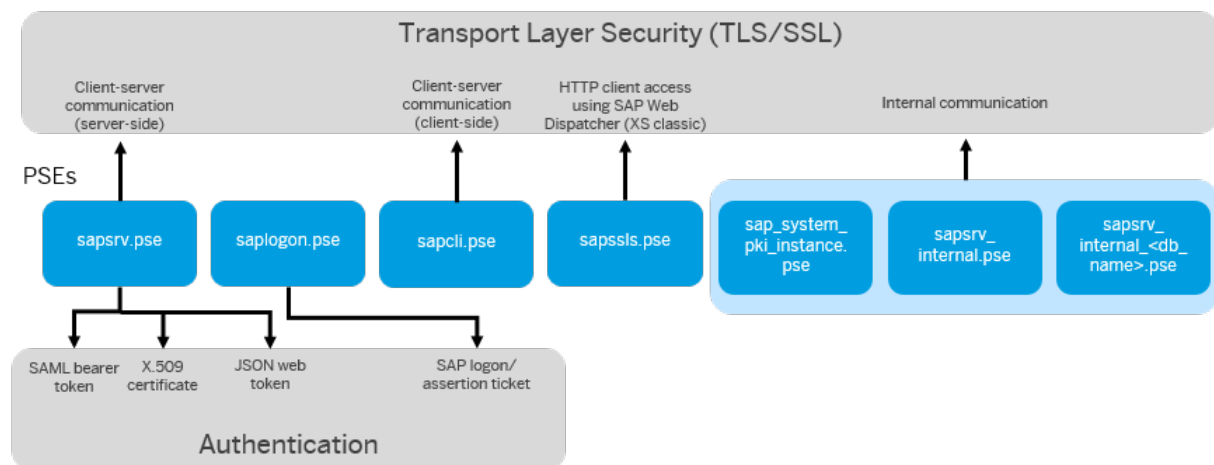
⚠ Caution

Do not delete these files from the file system.

The following figures shows for which purposes certificates stored in PSEs in the file system are possible. These PSEs are available by default and can be managed using for example the SAP Web Dispatcher administration tool or the SAPGENPSE tool, both of which are delivered with SAP HANA. If you are using OpenSSL, you can also use the tools provided with OpenSSL.

i Note

OpenSSL is deprecated. If you are using OpenSSL, migrate to CommonCryptoLib. For more information, see SAP Note 2093286.



Default File-Based PSEs

[In-Database Certificate Management Workflow \[page 264\]](#)

Managing certificates in the SAP HANA database follows a typical workflow. A full separation of duties is possible through user authorization. The full workflow is supported by the SAP HANA cockpit.

[Client Certificates \[page 265\]](#)

X.509 client certificates required for certificate-based authentication and secure communication between SAP HANA and clients that access the SQL interface of the database can be stored and managed directly in the SAP HANA database.

[Certificate Collections \[page 265\]](#)

A certificate collection (or PSE) is a secure location where the public information (public-key certificates) and private information (private keys) of the SAP HANA server are stored. A certificate collection may also contain the public information (public-key certificates) of trusted communication partners or root certificates from trusted Certification Authorities.

[SQL Statements and Authorization for In-Database Certificate Management \(Reference\) \[page 267\]](#)

All administration tasks related to in-database certificate management can be performed using SQL.

Related Information

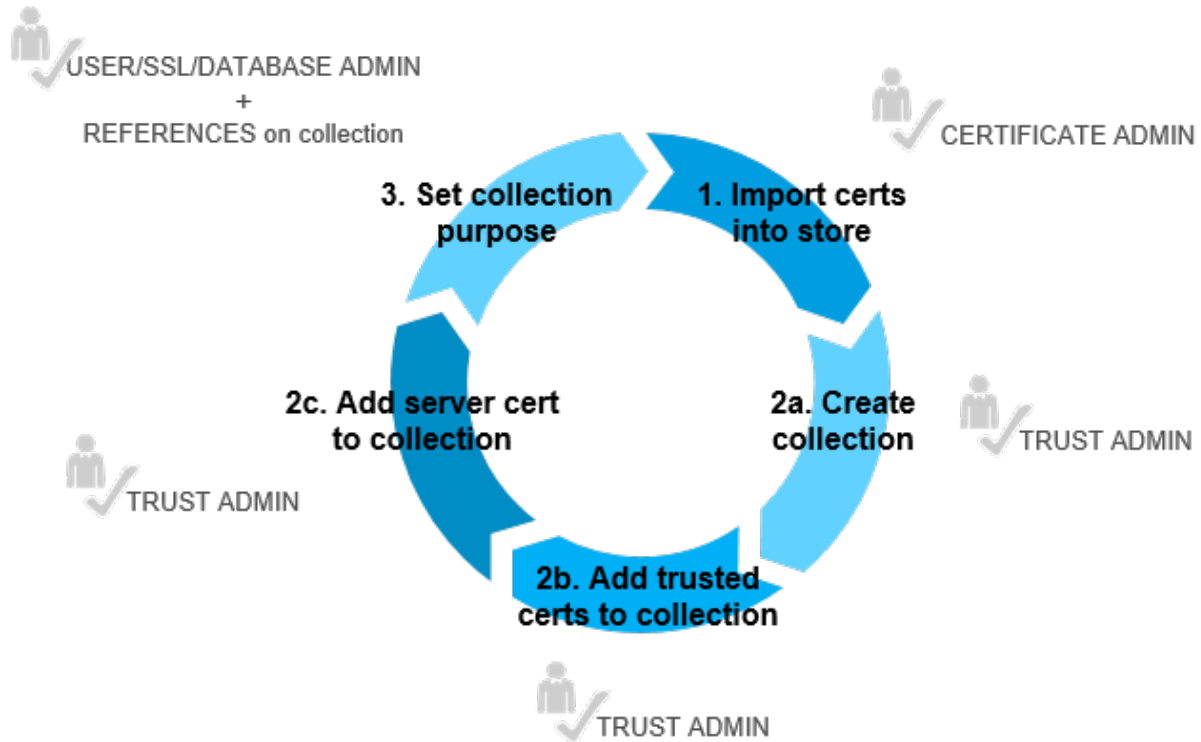
[TLS/SSL Configuration on the SAP HANA Server \[page 44\]](#)

[SAP Note 2175664](#)

[SAP Note 2093286](#)

12.1 In-Database Certificate Management Workflow

Managing certificates in the SAP HANA database follows a typical workflow. A full separation of duties is possible through user authorization. The full workflow is supported by the SAP HANA cockpit.



In-Database Certificate Management Workflow

1. A user with `CERTIFICATE ADMIN` privilege imports into the certificate store the public-key certificates of trusted communication partners, as well as the root certificates of trusted Certification Authorities.
2. A user with `TRUST ADMIN` privilege:
 1. Creates the required certificate collections.
 2. Adds trusted certificates from the certificate store to certificate collections.
 3. Adds the SAP HANA server certificate(s) to those collections that will be used for server authentication (for example, secure client-server communication over JDBC/ODBC).
3. A user with `USER ADMIN`, `SSL ADMIN`, or `DATABASE ADMIN` privilege sets the purpose of individual collections. Which privilege is required depends on the purpose being set.

12.2 Client Certificates

X.509 client certificates required for certificate-based authentication and secure communication between SAP HANA and clients that access the SQL interface of the database can be stored and managed directly in the SAP HANA database.

Certificates stored in the SAP HANA database can be used for:

- Trust validation
Certificates used for trust validation are the public-key certificates of trusted communication partners or root certificates from trusted Certification Authorities. These certificates contain the public part of a user's or component's public and private key pair.
- Server authentication
Certificates used for server authentication are the public-key certificates of the SAP HANA server used to identify the server to connecting clients. In addition to the public-key information of the server, these certificates contain the server's private keys, as well as the intermediate certificates that complete the trust chain from the server certificate to the root certificate that the communication partner (client) trusts.

i Note

Private keys are stored securely using the internal application encryption service of the SAP HANA database. For more information, see *Server-Side Data Encryption* in the *SAP HANA Security Guide*.

Once they have been imported into the database, certificates can be assigned to certificate collections. Certificate collections are also created and managed directly in the database, where they serve a unique purpose (either secure client-server communication or a certificate-based authentication mechanism).

i Note

Although we recommend creating and managing both certificates and certificate collections in the database, files containing certificates may also be stored in the file system.

Related Information

[Certificate Collections \[page 265\]](#)

[Server-Side Data Encryption Services \[page 205\]](#)

12.3 Certificate Collections

A certificate collection (or PSE) is a secure location where the public information (public-key certificates) and private information (private keys) of the SAP HANA server are stored. A certificate collection may also contain the public information (public-key certificates) of trusted communication partners or root certificates from trusted Certification Authorities.

Certificate collections can be created and managed as database objects directly in the SAP HANA database.

Certificate collections uniquely serve one of the following purposes in the database in which they exist:

i Note

Although we recommend creating and managing both certificates and certificate collections in the database, files containing certificates may also be stored in the file system.

- User authentication based on:
 - SAML assertions
 - X.509 certificates
 - Logon and assertion tickets
 - JSON Web Token (JWT)
- Client-server communication over JDBC/ODBC secured using TLS/SSL
- Database replication for the purposes of copying or moving a tenant database to another system
- Communication between SAP HANA and an LDAP server being used for user authentication and authorization

Only one certificate collection may serve one of these purposes at any given time.

The client certificates required for each purpose are assigned to the corresponding certificate collection from the in-database certificate store. A certificate can be assigned to more than one certificate collection.

Certificates used for server authentication, that is certificates that include the private key of the server, need only be assigned to the certificate collection used for secure client-server communication.

Ownership of Certificate Collections

A certificate collection is a database object created in runtime. It is therefore owned by the database user who creates it. If a certificate collection is in use, in other words it has been assigned one of the above purposes, it is not possible to change it (for example, add or remove certificates) or to delete it. However, if the owner of the certificate collection is deleted, the certificate collection will be deleted **even if it currently in use**.

⚠ Caution

The deletion of a certificate collection that is assigned a purpose could render the database unusable. For example, if TLS/SSL is being enforced for all client connections and the certificate collection used for TLS/SSL is deleted, no new client connections to the database can be opened.

Related Information

[SAP HANA Authentication and Single Sign-On \[page 92\]](#)

[Secure Communication Between SAP HANA and JDBC/ODBC Clients \[page 43\]](#)

12.4 SQL Statements and Authorization for In-Database Certificate Management (Reference)

All administration tasks related to in-database certificate management can be performed using SQL.

The following table lists the SQL statements for creating and managing certificates and certificate collections in the SAP HANA database, including the required authorization for each task.

i Note

Certificate collections are referred to as personal security environments (PSEs) in back-end terminology.

| To... | Execute the Statement... | With the Authorization... |
|--|--|--|
| See certificates in the in-database certificate store | <pre>SELECT * FROM CERTIFICATES</pre> <div data-bbox="603 840 991 1010"> <p>i Note</p> <p>You can also view certificates in the Certificate Store app of the SAP HANA cockpit.</p> </div> | <p>System privilege CERTIFICATE ADMIN or TRUST ADMIN</p> <p>If you have object privilege ALTER on a certificate collection, you'll also be able to see the certificates used in this collection.</p> |
| See certificate collections | <pre>SELECT * FROM PSES</pre> <div data-bbox="603 1081 991 1252"> <p>i Note</p> <p>You can also view certificate collections in the Certificate Store app of the SAP HANA cockpit.</p> </div> | <p>System privilege TRUST ADMIN</p> <p>If you have object privilege ALTER, DROP, or REFERENCES on a certificate collection, you'll also be able to see this collection.</p> |
| See which certificates are used in a certificate collection | <pre>SELECT * FROM PSE_CERTIFICATES</pre> <div data-bbox="603 1357 991 1527"> <p>i Note</p> <p>You can also see this information on the Certificate Store app of the SAP HANA cockpit.</p> </div> | <p>Object privilege ALTER, DROP, or REFERENCES on the certificate collection</p> |
| Add a certificate to the in-database certificate store | <pre>CREATE CERTIFICATE FROM <certificate_content> [COMMENT <comment>]</pre> | <p>System privilege CERTIFICATE ADMIN</p> |
| Delete a certificate from the in-database certificate | <pre>DROP CERTIFICATE <certificate_id></pre> | <p>System privilege CERTIFICATE ADMIN</p> |
| <div data-bbox="205 1749 585 1919"> <p>i Note</p> <p>If the certificate has already been added to a certificate collection, it can't be deleted.</p> </div> | | |

| To... | Execute the Statement... | With the Authorization... |
|---|--|--|
| View certificate collections in the database, including the certificates they contain | <pre>SELECT * FROM PSE_CERTIFICATES</pre> <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>You can also view certificate collections on the <i>Certificate Collection</i> app of the SAP HANA cockpit.</p> </div> | System privilege CATALOG READ and either TRUST ADMIN, USER ADMIN, or SSL ADMIN |
| Create a certificate collection | <pre>CREATE PSE <PSE_name></pre> | System privilege TRUST ADMIN |
| Add a public-key certificate to a certificate collection | <pre>ALTER PSE <PSE_name> ADD CERTIFICATE <certificate_id></pre> | <ul style="list-style-type: none"> Nothing if you're the owner of the certificate collection Object privilege ALTER on the certificate collection if you're not the owner |
| Remove a public-key certificate from a certificate collection | <pre>ALTER PSE <PSE_name> DROP CERTIFICATE <certificate_id></pre> <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If the purpose of the certificate collection already been set, then system privilege USER ADMIN or SSL ADMIN is additionally required depending on whether the purpose is user authentication or secure communication.</p> </div> | |
| Add a private key to a certificate collection | <pre>ALTER PSE <PSE_name> SET OWN CERTIFICATE <certificate_content></pre> | <ul style="list-style-type: none"> Nothing if you're the owner of the certificate collection Object privilege ALTER on the certificate collection if you're not the owner |
| Set the purpose of a certificate collection | <pre>SET PSE <PSE_name> PURPOSE <PSE_purpose></pre> <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p>i Note</p> <p>If the purpose of the PSE is SSL/TLS, then it must already have a private key added.</p> </div> | <ul style="list-style-type: none"> USER ADMIN if the purpose is user authentication (SAML, X.509, JWT, or logon tickets) SSL ADMIN if the purpose is secure client-server communication (SSL/TLS) DATABASE ADMIN if the purpose is copying or moving a tenant database between systems LDAP ADMIN if the purpose is LDAP-based user authentication and authorization |
| | <p>The following PSE purposes are possible:</p> <ul style="list-style-type: none"> DATABASE REPLICATION JWT LDAP SAML SAP LOGON SSL/TLS X509 | |

| To... | Execute the Statement... | With the Authorization... |
|---|---|--|
| Unset the purpose of a certificate collection | <pre>UNSET PSE <PSE_name> PURPOSE <PSE_purpose></pre> | <p>i Note</p> <p>Object privilege REFERENCES on the certificate collection is additionally required if you are not the owner of the collection.</p> |
| Delete a certificate collection | <pre>DROP PSE <PSE_name></pre> | <ul style="list-style-type: none"> • Nothing, if you're the owner of the certificate collection • Object privilege DROP on the certificate collection, if you're not the owner |

i Note

If the certificate collection has already been assigned a purpose, it can't be deleted.

13 Data Protection and Privacy in SAP HANA

SAP HANA provides the technical enablement and infrastructure to allow you run applications on SAP HANA to conform to the legal requirements of data protection in the different scenarios in which SAP HANA is used.

Introduction to Data Protection

Data protection is associated with numerous legal requirements and privacy concerns. In addition to compliance with general data privacy regulations, it is necessary to consider compliance with industry-specific legislation in different countries. SAP HANA provides specific features and functions to support compliance with regard to relevant legal requirements, including data protection.

This section and any other sections in this Security Guide do not give any advice on whether these features and functions are the best method to support company, industry, regional, or country-specific requirements. Furthermore, this guide does not give any advice or recommendations regarding additional features that would be required in specific IT environments; decisions related to data protection must be made on a case-by-case basis, taking into consideration the given system landscape and the applicable legal requirements.

Note

In the majority of cases, compliance with data privacy laws is not a product feature. SAP software supports data privacy by providing security features and specific functions relevant to data protection, such as functions for the simplified blocking and deletion of personal data. SAP does not provide legal advice in any form. The definitions and other terms used in this guide are not taken from any given legal source.

Glossary

| Term | Definition |
|-----------------------------|--|
| Blocking | A method of restricting access to data for which the primary business purpose has ended. |
| Business purpose | A legal, contractual, or in other form justified reason for the processing of personal data. The assumption is that any purpose has an end that is usually already defined when the purpose starts. |
| Consent | The action of the data subject confirming that the usage of his or her personal data shall be allowed for a given purpose. A consent functionality allows the storage of a consent record in relation to a specific purpose and shows if a data subject has granted, withdrawn, or denied consent. |
| Deletion | Deletion of personal data so that the data is no longer available. |
| End of purpose (EoP) | End of purpose and start of blocking period. The point in time, when the primary processing purpose ends (e.g. contract is fulfilled). |

| Term | Definition |
|-----------------------------------|---|
| End of purpose (EoP) check | A method of identifying the point in time for a data set when the processing of personal data is no longer required for the primary business purpose . After the EoP has been reached, the data is blocked and can only be accessed by users with special authorization (for example, tax auditors). |
| Personal data | Any information relating to an identified or identifiable natural person ("data subject"). An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural, or social identity of that natural person |
| Residence period | The period of time between the end of business and the end of purpose (EoP) for a data set during which the data remains in the database and can be used in case of subsequent processes related to the original purpose. At the end of the longest configured residence period, the data is blocked or deleted. The residence period is part of the overall retention period. |
| Retention period | The period of time between the end of the last business activity involving a specific object (for example, a business partner) and the deletion of the corresponding data, subject to applicable laws. The retention period is a combination of the residence period and the blocking period. |
| Sensitive personal data | A category of personal data that usually includes the following type of information: <ul style="list-style-type: none"> • Special categories of personal data, such as data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, genetic data, biometric data, data concerning health or sex life or sexual orientation, or personal data concerning bank and credit accounts. • Personal data subject to professional secrecy • Personal data relating to criminal or administrative offenses • Personal data concerning insurances and bank or credit card accounts |
| Where-used check (WUC) | A process designed to ensure data integrity in the case of potential blocking of business partner data. An application's where-used check (WUC) determines if there is any dependent data for a certain business partner in the database. If dependent data exists, this means the data is still required for business activities. Therefore, the blocking of business partners referenced in the data is prevented. |

SAP HANA Approach to Data Protection

Many data protection requirements depend on how the business semantics or context of the data stored and processed in SAP HANA are understood.

i Note

Using capabilities to communicate with other data sources, SAP HANA may also be used to process data that is stored in other systems and accessed through virtual tables.

In SAP HANA installations, the business semantics of data are part of the application definition and implementation. SAP HANA provides the features for working with technical database objects, such as tables.

It is therefore the application that "knows", for example, which tables in the database contain sensitive personal data, or how business level objects, such as sales orders, are mapped to technical objects in the database. Applications built on top of SAP HANA need to make use of features provided by SAP HANA to implement compliance requirements for their specific use case.

i Note

This is also true for SAP HANA installations that include SAP HANA dynamic tiering.

SAP HANA provides a variety of security-related features to implement general security requirements that are also required for data protection and privacy:

| Aspect of Data Protection and Privacy | SAP HANA Feature | More Information |
|--|--|--|
| Access control | <p>Several features in SAP HANA provide access control:</p> <ul style="list-style-type: none"> • Authentication • Authorization • Data masking • Data anonymization • Data encryption: <ul style="list-style-type: none"> ◦ Data volume encryption ◦ Redo log encryption ◦ Backup encryption ◦ Client-side column encryption | <ul style="list-style-type: none"> • Section SAP HANA Authentication and Single Sign-On [page 92] • Section SAP HANA Authorization [page 119] • Section Data Masking [page 188] • Section SAP HANA Data Anonymization [page 197] • Section Data Storage Security in SAP HANA [page 204] |
| Access logging | Audit logging | Section Auditing Activity in SAP HANA Systems [page 240] |
| Transmission control/communication security | Support for encrypted communication on all internal and external channels | Section SAP HANA Network and Communication Security [page 35] |
| Availability control | <p>Several features in SAP HANA provide availability control:</p> <ul style="list-style-type: none"> • Backup and recovery • Storage replication • System replication • Service auto-restart • Host auto-failover | Section on availability and scalability in the <i>SAP HANA Administration Guide</i> |
| Separation by purpose | <p>Separation by purpose is subject to the organizational model implemented and must be applied as part of the authorization concept.</p> <p>Isolated data storage can be achieved in SAP HANA using:</p> <ul style="list-style-type: none"> • Database schemas protected using authorization • Tenant databases | Section Technical System Landscape [page 18] |

⚠ Caution

Database trace and dump files may potentially expose personal data, for example, a trace set to a very high trace level such as DEBUG. In addition, if you are using the capture and replay feature, captured and

preprocessed workload files may also contain personal data since they contain input parameters for SQL statement execution. For more information, see the section on the security risks of trace, dump, and captured workload files.

⚠ Caution

The extent to which data protection is ensured depends on secure system operation. Network security, security note implementation, adequate logging of system changes, and appropriate usage of the system are the basic technical requirements for compliance with data privacy legislation and other legislation.

[Deletion of Personal Data \[page 273\]](#)

SAP HANA supports the deletion of data in tables using SQL deletion commands. Applications running on SAP HANA must make use of such commands to implement deletion requirements of personal data.

Related Information

[SAP HANA Implementation Scenarios \[page 27\]](#)

[Security Risks of Trace, Dump, and Captured Workload Files \[page 274\]](#)

[SAP HANA Dynamic Tiering](#)

[Data Protection and Privacy in SAP HANA XS Advanced \[page 341\]](#)

[Data Protection and Privacy in SAP HANA Cockpit \[page 284\]](#)

[Data Protection in SAP HANA Database Explorer \[page 288\]](#)

13.1 Deletion of Personal Data

SAP HANA supports the deletion of data in tables using SQL deletion commands. Applications running on SAP HANA must make use of such commands to implement deletion requirements of personal data.

End of purpose checks, including implementation of legally required retention periods and data blocking, are managed by the application. Applications can implement data blocking using SAP HANA mechanisms such as authorization and table creation. For example, an application could transfer blocked data to separate database tables that are protected by special authorizations.

Once data has been deleted, the delete operation cannot be undone using SQL statements.

For more information about permanent deletion of data in system-versioned tables, please refer to 'Deleting Data' in the System-Versioned Tables section of the *SAP HANA Administration Guide*.

i Note

Following standard practice, deletion of personal data is not enforced in backups. Common practice is that deleted data disappears from backups following typical backup-rotation mechanisms. SAP HANA supports backup lifecycle management by providing functions for deleting backups according to time stamps.

14 Security Risks of Trace, Dump, and Captured Workload Files

In exceptional situations, the data output in trace and dump files may expose certain security-relevant data. If you are using the capture and replay feature, captured and preprocessed workload files may also contain personal data.

Trace and Dump Files

Trace files are used to troubleshoot problems in the SAP HANA database. Dump files containing useful information for error analysis may also be created. Under normal circumstances, security-relevant data is not written to the files. However, if the default configuration is changed, for example when a trace is activated with a high trace level in a support situation, query strings including WHERE clause restrictions are written to trace files, for example, the database trace file of the index server. Query result sets and information about users may be output.

i Note

Passwords are never output.

The following files may contain security-relevant data:

- Trace files generated through the activation of the following trace types:
 - SQL trace
 - Database trace, including user-specific and end-to-end traces
 - Expensive statement trace
 - Performance trace
- Dump files
 - Core dump files (for example, crash dump files)
The system generates these files automatically.
 - Runtime dump files
The generation of these files can be triggered using the command line tool `hdbcons`.

Captured and Preprocessed Workload Files

SAP HANA capture and replay allows you to capture the workload of a production system and to replay the captured workload on a target system. Workload is any change executed on the database using SQL.

During both the capturing process and the preprocessing step required to make a workload replayable, files are written to the file system (by default to the trace directory `$DIR_INSTANCE/<host name>/trace`). These files contain input parameters for SQL statement execution, which may contain personal data:

- Captured workload files (*.cpt)
- Preprocessed workload files (files in the sub-directory <SID>_<capture_id>)

iNote

The system privileges WORKLOAD CAPTURE ADMIN and WORKLOAD REPLAY ADMIN are required to capture workloads and preprocess captured workloads respectively.

15 Security of Further SAP HANA Components and Capabilities

Security information for some SAP HANA components, as well additional capabilities that may be installed in the SAP HANA system, is available separately.

| For security information about... | See... |
|---|--|
| SAP HANA platform lifecycle management | Security Aspects of SAP HANA Platform Lifecycle Management [page 277] |
| SAP HANA auto content | Security of SAP HANA Content [page 278] |
| SAP HANA cockpit | Security Aspects of SAP HANA Cockpit [page 282] |
| SAP HANA database explorer | Securing the SAP HANA Database Explorer [page 286] |
| SAP HANA extended application services, advanced model | Security for SAP HANA Extended Application Services, Advanced Model [page 293] |
| SAP Web IDE for SAP HANA | Security Aspects of SAP Web IDE for SAP HANA [page 348] |
| SAP HANA smart data access | Security Aspects of SAP HANA Smart Data Access [page 280] |
| File Loader | File Loader Guide for SAP HANA |
| SAP File Processing | SAP File Processing for SAP HANA |
| SAP HANA R integration | Security Aspects of SAP HANA R Integration [page 281] |
| SAP HANA Hadoop integration | SAP HANA Spark Controller Installation Guide |
| SAP Enterprise Architecture Designer | SAP Enterprise Architecture Designer, Edition for SAP HANA Security Guide |
| SAP HANA accelerator for SAP ASE | SAP HANA Accelerator for SAP ASE: Administration Guide |
| SAP HANA data warehousing foundation | SAP HANA data warehousing foundation administration guides available on SAP Help Portal at SAP HANA Data Warehousing Foundation |
| SAP HANA smart data integration and SAP HANA smart data quality | Security for SAP HANA Replication Technologies [page 290] and the Installation and Configuration Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality |
| SAP HANA remote data sync | SAP HANA Remote Data Sync: Security Guide |
| SAP HANA streaming analytics | SAP HANA Streaming Analytics: Security Guide |
| SAP HANA real-time replication with SAP Landscape Transformation Replication Server | Security for SAP HANA Replication Technologies [page 290] and SAP Landscape Transformation Replication Server Security Guide |

[Security Aspects of SAP HANA Platform Lifecycle Management \[page 277\]](#)

Security information for SAP HANA platform lifecycle management

[Security of SAP HANA Content \[page 278\]](#)

SAP HANA is delivered with a set of preinstalled software components implemented as SAP HANA Web applications, libraries, and configuration data. These components are developed on SAP HANA Extended Services (SAP HANA XS), classic model, and together with other configuration components are referred to as SAP HANA content.

[Security Aspects of SAP HANA Smart Data Access \[page 280\]](#)

Security information for SAP HANA smart data access

[Security Aspects of SAP HANA R Integration \[page 281\]](#)

Security information for the integration of the SAP HANA database with R

[Security Aspects of SAP HANA Cockpit \[page 282\]](#)

Security considerations for SAP HANA cockpit include user management, single sign-on and certificate management.

[Securing the SAP HANA Database Explorer \[page 286\]](#)

Security considerations for SAP HANA database explorer include authentication, authorization, and secured connections.

[Security for SAP HANA Replication Technologies \[page 290\]](#)

SAP HANA supports several replication technologies. Security features and considerations depend on the implemented technology.

Related Information

[Important Disclaimer for Features in SAP HANA Platform \[page 382\]](#)

15.1 Security Aspects of SAP HANA Platform Lifecycle Management

Security information for SAP HANA platform lifecycle management

The SAP HANA database lifecycle manager (HDBLCM) is used to install, configure, and update the components of SAP HANA. The components of SAP HANA can only be installed by certified hardware partners, or any person holding the required certification on validated hardware running an approved operating system.

Before the installation and update of SAP HANA software components, the authenticity and integrity of the software should be verified. For more information about how to do this, see the *SAP HANA Server Installation and Update Guide*.

During the installation process, the initial passwords of a number of standard users are specified. Once you receive SAP HANA, we recommend that you change these initial passwords. If you are changing system identifiers (host name, SID, or instance number), it is possible to change the system administrator (<sid>adm) password and the password of the SYSTEM database user of the system database at the same time.

⚠ Caution

Do not use the `SYSTEM` user for day-to-day activities. Instead, use this user to create dedicated database users for administrative tasks and to assign privileges to these users. It is recommended that you then deactivate the `SYSTEM` user.

i Note

The `SYSTEM` user is not required to update the SAP HANA database system; a lesser-privileged user can be created for this purpose. However, to upgrade SAP support package stacks, SAP enhancement packages and SAP systems using the Software Update Manager (SUM) and to install, migrate, and provision SAP systems using the Software Provisioning Manager (SWPM), the `SYSTEM` user **is required** and needs to be temporarily reactivated for the duration of the upgrade, installation, migration or provisioning.

SAP HANA platform lifecycle management tasks can be performed on multiple-host SAP HANA systems centrally, by running the SAP HANA database lifecycle manager (HDBLCM) from any worker host and using remote execution to replicate the call on all remaining SAP HANA system hosts. Otherwise, the platform LCM tasks can be executed first on a worker host, and then re-executed manually on each remaining host. This method is considered decentralized execution.

Related Information

[Predefined Users \[page 83\]](#)

[Deactivate the SYSTEM User \[page 90\]](#)

[Recent changes in the SAP HANA Technology certification program 2016](#)

15.2 Security of SAP HANA Content

SAP HANA is delivered with a set of preinstalled software components implemented as SAP HANA Web applications, libraries, and configuration data. These components are developed on SAP HANA Extended Services (SAP HANA XS), classic model, and together with other configuration components are referred to as SAP HANA content.

i Note

SAP HANA XS, classic and the SAP HANA repository are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.

Software components delivered as SAP HANA content are an integral part of the SAP HANA platform. They provide essential features for Web-based configuration, administration and monitoring, application lifecycle management, and supportability.

Installation and Update

SAP HANA content is contained in delivery units (DUs). DUs containing automated content are deployed after the core SAP HANA database engine is started up during platform installation or upgrade and every time a new logical SAP HANA database is created. During an upgrade of an SAP HANA platform instance, the software components are updated to the version residing on the installation medium. DUs containing non-automated content need to be manually imported into the SAP HANA repository by a system administrator. DUs containing non-automated content are also automatically updated during an upgrade of an SAP HANA platform instance. For more information importing DUs, see *Deploy a Delivery Unit Archive (*.tgz)* in the *SAP HANA Master Guide*.

Content Security

Several software components available as SAP HANA content are Web applications and are therefore intended to be accessed by users through a Web browser. Only authenticated SAP HANA database users who have been explicitly authorized to use these software components by a user administrator can access them from their Web browser. The privileges required to use a software component are contained within roles delivered with the component itself. No user has these roles initially, except the user `_SYS_REPO` (as the owner of all repository content).

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. 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.

Users are authenticated and authorization checks are performed by the standard authentication and authorization mechanisms implemented by SAP HANA XS classic.

It is therefore guaranteed that no functionality is provided to or exposed to any user after a plain installation or upgrade.

More Information

For a list of all software components installed as SAP HANA content, including a detailed description of their purpose and functional scope, see the section *Components Delivered as SAP HANA Content*. The roles required to use each component are also listed with information about which functionality is made available by which role.

Related Information

[Components Delivered as SAP HANA Content \[page 363\]](#)

[SAP Note 2465027](#) 

15.3 Security Aspects of SAP HANA Smart Data Access

Security information for SAP HANA smart data access

SAP HANA smart data access makes it possible to connect remote data sources and to present the data contained in these data sources as if from local SAP HANA tables. This can be used, for example, in SAP Business Warehouse installations running on SAP HANA to integrate data from remote data sources.

In SAP HANA, virtual tables are created to represent the tables in the remote data source. Using these virtual tables, joins can be executed between tables in SAP HANA and tables in the remote data source. The linked database feature allows DML queries on remote data sources without the need to first create virtual tables.

Connections to the remote data source can be authenticated as follows:

- By one technical user credential
In this case, all connections to the remote data source share one and the same credential for the data source.
- By multiple secondary SAP HANA user-specific credentials
In this case, there is one credential per user per data source.
- By a Kerberos SSO credential
In this case, connections to the remote source (SAP HANA remote sources only) are authenticated through Kerberos single sign-on (SSO).

All credentials are stored securely in SAP HANA's internal credential store.

Authorization to access data in the remote data source is determined by the privileges of the database user as standard. In SAP Business Warehouse (BW) scenarios, authorization is applied in the BW layer.

The following privileges are required to manage remote sources and virtual tables:

| To... | You Need... |
|--|---|
| Manage remote sources | System privileges CREATE REMOTE SOURCE, CREDENTIAL ADMIN |
| To create and manage virtual tables on a remote source | <ul style="list-style-type: none">• Object privilege CREATE VIRTUAL TABLE• Additional object level privileges (for example, INSERT, UPDATE, DELETE) for virtual tables owned by others |
| Use the linked database feature | System privilege LINKED DATABASE |

Related Information

[Secure Internal Credential Store \[page 217\]](#)

15.4 Security Aspects of SAP HANA R Integration

Security information for the integration of the SAP HANA database with R

R is an open source programming language and software environment for statistical computing and graphics. The integration of the SAP HANA database with R makes it possible to embed R code in the SAP HANA database context.

R and SAP HANA

SAP does not ship the R environment with the SAP HANA database, as R is open source and is available under the General Public License. SAP does not provide support for R. In order to use the SAP HANA integration with R, you need to download R from the open-source community and configure it. You also need Rserve, a TCP/IP server that allows other programs to use facilities of R without the need to initialize R or link against R library. For more information, see the *SAP HANA R Integration Guide*.

Security Considerations

Users require additional privileges to execute R procedures. To ensure that only authorized users and programs can connect to Rserve, SAP also recommends implementing user authentication for calls from SAP HANA to Rserve. For more information, see *Security for R* in the *SAP HANA R Integration Guide*.

Secure Communication with SAP HANA

SAP recommends securing the communication channel between SAP HANA and the Rserve server using the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol. In this scenario, the SAP HANA server is the SSL client and the Rserve server is the SSL server. Configuration is required on both the SAP HANA server and the Rserve server.

On the SAP HANA side, the parameters for secure external client communication in the `global.ini` file apply, and it is not necessary to configure any of these parameters explicitly.

This communication channel only supports the use of a truststore stored in the file system. Therefore, you import the certificate of the Rserve server into the trust store specified by the parameter `[communication] sslTrustStore`, that is `sapsrv.pse`.

For more information about the configuration on the Rserve side and establishing communication via an SSL/TLS connection, see *Set Up SSL/TLS from SAP HANA to Rserve* in the *SAP HANA R Integration Guide*.

Related Information

[Server-Side TLS/SSL Configuration Properties for External Communication \(JDBC/ODBC\) \[page 46\]](#)

15.5 Security Aspects of SAP HANA Cockpit

Security considerations for SAP HANA cockpit include user management, single sign-on and certificate management.

User Authentication

Several types of credentials are used within SAP HANA cockpit:

| Credential | Details |
|---------------|--|
| COCKPIT_ADMIN | The master user for the cockpit created during the installation process. The password for the cockpit administrator user is the master password established during the installation process. This master user is assigned all three administrator roles, and can therefore access all aspects of the Cockpit Manager, and can create users, register resources, and assign users and resources to resource groups. |
| Cockpit Users | <p>The business users with access to the cockpit.</p> <p>Each is assigned one or more roles:</p> <ul style="list-style-type: none">• The cockpit administrator role can access the <i>Cockpit Settings</i> section of the Cockpit Manager, where they can configure the cockpit.• The cockpit resource administrator role can access the <i>Registered Resource</i> and <i>Resource Groups</i> sections of the Cockpit Manager, where they can register resources, create resource groups, and assign cockpit users to resource groups.• The cockpit user administrator role can access the <i>Manage User</i> section of the Cockpit Manager, where they can create and manage cockpit users.• The cockpit user role can access the SAP HANA cockpit, where they can view the resources in the resource groups to which they have been granted access.• The cockpit power user role can access the SAP HANA cockpit and the <i>Registered Resource</i> section of the Cockpit Manager |

| Credential | Details |
|-----------------------------------|--|
| Technical User | <p>An application global per-resource set of database credentials for access to remote resources and necessary to regularly gather information such as state, status, and other generalized KPIs.</p> <p>The technical user is created outside of the cockpit, but the technical user must be specified when a resource is registered in the cockpit. The technical user requires the privileges necessary to perform registration and statistics gathering: at a minimum, CATALOG READ system privilege and SELECT permission on _SYS_STATISTICS catalog.</p> |
| Database User (User Remote Login) | <p>The per resource/per user set of database credentials for a remote resource used by the cockpit user to view more sensitive information, and to make changes within their roles as defined on that resource.</p> <p>Each cockpit user needs to provide database user credentials with the system privilege CATALOG READ and SELECT on _SYS_STATISTICS in order to drill down in the cockpit to the overview information for a specific resource. The cockpit securely encrypts and stores separate database credentials for each cockpit user, but you can clear and re-enter the credentials through the <i>Resource Directory</i> when you desire to do so.</p> |
| Single Sign-On (SSO) | <p>For any systems running version SAP HANA 2.0 SPS 01, or later, the cockpit resource administrator has the option to enable or enforce SSO. See <i>Setting Up Single Sign-On</i>.</p> |
| Operating System User | <p>A per resource set of credentials for accessing the SAP Control process (starting and stopping the resource, and restoring features). This is usually the <sid>adm account. The cockpit securely encrypts and stores these credentials, but you can clear and re-enter the credentials through the <i>Resource Directory</i> when you desire to do so.</p> |
| Internal Communication | Service-to-service authentication |
| SAP HANA Service Broker User | For application persistence using the application's SAP HANA express database |

Note

It's not possible by using the cockpit to create the technical user required to register a resource in the SAP HANA cockpit. You need to create this user and grant the minimum necessary authorization by using SQL as follows:

```
CREATE USER <username> PASSWORD <password> NO FORCE_FIRST_PASSWORD_CHANGE;
GRANT CATALOG READ to <username>;
GRANT SELECT on SCHEMA _SYS_STATISTICS to <username>
```

Network and Communication Security

The cockpit uses secure protocols on all client browser connections to HTTPS ports. Communication to SAP HANA databases uses JDBC, and may be secured by importing certificates into the cockpit. Additional communication is made to the remote hosts using a restful interface which also may be secured. You can use properly signed certificates for the cockpit's external ports as well. For more information about obtaining certificates, see *Certificate Management in SAP HANA* in the *SAP HANA Security Guide*.

In a large enterprise it's likely that you may generate internal certificates that are signed by an internal certificate signing authority. In this case, you could insert the single (root) certificate from the signing authority. Any certificates signed by that authority (such as HTTPS or JDBC certificates) are automatically trusted. However, in a default installation, the SAP HANA system generates a self-signed certificate. In this situation, if the certificate is not replaced by a correctly signed one, then that specific certificate should be imported in order to enable trust.

15.5.1 Data Protection and Privacy in SAP HANA Cockpit

SAP HANA cockpit provides tools you can use to conform to legal and business requirements for protecting personal data stored in the system.

Introduction

Data protection is associated with numerous legal requirements and privacy concerns. In addition to compliance with general data privacy regulation, it is necessary to consider compliance with industry-specific legislation in different countries. SAP provides specific features and functions to support compliance with regard to relevant legal requirements, including data protection. SAP does not give any advice on whether these features and functions are the best method to support company, industry, regional, or country-specific requirements. Furthermore, this information should not be taken as advice or a recommendation in regards to additional features that would be required in specific IT environments; decisions related to data protection must be made on a case-by-case basis, taking into consideration the given system landscape and the applicable legal requirements.

i Note

SAP does not provide legal advice in any form. SAP software supports data protection compliance by providing security features and specific data protection-relevant functions, such as simplified blocking and deletion of personal data. In many cases, compliance with applicable data protection and privacy laws will not be covered by a product feature. Definitions and other terms used in this document are not taken from a particular legal source.

Glossary

| Term | Definition |
|---------------|--|
| Consent | The action of the data subject confirming that the usage of his or her personal data shall be allowed for a given purpose. A consent functionality allows the storage of a consent record in relation to a specific purpose and shows if a data subject has granted, withdrawn, or denied consent. |
| Deletion | The irreversible destruction of personal data. |
| Personal data | Any information relating to an identified or identifiable natural person ("data subject"). An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural, or social identity of that natural person. |
| Purpose | A legal, contractual, or in other form justified reason for the processing of personal data. The assumption is that any purpose has an end that is usually already defined when the purpose starts. |

User Consent

SAP HANA cockpit stores only personal data entered by users; it never collects personal data without a user's knowledge.

Logging Read Access and Changes

SAP HANA cockpit provides tools for auditing access and changes to personal data stored in SAP HANA databases. For details, see *Auditing Database Activity*.

Information Report

SAP HANA cockpit does not store any personal data except what is entered (if anything) in the optional contact information for resources. This typically includes the name and e-mail address of the contact person, and you can see it in the resource registration.

Deletion of Personal Data

You can remove unneeded user accounts and resource contact information. See:

- *Edit Resource Settings, Including SSO*

- *Delete a Cockpit User or Revoke Cockpit Access*

15.6 Securing the SAP HANA Database Explorer

Security considerations for SAP HANA database explorer include authentication, authorization, and secured connections.

Authentication and authorization

Authentication and authorization for the database explorer is governed by the application in which it resides.

The database explorer is integrated into SAP HANA cockpit and SAP Web IDE for SAP HANA. To use the database explorer you must be a non-administrator user of one of these applications.

Access to registered cockpit resources

As a cockpit user, you can access any HDI container that exists in a space in which you have been assigned the SpaceDeveloper role.

As a cockpit user, you can access any cockpit resource that is assigned to a group of which you are a member.

Access to HDI containers in Web IDE

As a Web IDE user, you can access any HDI container that is configured to run on the same XS advanced server that Web IDE runs on, and that exists in a space in which you have been assigned the SpaceDeveloper role.

By default, an HDI container user is assigned a few basic database privileges. For example, the object owner (“#OO” user) is only assigned the CREATE ANY privilege on the container’s run-time schema (schema “TEST” for an HDI container “TEST”). To access database objects inside other database schemata or other HDI containers, and to be able to deploy synonyms into the HDI container that point to objects outside the container, or to be able to import catalog objects into an HDI container, you must grant the object owner additional privileges.

Saving SQL console information to your browser's local storage

If your browser supports saving content to local storage, then, for the duration of your session, the SQL content in your SQL console is saved to your browser. (Configure this behavior in your user preferences within the Web IDE.)

Exporting and importing catalog objects

When you export catalog objects, the content that is saved is not encrypted.

When you import catalog objects into a database, no automatic virus scan or content validation of the files is performed before they are imported. Malicious content can be imported. Use external tools to validate the content of the files and scan them for viruses before importing them.

Related Information

[Secure User Store \(hdbuserstore\) \[page 223\]](#)

[Known Security-Related Issues \[page 351\]](#)

15.6.1 Secure the SAP HANA Database Explorer from Web Socket Attacks

When the host name in the URL for the SAP Web IDE for SAP HANA or SAP HANA cockpit does not match the hostname of its XS advanced server, then you must provide the XS advanced server with the accepted host names to use when the XS advanced server verifies connections to the SAP HANA database explorer.

Prerequisites

You must have one of the following roles:

- OrgManager
- SpaceManager
- SpaceDeveloper

Procedure

1. Connect to the XS command line interface.

For example:

```
xs login -a https://myhostname.mycorporatesite.com:30030 -u XSA_ADMIN -p  
mypassword
```

2. Set the WEBSOCKET_ORIGIN environment variable for the XS advanced server to the list of acceptable host names. Use the set-env command and enclose the acceptable host names between square brackets.

On a Unix system, the value must be enclosed in single quotes. For example:

```
xs set-env hrtt-service WEBSOCKET_ORIGIN ["myhost1", "myhost2"]'
```

On Windows, the double quote characters must be escaped. For example:

```
xs set-env hrtt-service WEBSOCKET_ORIGIN ["myhost1\", \"myhost2\"]
```

3. Execute the following commands to restage the hrtt-service.

```
xs restage hrtt-service
```

- Execute the following commands to restart hrtt-service.

```
xs restart hrtt-service
```

15.6.2 Data Protection in SAP HANA Database Explorer

Data protection is associated with numerous legal requirements and privacy concerns. In addition to compliance with general data privacy regulation, it is necessary to consider compliance with industry-specific legislation in different countries. SAP provides specific features and functions to support compliance with regards to relevant legal requirements, including data protection. SAP does not give any advice on whether these features and functions are the best method to support company, industry, regional, or country-specific requirements. Furthermore, this information does not give any advice or recommendation in regards to additional features that would be required in particular IT environments; decisions related to data protection must be made on a case-by-case basis, under consideration of the given system landscape and the applicable legal requirements.

i Note

In the majority of cases, compliance with applicable data protection and privacy laws will not be covered by a product feature. SAP software supports data protection compliance by providing security features and specific data protection-relevant functions, such as simplified blocking and deletion of personal data. SAP does not provide legal advice in any form. Definitions and other terms used in this document are not taken from any given legal source.

Glossary

| Term | Definition |
|---------------|--|
| Consent | The action of the data subject confirming that the usage of his or her personal data shall be allowed for a given purpose. A consent functionality allows the storage of a consent record in relation to a specific purpose and shows if a data subject has granted, withdrawn, or denied consent. |
| Deletion | The irreversible destruction of personal data. |
| Personal data | Any information relating to an identified or identifiable natural person ("data subject"). An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural, or social identity of that natural person. |
| Purpose | A legal, contractual, or in other form justified reason for the processing of personal data. The assumption is that any purpose has an end that is usually already defined when the purpose starts. |

Collection of Personal Data

SAP HANA database explorer stores only personal data entered by users; it never collects personal data without a user's knowledge. The only personal information stored by the SAP HANA database explorer consists of the following:

| | |
|---|---|
| XSA user ID | The XSA user ID is received from the XSA or Cloud Foundry login page and is recorded in both the SAP HANA database explorer and in log files. The following information associated with the user ID is also stored in the SAP HANA database explorer: |
| User-Defined SQL Statements | User-defined SQL statements are associated with a specific user and may contain personal data. |
| Long-Running SQL Queries | These queries are associated with a specific user and may contain personal data. |
| User ID for Cloud Foundry Login When Connecting to the Database Explorer from WebIDE | This ID is retained only until WebIDE is restarted or upgraded. |
| Database Credentials | User credentials associated with a specific user for HANA databases are stored in the SAP HANA database explorer. |
| User Preferences | These preferences are associated with a specific user and are stored in the SAP HANA database explorer. |

Logging SAP HANA database explorer collects XSA logs (when running on premise), Cloud Foundry logs (when running on SCP), and audit logs. Audit logs can track read access and changes and may contain user-specific data. XSA logs are retained indefinitely by XSA. For more information about XSA logging, see the *SAP HANA Developer Guide for XS Advanced Model*. Cloud Foundry logs are retained for one week. Audit logs cannot be deleted.

By default, logging is set to an application-wide level that does not record any personal data. However, you can set logging to session level, which collects all SQL statements for the current session. These statements may contain personal data.

15.6.2.1 Deletion of Personal Data

Delete user-specific information stored by the SAP HANA database explorer.

Context

This procedure deletes the following personal data associated with the connected user ID:

- All database credentials
- All user-defined SQL statements

- All user preferences
- All background tasks
- All content in open SQL or MDX consoles

Procedure

1. Click the *Preferences* icon (🔧) and click *Database Explorer*.
2. Click *Remove all user data* then click *Yes*.

Results

Your user data is deleted and SAP HANA database explorer restarts.

15.7 Security for SAP HANA Replication Technologies

SAP HANA supports several replication technologies. Security features and considerations depend on the implemented technology.

SAP HANA Extraction-Transformation-Load (ETL) Data Services

The SAP HANA Extraction-Transformation-Load (ETL) data replication technology uses SAP Data Services (hereafter referred to as Data Services) to load the relevant business data from the source system (for example, SAP ERP) and replicate it to the target SAP HANA database. This method allows you to read the required business data at the application layer level. You deploy this method by defining data flows in Data Services and scheduling the replication jobs.

Since this method uses batch processing, it also enables data checks, transformations, synchronization with additional data providers, and the merging of data streams. The main components are the Data Services Designer, where you model the data flow, and the Data Services Job Server for the execution of the replication jobs. An additional repository is used to store the metadata and the job definitions.

Data Services relies on the Central Management Server (CMS) for authentication and security features. For information about the security features provided by the CMS, see the *SAP Data Services Administrator Guide* or the *Information platform services Administrator Guide*.

To ensure security for your Data Services environment, use a firewall to prevent unintended remote access to administrative functions. In a distributed installation, you need to configure your firewall so that the Data Services components are able to communicate with each other as needed. For information about configuring ports on your firewall, see your firewall documentation.

For more information about ETL data replication technology using the SAP Data Services database, see the security section in the *SAP Data Services Administrator Guide*.

SAP HANA Direct Extractor Connection (DXC)

By default, the SAP HANA Direct Extractor Connection technology is switched off. For more information about how to switch it on, see the *SAP HANA Direct Extractor Connection Implementation Guide*.

For secure communication, the SAP HANA Direct Extractor Connection technology uses the SSL protocol (HTTPS) based on the Internet Communication Manager (ICM), a component of the SAP NetWeaver Application Server.

Trigger-Based Data Replication using SAP LT (Landscape Transformation) Replication Server (SLT)

SAP Landscape Transformation replication server is a replication technology that provisions data from SAP systems to an SAP HANA environment.

When using a distributed system, you need to ensure that your data and processes support your business needs without allowing unauthorized access to critical information. User errors, negligence, or attempted manipulation of your system should not result in loss of information or processing time. These demands on security apply likewise to the trigger-based data replication using the SAP LT replication server.

The SAP LT replication server and the SAP source system use the user management and authentication mechanisms provided by the SAP NetWeaver platform, in particular the SAP NetWeaver Application Server. Therefore, the security recommendations and guidelines for user administration and authentication as described in the *SAP NetWeaver Application Server ABAP Security Guide* also apply to the SAP LT Replication Server and an SAP source system.

The SAP LT replication server and the SAP source system use the authorization concept provided by the SAP NetWeaver AS ABAP. Therefore, the recommendations and guidelines for authorizations as described in the *SAP NetWeaver Application Server ABAP Security Guide* also apply to the SAP LT replication server. In SAP NetWeaver, authorizations are assigned to users based on roles. For role maintenance, use the profile generator (transaction PFCG) on the AS ABAP.

SAP HANA Smart Data Integration

SAP HANA smart data integration is data provisioning technology that allows real-time change data capture and batch loading from any source into SAP HANA. Because smart data integration uses a Data Provisioning Agent that is installed on a separate system than the SAP HANA system to manage adapters that link a source to SAP HANA, care must be taken to ensure secure connections. Security recommendations, as well as guidelines for user administration and authentication, are described in the *Installation and Configuration Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality*.

Related Information

[SAP Data Services on SAP Help Portal](#)

[SAP Real-Time Replication on SAP Help Portal](#)

[SAP NetWeaver on SAP Help Portal](#)

[SAP HANA Smart Data Integration and SAP HANA Smart Data Quality on SAP Help Portal](#)

16 Security for SAP HANA Extended Application Services, Advanced Model

As an application platform, SAP HANA extended application services, advanced model, provides a comprehensive runtime environment, in which deployed applications may be run in a secure manner. Application developers are encouraged to make use of the available platform services such as the identity provider to protect critical data from unauthorized access.

This section of the *SAP HANA Security Guide* describes the security aspects of the SAP HANA XS advanced server infrastructure and covers the following main areas:

- The technical components and communication paths used by the SAP HANA XS advanced server
- User administration and authentication
- Authorization concepts such as organization and spaces, scopes and role collections, and the Controller role model
- Communication paths used by the SAP HANA XS advanced server infrastructure and the security mechanisms that apply
- Critical data that is managed by the SAP HANA XS advanced model infrastructure and the security mechanisms that apply
- Security aspects involved throughout the most widely-used processes within the SAP HANA XS, advanced model
- Audit log files that contain security-relevant information, so you can reproduce activities if a security breach does occur
- The development environment, SAP Web IDE for SAP HANA

i Note

SAP HANA XS advanced is fully based on the SAP HANA platform. Therefore, information in other sections of the *SAP HANA Security Guide* also applies.

→ Recommendation

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 runtime 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. For more information, see the *SAP HANA XS Advanced Migration Guide*.

Additional Documentation Resources

Further SAP HANA Guides

| SAP HANA Guide | Comment |
|--|---|
| SAP HANA Security Checklists and Recommendations | Overview of recommendations to help you operate and configure the SAP HANA XS advanced model runtime securely |
| SAP HANA Administration Guide | How to manage the various components of the SAP HANA XS advanced model runtime |
| SAP HANA Developer Guide for XS Advanced Model | How to build and deploy applications to run in the SAP HANA XS advanced model runtime |
| SAP HANA XS Advanced Migration Guide | How to migrate applications built for SAP HANA XS, classic model, to run in the SAP HANA XS, advanced model runtime environment |

Important SAP Notes

The following table lists important SAP Notes that apply to the security of SAP HANA XS advanced:

| SAP Note | Title | Comment |
|-------------------------|--|--|
| 2243019 | Providing SSL certificates for domains defined in SAP HANA extended application services, advanced model | How to upload certificates for domains defined in SAP HANA XS advanced |
| 2245631 | Domains and routing configuration for SAP HANA extended application services, advanced model | <ul style="list-style-type: none">How to enable hostname routing for SAP HANA XS advancedHow to expose only a single port for all applications and services |
| 2300943 | Enabling JDBC SSL encryption for SAP HANA extended application services, advanced model | How to enable JDBC SSL encryption for SAP HANA XS advanced applications or services |
| 2243156 | Secure user setup for SAP HANA extended application services, advanced model | How to set up SAP HANA XS advanced to run application processes and staging processes as different operating-system (OS) users on Unix |

For a complete list of additional security-relevant SAP Hot News and SAP Notes, see also SAP Support Portal at <https://launchpad.support.sap.com/#/securitynotes>.

Downloading XS Advanced from SAP Marketplace

SAP HANA Extended Application Services, advanced model, is available not only on the SAP HANA media but also as a separate component on SAP Marketplace. Users with the required S-User ID can download the latest

version of XS advanced component in the package `SAP_EXTENDED_APP_SERVICES_1` from the following location:

► [Service Marketplace](#) ► [Software Downloads \[Downloads\]](#) ► [SUPPORT PACKAGES & PATCHES](#) ► [By Alphabetical Index \(A-Z\)](#) ► [H](#) ► [SAP HANA PLATFORM EDITION](#) ⌵:

- ► [SAP HANA PLATFORM EDITION 1.0](#) ► [XS ADVANCED RUNTIME](#) ► [SAP EXTENDED APP SERVICES 1](#) ⌵
- ► [SAP HANA PLATFORM EDITION 2.0](#) ► [SAP EXTENDED APP SERVICES 1](#) ⌵

→ Tip

SAP HANA Extended Application Services, advanced model, is backwards compatible; you can provide access to new features by installing the latest version of the XS advanced component even on older versions of SAP HANA. To download the package `SAP_EXTENDED_APP_SERVICES_1`, see *SAP Software Download Center in Related Information* below.

[Technical System Landscape of SAP HANA XS Advanced \[page 296\]](#)

SAP HANA extended application services, advanced model (XS advanced for short) provides a comprehensive platform for the development and execution of micro-service oriented applications, taking advantage of SAP HANA's in-memory architecture and parallel execution capabilities.

[User Administration and Authentication in SAP HANA XS Advanced \[page 302\]](#)

Both applications and platform services require user information to perform operations on behalf of an end user. User information in this context covers both authentication and authorization. A user management service lets you control precisely the group of users that are allowed to use specific system services or applications, modify sensitive data, or even do global system configuration.

[Authorization in SAP HANA XS Advanced \[page 313\]](#)

XS advanced users can access system services or interact with hosted applications. Since you don't want all XS advanced users to be able to view or even modify all resources, an appropriate authorization concept is required to allow you to control precisely which resource entities may be read or edited by a specific user.

[Network and Communication Security with SAP HANA XS Advanced \[page 328\]](#)

Security mechanisms are applied to protect the communication paths used by the SAP HANA XS advanced server infrastructure. SAP provides network topology recommendations to restrict access at the network level.

[Data Storage Security \[page 336\]](#)

Security mechanisms are applied to protect critical data managed by the SAP HANA XS advanced model infrastructure.

[Security-Relevant Logging and Tracing \[page 338\]](#)

Auditing makes it possible to trace who has performed which kinds of operations in the XS advanced system and applications.

[Data Protection and Privacy in SAP HANA XS Advanced \[page 341\]](#)

As an application platform server, the SAP HANA extended application server, advanced model (XS advanced), provides operators with a selection of tools and functions to conform to the legal requirements of data protection concerning data processed by deployed applications, as well as the server infrastructure itself.

[Security Aspects of SAP Web IDE for SAP HANA \[page 348\]](#)

SAP Web IDE for SAP HANA (SAP Web IDE) is a browser-based integrated development environment for the development of SAP HANA-based applications. These applications are comprised of web-based or mobile UIs, business logic, and extensive SAP HANA data models.

Related Information

[SAP Software Download Center \(Logon required\)](#)

16.1 Technical System Landscape of SAP HANA XS Advanced

SAP HANA extended application services, advanced model (XS advanced for short) provides a comprehensive platform for the development and execution of micro-service oriented applications, taking advantage of SAP HANA's in-memory architecture and parallel execution capabilities.

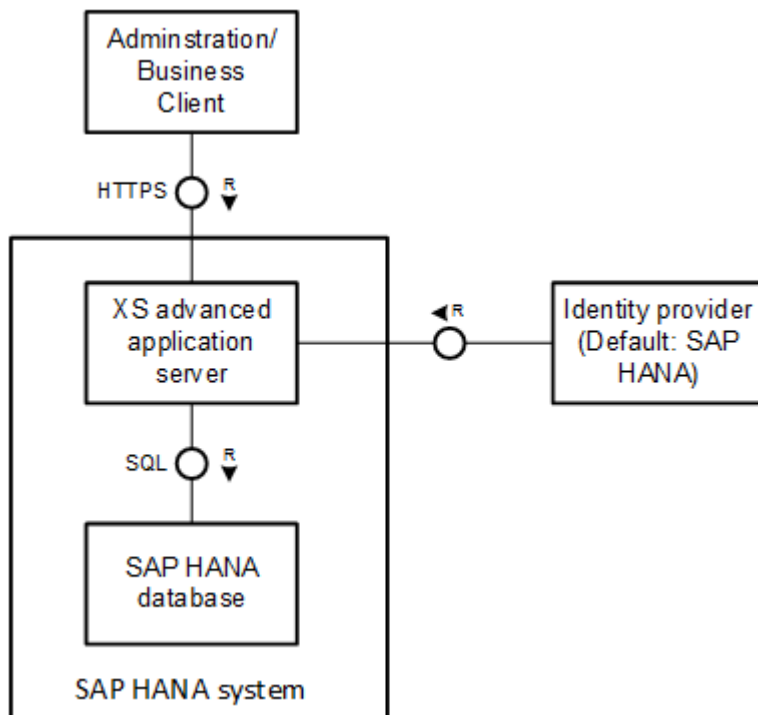
About SAP HANA XS Advanced

SAP HANA XS advanced offers a rich set of embedded services that enable an end-to-end support for web-based applications including lightweight web servers, persistency services, and a configurable identity provider. Furthermore, the platform supports polyglot application development with a core set of pre-deployed runtimes that are accepted as industry standard, for example, node.js or JavaEE.

Although the built-in runtimes come with first-class development and monitoring support, the platform has an open architecture that allows you to add custom runtimes. This high flexibility makes it essential that you put a strong focus on security concepts, not only when configuring and setting up the infrastructure, but also throughout operating the system.

Architecture Overview

As illustrated in the following diagram, the basic system architecture has a classic 3-tier approach:



3-Tier Architecture of SAP HANA with XS Advanced

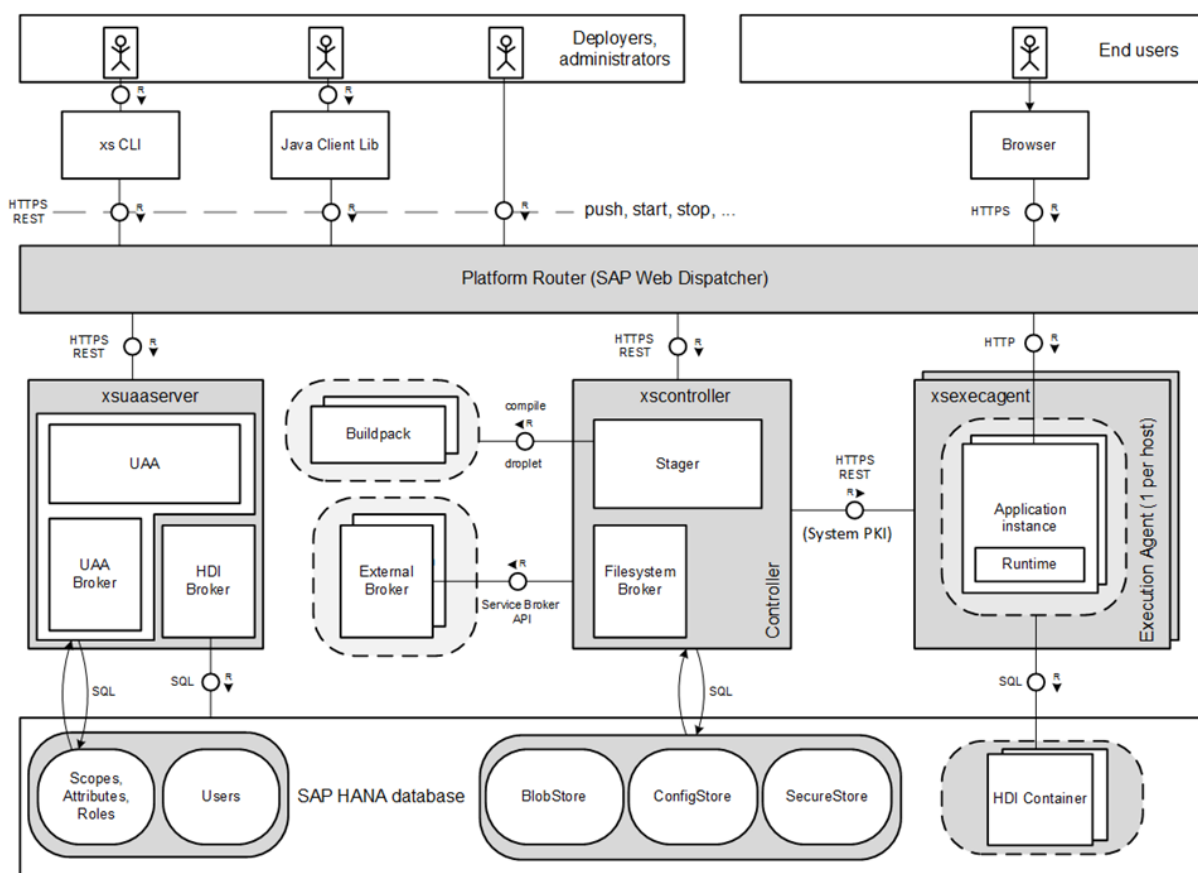
There is a distinction between the overall SAP HANA system and the SAP HANA XS advanced application server. The SAP HANA system refers to the entire SAP HANA platform part of the integrated solution. The SAP HANA XS advanced application server describes only the runtime platform as an integral part of the solution. All services of the SAP HANA system share the same system identifiers (that is, instance number and SID) and are controlled by the `hdbdaemon` service.

The third tier, represented by an SAP HANA database, provides persistency services, that is, data storage. In contrast, the application server components in the middle tier are responsible for deploying, running, and monitoring the applications. Most security-related features such as authentication, authorization, and auditing are primarily enforced in this layer. End users interact on the client layer with system or business users that are authenticated by an identity provider (IdP), which is SAP HANA user management by default. However, both the server components and the applications themselves access the SAP HANA database only through technical database users that the platform generates implicitly. Direct access to the database is only intended for database administration purposes.

⚠ Caution

As the XS advanced application server is based on the SAP HANA database, security-related configuration settings in the database also have direct effects on the SAP XS advanced application server. For example, if you configure JDBC connections not to support TLS/SSL (which is not the default), application artifact and runtime data could be compromised when transferred between applications and database.

The following diagram provides a more detailed overview of the technical system landscape of the XS advanced application server. All relevant components and storages used by the application server layer are highlighted with a gray background.



Technical System Landscape of XS Advanced Application Server

The XS advanced application server relies on the following SAP HANA services contributing to the integrated platform solution:

1. `xscontroller` (Controller, Filesystem Broker, Platform Router)
2. `xsexecagent` (Execution Agent)
3. `xsuaaserver` (UAA, UAA Broker and HDI Broker)

The exact functions of these services are explained in the section *Application Server Components*. These services are configured and administrated with the same tools that are already available for other SAP HANA services, for example, the `hdbsql` or `sapcontrol` command line tools, or the SAP HANA studio. Be aware that all SAP HANA services share the same administrative `<sid>adm` user at operating system (OS) level.

→ Recommendation

Due to the fact that the `<sid>adm` has the role of the system super user at the OS level and thus is enabled to access all critical data, it is strongly recommended to limit the number of people who know these credentials.

Related Information

[Application Server Components \[page 299\]](#)

16.1.1 Application Server Components

The XS advanced application server comprises the SAP HANA services `xscontroller`, `xsexecagent`, and `xsuaaserver` services, which are complemented by the Platform Router.

The services `xscontroller` and `xsuaaserver` run on a dedicated host of the system referred to as the XS advanced master host, which is not necessarily the master host for the database. The Platform Router, which is responsible for processing external requests, is managed by the `xscontroller` service and thus always runs on the XS advanced master host.

The execution agent is capable of running application instances on a host where the underlying `xsexecagent` service is started. To deploy an application, at least one execution agent is necessary. But in general, application instances may be scattered on different hosts of a distributed system.

xscontroller Service

The `xscontroller` service provides the central HTTP/REST interface to deploy, run, and monitor web applications. Deploying an application (or more generally a multi-target application or MTA) to the platform consists of several consecutive steps starting with the upload of the application files to the controller. These design-time artifacts typically include various types of content such as code binaries, source files, configuration files, or static HTML content.

Staging (according to Cloud Foundry terminology) denotes the process of transforming the application files into an executable representation by adding an appropriate runtime environment with an integrated web server. This step is performed by the Stager, which has to choose a suitable buildpack to apply to the application files in an external process. For instance, a Java web application archive is placed in a Tomcat server environment. The result of this compilation is a droplet, which represents the executable application on the file system. Due to the fact that the platform can be enriched with third-party buildpacks that support arbitrary runtime containers (they may even be downloaded during staging from a GIT repository), staging is highly security relevant.

The Controller stores the compiled droplets in the BlobStore, which resides in the SAP HANA database. The BlobStore is optimized to store file contents in a very efficient manner. For application start-up, the controller needs to download droplets from BlobStore very quickly to serve the HTTP/REST endpoint of the chosen execution agent. Controller resources such as application or buildpack metadata are stored by the ConfigStore, which is also located in the controller's database schema.

As part of the deployment, applications may be bound to services offered by (external) service brokers. Administrators are free to register arbitrary service brokers that implement the standardized Service Broker API, but most use cases are covered by the system platform brokers for SAP HANA persistency (HDI Broker), user authorization (UAA Broker) and file storage (FileSystem Broker). To consume an offered service, an application has to be bound to the service and typically receives credentials to access the service. These credentials passed by the brokers are generally stored in an SAP HANA secure store.

Platform Router

The Platform Router, which is realized by an SAP Web Dispatcher instance, exposes the public endpoint for the entire system. The router is configured in a way that all application and public server endpoints are represented

by an external URL. External requests are routed to the appropriate back-end instance according to the internal routing table.

→ Recommendation

It is strongly recommended that you limit network access to your system in a way that only the Platform Router's endpoints are accessible from outside the system. This can be accomplished by means of network zones and firewalls. For more information, see the section *Network and Communication Security* in the SAP HANA Security Guide.

The Platform Router instance is managed by the `xscontroller` service.

xsexecagent Service

Execution Agents, established through the `xsexecagent` service, are primarily responsible for starting and stopping application instances in a well-defined environment. To be reachable for end users, launched application instances typically provide a public HTTP port. As a basic monitoring service, the availability of this endpoint is checked periodically by the Execution Agent. Instances that lose reachability are restarted automatically. Different instances of the same application do not necessarily run on the same host in a distributed system (only the concept of host pinning could enforce this). Execution Agents also ensure that application instances of different spaces are not visible to each other at the operating system (OS) layer, if spaces have different OS users attached. For more information, see *Organizations and Spaces*.

i Note

Even if application instances run on different OS users, they compete for common system resources like CPU, memory, and disk space by default. To isolate a set of applications, consider pinning the applications (or alternatively their space) to a dedicated host.

As part of the deployment process, applications may be bound to services. The resulting credentials, for example, for accessing a database schema, are added to the process environment of the application instance.

xsuaaserver Service

The `xsuaaserver` service bundles a set of additional server components to complete the platform offering:

- The **User Account and Authentication** service (UAA) is the central user management for all end users interacting either with applications or server components. These are referred to as XS advanced users. The UAA uses the OAuth2 protocol based on the exchange of access tokens (see *User Authentication*). XSA users are named SAP HANA database users by default, but they could also originate from an external identity provider (IdP).
- The **UAA Broker** helps applications to protect their services from unauthorized access. Its API is fully Service Broker API compliant. Its services are consumed at deployment time when application-specific authorizations are requested. The UAA Broker runs on the same HTTP server as the UAA.
- **HDI containers** provide application-specific data storage and the deployment infrastructure in SAP HANA. They can be created during binding of applications against services offered by the HDI Broker. To access

HDI containers, technical SAP HANA users are created and the bound applications receive the corresponding credentials. The HDI Broker runs on a dedicated HTTP server.

Related Information

[SAP HANA Network and Communication Security \[page 35\]](#)

[Organizations and Spaces \[page 314\]](#)

[User Authentication \[page 312\]](#)

16.1.2 Users and Clients

All end users that access XS advanced application server components or applications are called XS advanced users.

We can distinguish between three different types of XS advanced user:

- **Application users** are all end users who interact with applications hosted on the application server (employees, customers and so on)
- **Developers** are users who develop, deploy, or maintain applications on the platform server
- **Administrators** are users who are allowed to set up and change the configuration of the application server; for instance, they may add new buildpacks or upload custom SSL certificates

i Note

Administrators of the XS advanced application server cannot manage the lifecycle of the SAP system, that is they cannot install, configure, start or stop SAP HANA services. This is handled at the OS level.

An XS advanced user's identity generally has its source in the UAA instance. To access a back-end instance, they first have to be authorized at the UAA endpoint and fetch an OAuth2 access token. For instance, if a developer using the `xs` command-line tool wants to push an application, she first has to enter her credentials as the basis for UAA authentication. As a result, the client receives the signed access token. This contains not only the user's identity but also the set of granted privileges. Based on the user information in the token, the Controller performs an authorization check and rejects invalid requests. The same procedure applies to business application requests. Application users represent the vast majority of all users, interacting with deployed web applications from local browsers.

For more information about user management, see *User Administration and Authentication*.

i Note

Although XS advanced users are named SAP HANA users, both applications and server components access SAP HANA artifacts by means of technical users that are generated during the deployment process.

In addition to application requests initiated by users via a web browser, developers and administrators interact with the Controller's HTTP/REST interface. The `xs` command-line tool, which is installed in the `bin` directory of `/hana/shared/<SID>/xs`, provides a user-friendly way to accomplish typical tasks like listing deployed applications or uploading a custom SSL certificate. Similarly, the Controller API can be consumed

programmatically using the delivered Java client library within Java processes. In general, the server endpoints are intended to be consumed with remote clients not necessarily running on the same host.

Related Information

[User Administration and Authentication in SAP HANA XS Advanced \[page 302\]](#)

16.2 User Administration and Authentication in SAP HANA XS Advanced

Both applications and platform services require user information to perform operations on behalf of an end user. User information in this context covers both authentication and authorization. A user management service lets you control precisely the group of users that are allowed to use specific system services or applications, modify sensitive data, or even do global system configuration.

Related Information

[User Management \[page 302\]](#)

[Predefined XS Advanced Users \[page 305\]](#)

[Predefined Database Roles for XS Advanced \[page 310\]](#)

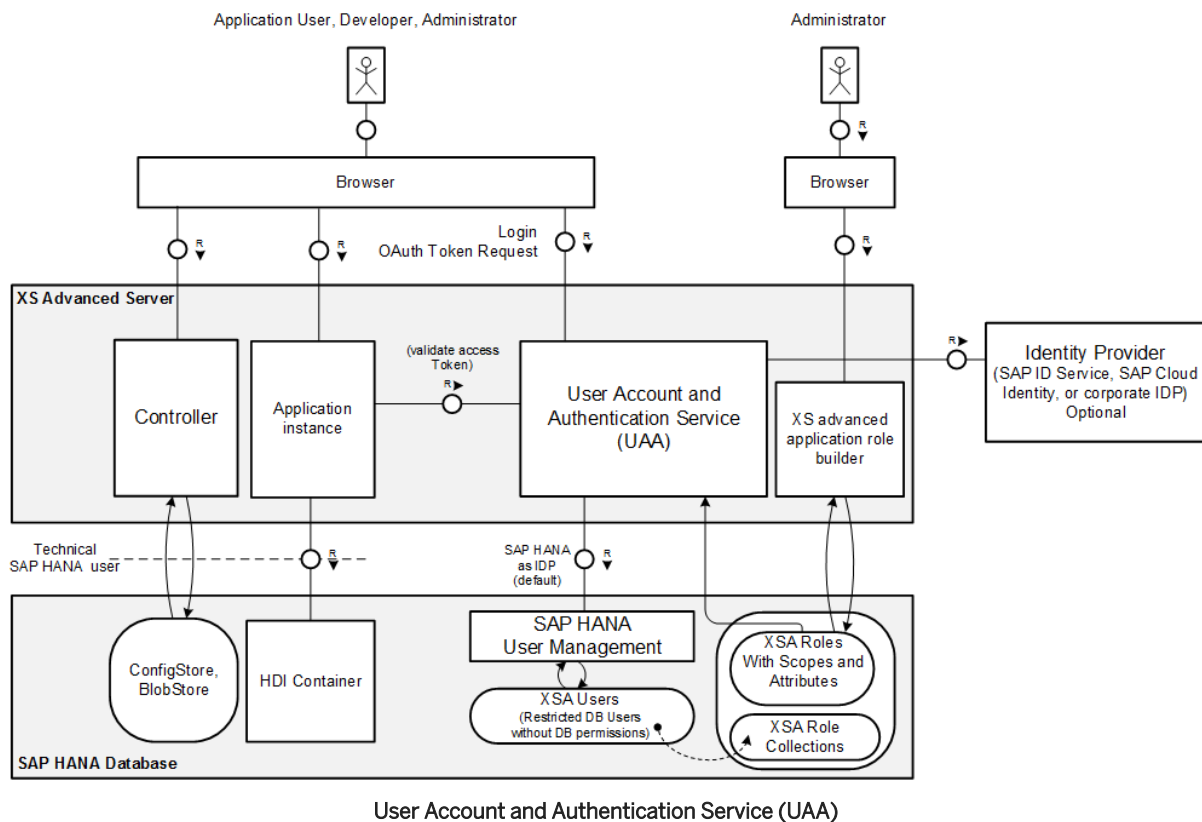
[User Authentication \[page 312\]](#)

[User Administration Tools \[page 313\]](#)

16.2.1 User Management

In traditional application servers, user information is kept in a local user store. In contrast, the SAP HANA XS advanced platform allows the use of SAP HANA as an identity provider (IdP) but enables the integration of an external IdP such as SAP ID Service or SAP Cloud Identity. Custom IdPs can also be configured, as long as they implement the SAML 2.0 standard. The XS advanced platform uses the underlying SAP HANA user store as IdP by default.

The **User Account and Authentication service (UAA)** represents the central platform service for user management and authentication, as depicted in the following diagram:



User information, such as first name, last name, user ID and user privileges, is provided in the form of signed OAuth2 access tokens the central UAA issues when a client logs in successfully. For more information about the authentication procedure, see the section on XS advanced user authentication.

XS Advanced User Categories

XS advanced users access the back-end instances typically through end-user interfaces such as a Web browser or command-line tools. Unlike technical users, application users and some types of system user can be also identified by personal data such as name, e-mail address, and so on. As the same identity provider is the basis for all of XS advanced users, an application user may also be granted developer privileges and the other way around.

XS advanced users who have their source in the SAP HANA user store (default) are typically restricted users with no access to SAP HANA database schemas. In contrast, applications and server components use **technical SAP HANA users** with certain access privileges. The platform passes these credentials to applications, enabling them to execute SQL statements, if the XS advanced user has sufficient privileges. Decoupling XS advanced users from technical users is the precondition for leveraging external IdPs, even though XS advanced users are also SAP HANA users by default. As technical SAP HANA users are generated by the platform in the background, you typically won't use them to interact with the system.

The XS Advanced User Account and Authentication (UAA) service provides the authentication end point for individual users who need to interact either with SAP HANA XS, advanced model, or with applications deployed

and running on the XS advanced model platform. Although such users are often referred to simply as **XS advanced users**, they can have the following roles and areas of responsibility:

- [Platform users \[page 304\]](#)
- [Application or business users \[page 304\]](#)
- [Operating-system users \[page 304\]](#)
- [Technical database users \[page 305\]](#)

XS Advanced Platform Users

Platform users are administrators or developers who are assigned to one or more specific organizations or spaces in the XS advanced platform. An XS advanced administrator user (for example, `XSA_ADMIN`) is allowed to perform any platform operation in any organization or space. However, it is also possible to maintain additional platform user **roles** and use them to restrict the type of access granted to certain users for particular organizations or spaces. XS advanced “administration users” are system users who manage the configuration of the XS advanced application server components, and in particular the XS Controller.

XS advanced platform users are SAP HANA users who have been assigned to a specific XS advanced role collection. Non-administrator platform users can also be managed by means of an external Identity Provider (IdP).

→ Tip

You can use the `xs` command-line interface to maintain XS advanced platform users. For more information, see *Maintaining Platform Users in XS Advanced with the XS CLI* in *Related Information* below.

XS Advanced Application (Business) User

Often referred to as “business users”, application users interact with application instances deployed to and running on the XS advanced run-time platform. Application users are also referred to as business users, for example, employees, customers, and so on.

Application users can be identified by personal data such as name or e-mail address, and this data along with other credentials are stored in a user store, for example, an Identity Provider (IdP); any request to log on to an XS advanced application is managed by the XS advanced User Account and Authentication service (UAA). Authorization scopes (defined in user roles) are granted to application users to restrict or enable access to particular data.

XS Advanced Operating-System Users

In the context of XS advanced, the following predefined operating system users are available by default:

- `<sid>adm`
Operating-system and administrative SAP HANA system user who owns all platform services as well as the system's file storage.
- `sap<sid>xsa`
Operating-system user required for staging and running applications in the pre-configured `SAP` space.
- `<sid>xsa`
Operating-system user required for staging and running applications in the pre-configured `PROD` space.

The `<SID>adm` operating system user exists to provide an operating system context. From the operating system perspective, the operating system administrator is the user that owns all SAP HANA files and all related operating system processes. Certain administration operations require the operating system user's credentials, for example, starting or stopping the system.

i Note

XS advanced application files are also owned by the *xsa operating-system users `sap<sid>xsa` and `<sid>xsa`.

XS Advanced Technical Database Users

A technical database user does not correspond to a real person and should be used for administrative tasks such as creating objects and granting privileges for a particular application. For example, an application server may log on to the SAP HANA database using a dedicated technical database user. In the context of XS advanced, technical SAP HANA users are generated by the platform in the background.

For XS advanced, the technical user `SYS_XS_RUNTIME` owns the XS Advanced Controller's SAP HANA schema, which contains the Blob Store, Config Store, and Secure Store. Similarly, the technical user `SYS_XS_UAA` owns the SAP HANA schema provided for the User Account and Authentication (UAA) for user management.

Additional technical database users are created on demand and as required for application-specific purposes. For example, the `SBSS_*` users are created as a result of an application-service binding. XS advanced also makes use of a number of `USR_*` users, too; `USR_*` users are created by the SAP HANA Service Broker for the service plans `schema`, `securestore`, and `sbss`. Similar to the predefined users created when binding an application to an HDI container, `USR_*` users are used by applications to access their schema. For more information, see the section on predefined users.

i Note

As of SAP HANA 2.0 SPS 03, the SAP HANA Service Broker no longer uses the `SBSS_*` prefix for HDI container users. Instead, these HDI container users have the name of the corresponding HDI container as the prefix. For example, for users created during service binding, the following format is used: `<HDI_Container_Name>_<GUID>_DT` (design-time access) or `<HDI_Container_Name>_<GUID>_RT` (run-time access). Binding users are assigned the role `PUBLIC` by default.

Related Information

[User Authentication \[page 312\]](#)

[Predefined XS Advanced Users \[page 305\]](#)

16.2.2 Predefined XS Advanced Users

The installation of the XS advanced application server creates a small set of predefined users that enable the operation of the underlying system.

The system's super user (`<sid>adm`) needs to be available in order to manage the life cycle of the system. Similarly, an administrative XS advanced system user (`XSA_ADMIN` by default) is necessary to perform the initial setup of the application server, for example, granting other users the privilege to create spaces in a dedicated organization and so on. Technical database users are created during installation for all server components that need to persist data in SAP HANA schemas.

The following predefined users exist:

- [Predefined XS Advanced System Users \[page 306\]](#)
- [Predefined Technical SAP HANA Users \[page 306\]](#)
- [Predefined OS Users \[page 309\]](#)

Predefined XS Advanced System Users

The table below lists the predefined XS advanced system users that are necessary for operating the XS advanced application server. First, an administrative user named `XSA_ADMIN` is required for the XS advanced Controller; this administrative user configures the application server at a global level. Non-administrative users of the XS advanced Controller are not allowed to perform administration tasks, for example, uploading custom certificates, adding custom buildpacks, or registering platform service URLs. Bear in mind that, although the credentials for the technical users for the SAP HANA Service Broker and UAA Broker are generated automatically during installation, the `XSA_ADMIN` user is created interactively with a user-defined password. As a first-level administrator user with irrevocable privileges, the `XSA_ADMIN` has unlimited access to the XS advanced Controller and therefore needs to be handled carefully.

→ Recommendation

- Keep the number of people with `XSA_ADMIN` credentials as small as possible. Delegate specific tasks like space management to lower-privileged users instead.
- Avoid creating other powerful users with privileges similar to `XSA_ADMIN`.
- Change the `XSA_ADMIN` password at regular intervals.

| User ID | User Type | Description |
|------------------------------------|------------------|--|
| <code>XSA_ADMIN</code> | XS advanced user | Administrative user for the XS advanced application server with unlimited access to XS advanced Controller API |
| <code>HDI_BROKER_CONTROLLER</code> | Technical user | Technical user for the SAP HANA Service Broker API |

Although the technical users in this table are created in the SAP HANA database, and database authentication checks are used to confirm the technical users' credentials, the technical users are not used to connect to the SAP HANA database.

Predefined Technical SAP HANA Users

Most of the server agents require a data store in the SAP HANA database and therefore need secure access to schemas. For this reason, a dedicated technical SAP HANA user is generated for each such schema, and the credentials of the technical SAP HANA user are passed to the server agent. As the management of technical users is performed at the infrastructure level, end users typically do not interact with these users. The technical

users listed in the following table are used to connect to the SAP HANA database with a specific set of conditions.

| User ID | Service | Type | Description |
|-------------------------------------|------------------|-------------------------|--|
| HDI_ADMIN_USER | SAP HANA Broker | Technical database user | Owens SAP HANA schema of SAP HANA Service Broker |
| HDI_BROKER_CONTROLLER | SAP HANA Broker | Technical database user | Has authorization to access the service broker API of SAP HANA broker |
| SYS_XS_HANA_BROKER | SAP HANA Broker | Technical database user | Owens the SAP HANA Service Broker's SAP HANA schema |
| SYS_XS_HANA_BROKER_INTERNAL | SAP HANA Broker | Technical database user | Has authorization to execute stored procedures for creating users, and so on. |
| SYS_XS_INSTANCE_MANAGER_ADMIN_USER | Instance Manager | Technical database user | Owens SAP HANA schema of the Instance Manager |
| SYS_XS_INSTANCE_MANAGER_BROKER_USER | Instance Manager | Technical database user | Has authorization to access service broker API of Instance Manager |
| SYS_XS_OID_USER | OIDC | Technical database user | Owens the SAP HANA schema for the OpenID Connect provider |
| SYS_XS_OID_USER_SEC | OIDC | Technical database user | Owens the SAP HANA secure store for the OpenID Connect provider |
| SYS_XS_RUNTIME | Controller | Technical database user | Owens the Controller's SAP HANA schema containing BlobStore, ConfigStore and SecureStore |
| SYS_XS_SBSS | SAP HANA Broker | Technical database user | Owens SAP HANA schema containing procedures to generate user passwords in a secure manner; used by the SAP HANA Service Broker |
| SYS_XS_SYSTEMDB_INFO | Controller | Technical database user | Has authorization to access database system catalog and configuration |

| User ID | Service | Type | Description |
|-------------------|-----------|--|---|
| SYS_XS_UAA | UAA | Technical database user | Owns the UAA's SAP HANA schema for user management |
| SYS_XS_UAA_SEC | UAA | Technical database user | Owns the UAA's SAP HANA secure store for user credentials |
| SYS_XSA | Installer | Owns SAP HANA schema containing a unique tenant ID | Owns SAP HANA schema containing a unique tenant ID |
| _SYS_DI | HDI | Technical database user | Owns all HDI SQL-based APIs, for example all API procedures in the <code>_SYS_DI</code> schema and API procedures in containers |
| _SYS_DI_*_CATALOG | HDI | Technical database user | Technical users used by the HDI to access database system catalog tables and views |
| _SYS_DI_SU | HDI | Technical database user | Technical superuser of the HDI created at installation time |
| _SYS_DI_TO | HDI | Technical database user | Owns transaction and connections of all internal HDI transactions |

Technical Users for HDI Schema-Based Containers

The deployment of database objects with SAP HANA Deployment Infrastructure (HDI) is based on a container model where each container corresponds roughly to a database schema. Each schema, and the database objects deployed into the schema, are owned by a dedicated technical database user.

For every container deployed, a new technical database user and schema with the same name as the container are created. Additional schemas and technical users required for metadata and deployment APIs are also created.

For example, for a container named `s`, HDI creates the following users:

- `S`:
The user who is the owner of the container schema `s`
- User `S#DI`:
The user who is the owner of the schema containing metadata and deployment APIs
- User `S#OO`:
The user who is the owner of database objects in schema `s`
- Users `_DI#S#METADATA_COM_SAP_HANA_DI_<metadata>`:
The users who are the owners of schemas containing build plug-in metadata

These technical users are used internally by HDI only. They are created as restricted database users who do not have any privileges by default (not even the role `PUBLIC`). They cannot be used to log on to the database.

For more information, see the section on maintaining HDI containers in the *SAP HANA Developer Guide (For SAP HANA XS Advanced Model)*.

Technical Users for Default Application Services

XS advanced applications can make use of a number of services managed by a service broker. To make use of a service, an instance of the service must be created and the application must be bound to the specified service instance. Several services are available by default; they are installed with the XS advanced run-time platform.

The installation of the following default application services results in the creation of a number of internal technical users:

- `Product-Installer`
Used for the installation and installation management of applications
- `Deploy-Service`
Used in the technical deployment of applications packaged in multi-target application (MTA) archives

The operation of binding these services to an application generates a technical user and random password according to the following naming convention `USR_<generated_ID>`. These technical users are required to make database schemas available for applications. For every combination of application and schema, such a technical user is created.

In addition, the `Job-Scheduler` service, used to create and schedule long-running operations in the XS advanced environment, uses an HDI container with the `SBSS_` prefix and a randomly generated name. The above-mentioned HDI schemas and users will be created for this container.

For more information, see *The SAP HANA XS Advanced Services: Deployment Infrastructure* in the *SAP HANA Developer Guide (For SAP HANA XS Advanced Model)*.

Predefined OS Users

Ultimately all platform services are made up of operating-system (OS) artifacts such as OS processes, network sockets, and file storage. Since operating systems come with their own user management features, these artifacts are out of necessity owned by OS users. Consequently, the XS advanced application server cannot be run without at least one OS user, although dedicated XS advanced users are able to perform the majority of the operational tasks.

The installation procedure creates the “super” OS user `<sid>adm` for the entire SAP HANA system. As the owner of all operating-system processes, the `<sid>adm` user is very powerful from a security perspective. For this reason, we strongly recommend that you limit the number of people with `<sid>adm` credentials as far as possible.

As described in the section *Application Server Components*, some platform services launch new processes at runtime:

- Execution Agents start application instances.
- The application “Stager” spawns processes running build packs during application staging.

In both cases, custom code comes to execution. If these processes ran as the system's `<sid>adm` user, the whole system could be compromised. To prevent this, the platform generally spawns external processes with OS users that are attached to the application's space. To support this approach, the initial setup includes OS user `<sid>xsa` user for the `PROD` space and OS user `sap<sid>xsa` for the SAP space. For more information about this isolation concept, see *Organizations and Spaces*.

The following table summarizes the OS users that are available immediately after installation:

| User ID | Type | Description |
|-------------|---------|---|
| <sid>adm | OS user | Administrative SAP HANA system user who owns all platform services as well as the system's file storage |
| <sid>xsa | OS user | OS user for staging and running applications in the pre-configured PROD space |
| sap<sid>xsa | OS user | OS user for staging and running applications in the pre-configured SAP space |

Related Information

[Application Server Components \[page 299\]](#)

[Organizations and Spaces \[page 314\]](#)

16.2.3 Predefined Database Roles for XS Advanced

Several predefined database roles are necessary for operating the XS advanced application server.

i Note

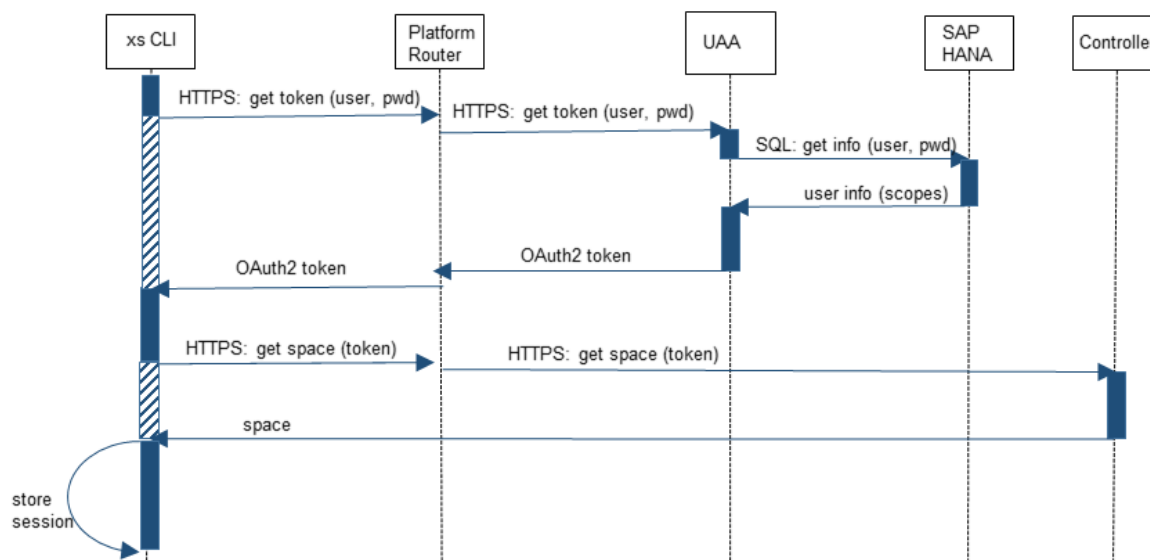
The following roles are SQL-based roles available in the catalog of the SAP HANA database.

| Role | Description |
|---------------------|--|
| _SYS_DI_OO_DEFAULTS | <p>This role contains the set of default privileges that are granted to all HDI container object owner users (<container>#OO users). SAP HANA Deployment Infrastructure (HDI) uses this role internally to grant default privileges instead of using the PUBLIC role. It contains only privileges to SYS views where additional security checks apply.</p> <p>The role contains SELECT privileges on the views: SYS.DUMMY, SYS.PROCEDURES, SYS.PROCEDURE_PARAMETERS, SYS.TABLES, SYS.TABLE_COLUMNS.</p> <p>This role is not intended to be granted to database users.</p> <div data-bbox="804 741 1393 853" style="background-color: #f0f0f0; padding: 5px;"> <p>i Note Do not extend this role in a production system.</p> </div> |
| SYS_XB_SBSS_VIEWER | <p>This role contains selected privileges for monitoring the status of the Service Broker Security Support (SBSS) component.</p> <p>The SBSS component provides service brokers with functions for creating, validating, and deleting the credentials they need for service bindings. Credential handling is achieved by creating restricted database users with secure random passwords.</p> <p>Specifically, this role contains read access to the SBSS component version table, in addition to read access to the SBSS bindings table that lists the credential names that have already been created with the SBSS API as well as some meta data for the bound credentials.</p> <p>This role is intended only for support users so they can query information such as SBSS version, number of credentials, names of services brokers that called the SBSS API.</p> <div data-bbox="804 1480 1393 1592" style="background-color: #f0f0f0; padding: 5px;"> <p>i Note This role does not grant access to any SBSS credentials.</p> </div> |

16.2.4 User Authentication

XS advanced user management is supported by a state-of-the-art user authentication strategy.

For technical SAP HANA users, the basic authentication mechanism applies as described in the SAP HANA Security Guide. In contrast, XS advanced users managed by the UAA are authenticated on the basis of the standardized OAuth2 protocol as depicted in the following sequence diagram:



OAuth2 Authentication Sequence Diagram

Using OAuth2 terminology, a client needs to fetch a protected resource from a resource server. Before being able to receive the resource from the server, the client sends a token request to the authorization server along with the user credentials. The authorization server checks the user credentials and composes the maximum set of privileges the user is granted. The user's identity, together with the authorization information, is encoded into a signed OAuth2 token, which is then sent back to the client. Now, the client can submit the resource server request with the attached access token. The resource server decodes the token (done offline without the authorization server), validates the user, and checks the privileges. If the privileges shown in the token allow access the resource, the server responds to the client request by sending the relevant resource. In the XS advanced application server infrastructure, the central UAA instance fulfils the role of authorization server. Application instances and the Controller are resource servers.

Note

Since providing a valid token at a server endpoint allows clients to access resources, tokens are never transferred unencrypted.

16.2.5 User Administration Tools

XS advanced users managed by the User Account and Authentication service (UAA) need to be administrated, in other words, users need to be created, updated, and also possibly deleted.

As the UAA service uses SAP HANA's user management mechanisms by default, all tools described in the SAP HANA documentation for managing SAP HANA users can be used, including:

- The `hdbsql` command-line tool
- SAP HANA cockpit
- SAP NetWeaver Identity Management

In addition, the SAP HANA XS advanced platform comes with the application *SAP HANA XS Advanced Cockpit*, which provides tools for performing all tasks related to user management in XS advanced. The `xs` command line interface (`xs` CLI) can also be used for some user management tasks. For more information, see the section on maintaining the SAP HANA XS advanced model runtime in the *SAP HANA Administration Guide*.

16.3 Authorization in SAP HANA XS Advanced

XS advanced users can access system services or interact with hosted applications. Since you don't want all XS advanced users to be able to view or even modify all resources, an appropriate authorization concept is required to allow you to control precisely which resource entities may be read or edited by a specific user.

In the XS advanced model, user permissions are derived from assigned roles. In addition, resources from different applications may be isolated by leveraging the concept of organizations and spaces. As the central entry point for system users, the Controller comes with a complementary role model. Various tools help you to define roles and assign them to users.

Related Information

[Organizations and Spaces \[page 314\]](#)

[Scopes, Attributes, and Role Collections \[page 320\]](#)

[Controller Role Model \[page 322\]](#)

[Authorization Management Tools \[page 326\]](#)

16.3.1 Organizations and Spaces

Resources from different applications may be isolated by leveraging the concept of organizations and spaces in combination with separated operating system users.

Introduction to Multi-Target Applications (MTAs)

A microservice-driven architecture of a solution is typically characterized by the cooperation of several service instances fulfilling dedicated task. Only by combining microservices can the solution meet its overall requirements. In the XS advanced context, several applications work closely together, forming what is referred to as a **multi-target application**, or MTA.

To illustrate, let's assume a simple MTA consists of two applications. A UI-based application needs to read database tables, which in turn are written by the other application. Consequently, both applications need exclusive access to the same database schema. The applications should also have a common authorization concept that applies to the same pool of end users. However, all other applications outside this MTA should be strongly isolated from the MTA's resources, in other words they should not be allowed to access the stored data nor the MTA's HTTP endpoints.

For more information about MTAs, see the *SAP HANA Developer Guide for SAP HANA XS Advanced*.

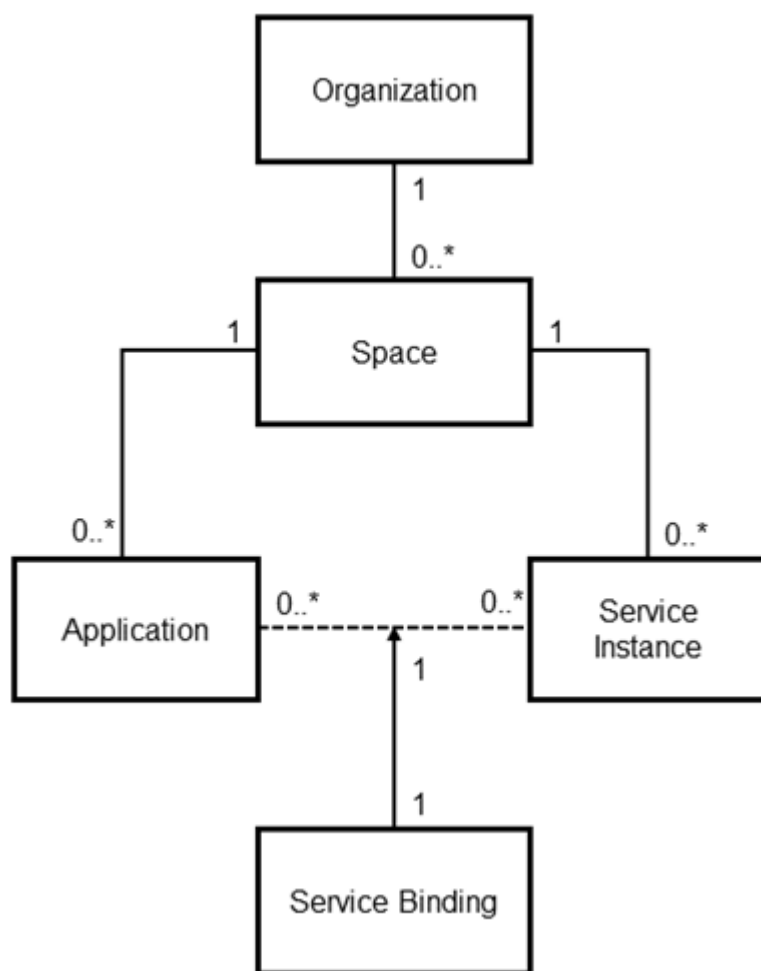
Controller Model

In general, you'll have applications that were deployed by the same Controller user, share the same set of resources, and are used by the same group of end users. This tight coupling of applications can be modeled by leveraging the central concept of **spaces** in the Controller model.

The main idea of spaces is that they form a kind of trust zone, which basically means that all applications deployed to the same space may share common resources like data storage and user authorizations and passwords. A space is intended to be shared by several developers, but developers may also have their own private space as well. Each application must be deployed to an existing space that has already been set up. Also service instances, provided by a service broker and typically representing a resource, can only be created within a space. An application in the space of this service instance may gain access to its resource by explicitly binding it to the service instance. The service binding entity then bears the credentials the service broker has issued during binding. The Execution Agent passes these credentials to the instances of bound applications by writing them to their process environment during start-up.

An **organization** may comprise several spaces. This helps to manage and administrate the spaces in a collective manner. For instance, an organization may group all spaces of a specific functional area of a company. In contrast to spaces, the organization of an application does not have an essential impact on the runtime behavior. There is only one exception: you can specify organization-specific domains that are the basis of the applications' external URLs in case of name-based routing.

The relationship between the involved model entities are represented in the following diagram:



Organizations and Spaces

All applications of an MTA are deployed to the same space, but other applications might be deployed there as well. Here, the Controller role model comes into play demanding deployment privileges for each single space. At the level of organizations, privileged Controller users with the role `OrgManager` are allowed to create new spaces within their organization and appoint other Controller users to be the manager of this specific space (for example, `SpaceManager`). Space Managers in turn may grant Controller users deployment privileges. For more information, see the section on controller role model.

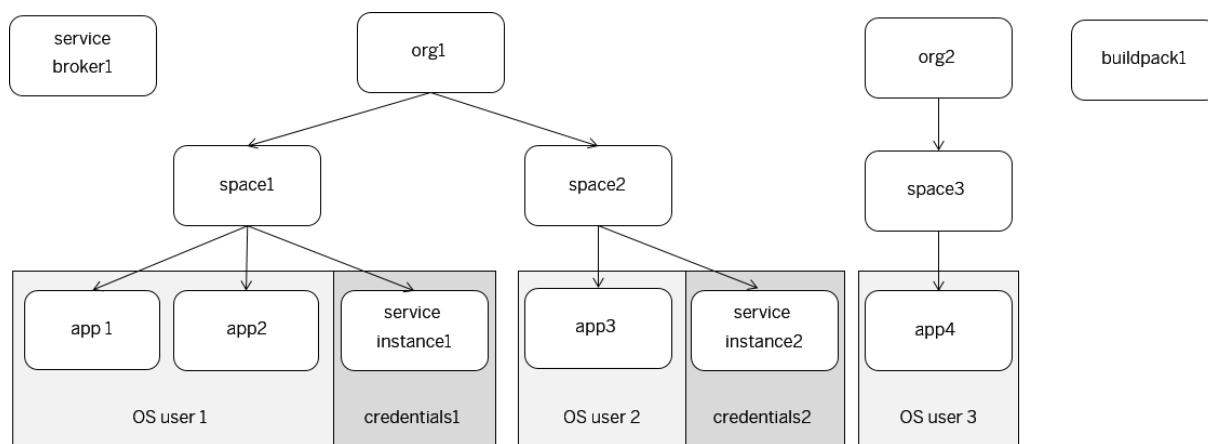
Spaces and Operating System (OS) Users

The isolation of spaces depends on different OS users. Spaces can be mapped to dedicated OS users, and only spaces running with different OS users are isolated from each other.

Applications running in the same space share all resources such as data storage, user authorizations, and passwords. Furthermore, the external buildpack process that is forked by the Stager run with this OS user.

Application instances and buildpacks in different spaces are only isolated at OS level if their space is running with a dedicated OS user. This is important from a security perspective when you consider that both types of OS processes run custom code.

The following figure shows a sample Controller model with regards to organizations, spaces, and applications and isolation on OS level.

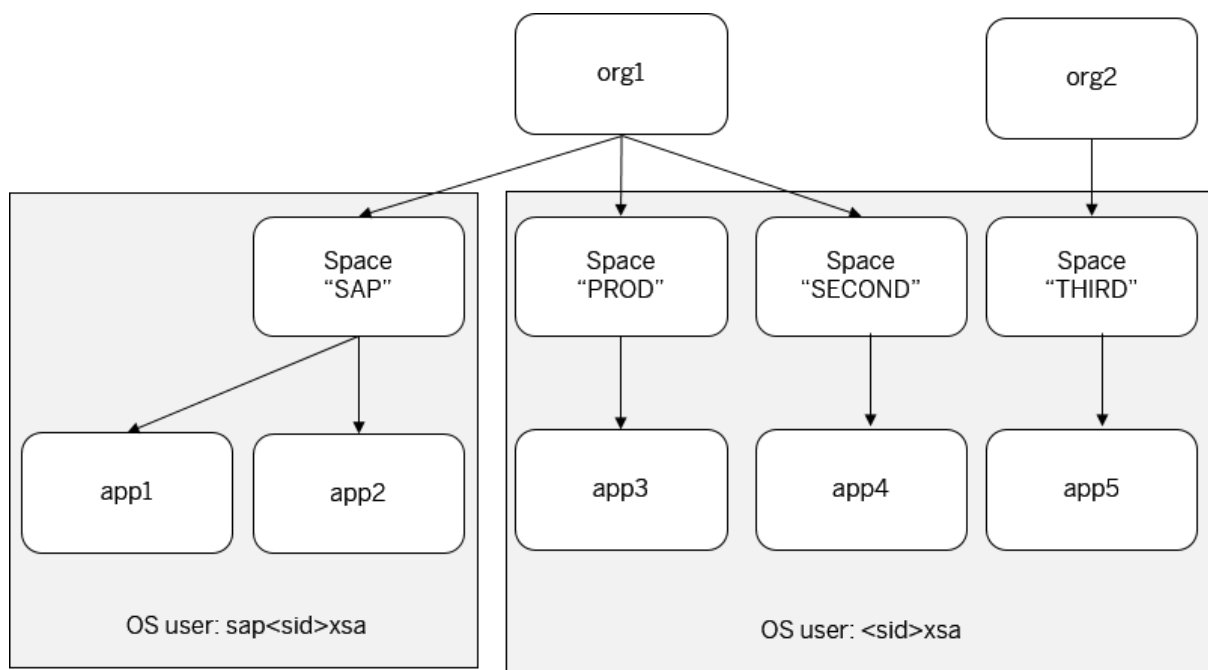


Example: Organizations and Spaces

Default Organizations and Spaces

The XS advanced application server comes by default with two spaces that are generated and mapped to dedicated OS users, and therefore isolated on OS level.

| Organization | Space | Space OS User | Content |
|----------------------|-------|---------------|--|
| Initial organization | SAP | sap<sid>xsa | Contains pre-installed SAP applications, for example: <ul style="list-style-type: none"> • Deploy-service for MTAs • Product-Installer and component-registry for application installation and update This space is an appropriate location for other system-relevant applications from SAP such as the optional administration UI or the Application Role Builder tool. |
| Initial organization | PROD | <sid>xsa | Empty after installation The space PROD can be used to deploy your custom applications. |



Space Isolation of Default Spaces

Custom Organizations and Spaces

By default, all spaces created by customers use the default pre-defined user `<sid>xsa`. This user is also assigned the initial default space named `PROD`.

i Note

In the default set-up, all applications in the initial space `PROD` and all additionally created spaces share the same OS user `<sid>xsa` and as a result, share resources such as data storage, user authorizations, and passwords on the OS level.

As `SpaceManager` you can configure space isolation by attaching a dedicated OS user to a newly created space. You can do this with the XS command line tool, either when you create the space or as a configuration step afterwards:

- `xs create-space <OS user name>`
- `xs update-space <OS user name>`

i Note

The changes only take effect on newly staged or restarted applications.

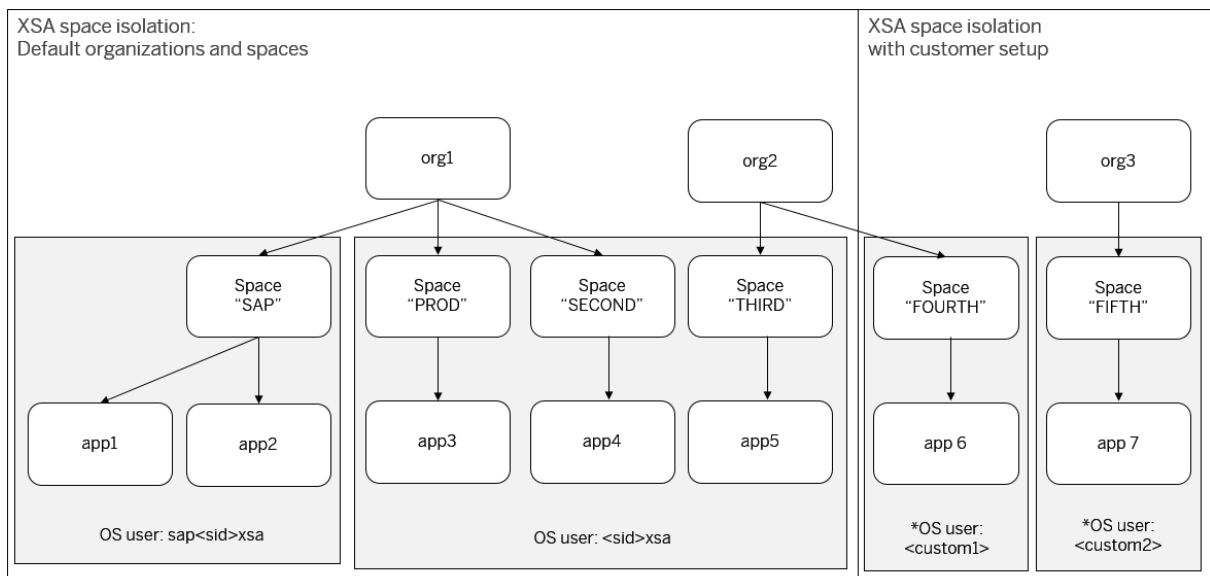
You can review the assign of spaces to OS users with the XS command `xs spaces`. All spaces mapped to the same OS user share resources.

The creation of OS users for space isolation is described in SAP Note 2243156. The OS user must be available on all hosts of the system that run services of the XS advanced application server, especially `xscontroller` and `xsexecagent` services.

i Note

For security reasons, you are not allowed to set the `<sid>adm` as a space OS user. Also be aware that the space OS user runs custom code on the executing host. So, restrict its privileges as much as possible.

The following figure shows a sample Controller model with regards to organizations, spaces, and applications and OS users, and the resulting space isolation.



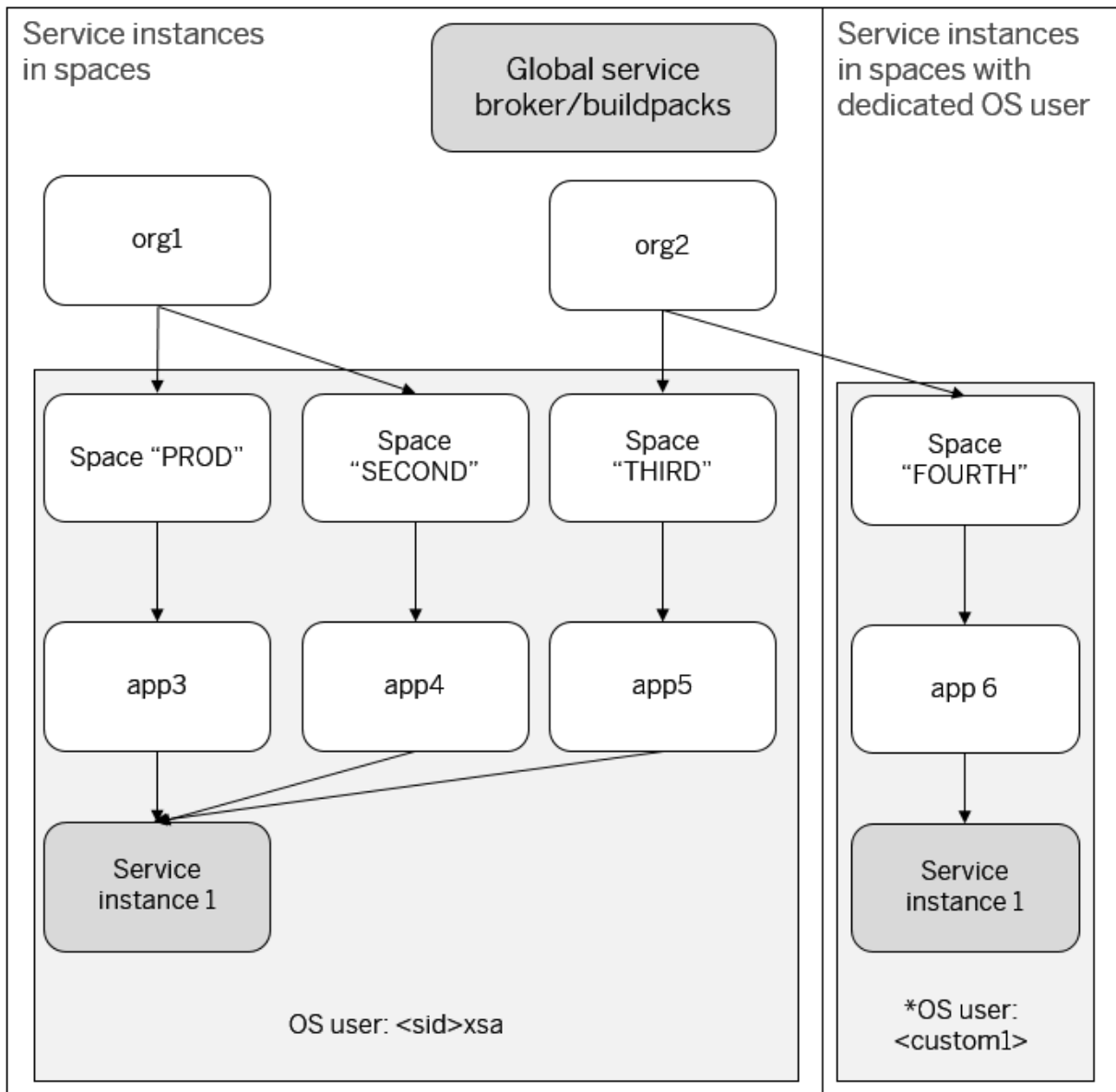
* OS user must be created and mapped to space.

Space Isolation with Custom OS Users

- Space SAP in organization org1 is running with the OS user `sap<sid>xsa` and therefore all applications running in the space SAP are isolated from other spaces not running with `sap<sid>xsa`. Applications app1 and app2 share resources such as data storage and passwords on the OS level.
- Spaces PROD, SECOND, and THIRD are all running with the same OS user `<sid>xsa` and applications app3, app4 and app5 share resources such as data storage and passwords on the OS level. Spaces are not isolated as they are mapped to the same OS user.
- Space FOURTH is mapped to a dedicated OS user `<custom1>` and therefore the space and application app6 is isolated from applications in other spaces. The OS user `<custom1>` must be created manually before it can be mapped to the space.
- Space FIFTH is also isolated as it too is running with a dedicated OS user `<custom2>`. Similarly, the OS user `<custom2>` must be created manually before it can be mapped to the space.

Service Instances in Spaces

Service instances are created per space, but for isolation they also depend on the OS user mapped to the space. Service instances can be reached by all applications running in spaces that share the same OS user. Service bindings in applications might contain credentials to service instances. Note that some entities like service brokers and buildpacks are created at a global model level and are shared by all spaces.



*OS user must be created and mapped to space.

Isolation of Service Instances

More Information

- For more information about how to assign Controller roles to Controller users, see the section *Controller Role Model*.
- For more information about how to create new organizations and spaces, see the *SAP HANA Administration Guide*.
- To isolate application instances with respect to resource consumption, make use of host pinning as described in the *SAP HANA Administration Guide*.

Related Information

[Controller Role Model \[page 322\]](#)

[SAP Note 2243156](#)

16.3.2 Scopes, Attributes, and Role Collections

Scopes define the actions that can be performed within a service. Attributes define the application's entities a user may access. A role collection is a list of scopes combined with a list of attributes.

The XS advanced authorization concept is based on the OAuth2 protocol, which requires users to pass an access token with each server request. This not only applies to business users who want to access the endpoints of deployed applications (to be more precise, they are redirected to UAA's login page to fetch the token), but also Controller users. Privileges to perform specific operations are associated with so-called **scopes**, which are simply represented by static strings defined by the resource servers (applications or server components like Controller). In other words, scopes define the actions that can be performed within a service.

Attributes, on the other hand, define the application's entities a user may access. A list of scopes combined with a list of attributes define a role, and a list of several roles define a **role collection**. Provided he or she has the proper privileges, an XS advanced user may be assigned several role collections. The key point is: An OAuth2 token issued by the UAA on a user request contains all scopes and attributes that are granted to the user based on his or her assigned role collections. On the basis of these scopes and attributes, an application can do an authorization check after having decoded the access token.

But where do the role collections actually come from?

MTA-specific role collections are design-time artifacts. The scopes and attributes they are based on are specified in the `xs-security.json` file, which is evaluated during the deployment of the MTA. The resulting role templates can be instantiated to roles and then grouped into role collections using the XS CLI role collection commands or the SAP XS Advanced cockpit. Finally, XS advanced users are assigned role collections, also using the XS CLI role collection commands or the SAP XS Advanced cockpit. For more information about how to handle application role collections, see the SAP HANA Developer Guide for SAP HANA XS Advanced. XS advanced users that need privileges to interact with system components (for instance the Controller) need to be assigned predefined role collections. These standard role-collections are created automatically during installation.

For more information about how to assign role collections to users, see *Authorization Management Tools*.

Standard Controller Role Collections

Users who interact with the Controller may trigger different types of operations, for example:

- Create, update, or view a space
- Create, update, or view an application in a space
- Create a service instances and bind an application to it
- Start, stop, or scale an application

- Add a custom buildpack
- Add an external service broker
- Upload custom SSL certificate
- Grant Controller roles to other Controller users

Some of the operations with a system-wide effect require administrative privileges (for example, uploading certificates). Others need to operate on the Controller's resources (applications, spaces and so on) with read or write permission. Therefore, the Controller defines three different scopes to cover these basic use cases:

- `cloud_controller.admin` (unlimited access)
- `cloud_controller.write` (write access)
- `cloud_controller.read` (read access)

In line with the authorization concept described above, these scopes are combined into three different Controller role collections:

- `XS_CONTROLLER_ADMIN` (`cloud_controller.admin` + `cloud_controller.write` + `cloud_controller.read`)
- `XS_CONTROLLER_USER` (`cloud_controller.write` + `cloud_controller.read`)
- `XS_CONTROLLER_AUDITOR` (`cloud_controller.read`)

To overcome the bootstrap problem when an XS advanced application server is installed, a single administrative Controller user (named `XSA_ADMIN` by default) is created. This user has the Controller role collection `XS_CONTROLLER_ADMIN`, which comprises all three Controller scopes. This means that the `XSA_ADMIN` can use the Controller without any restrictions and is in a position to do the initial setup of the model, that is appointing at least one OrgManager who is able to set up the spaces. Global resources like buildpacks or external brokers can also only be managed by an administrative Controller user.

→ Recommendation

After you have finished the initial setup of the system, deactivate the bootstrap administrative user `XSA_ADMIN` with the following SQL statement:

```
ALTER USER XSA_ADMIN DEACTIVATE USER NOW
```

In an emergency, a user with system privilege `USER ADMIN` can reactivate this user with the SQL statement:

```
ALTER USER XSA_ADMIN ACTIVATE USER NOW
```

The role collection `XS_CONTROLLER_USER` is designed for typical Controller users such as developers who work in one or more spaces (or even at organization level), reading and modifying their resources. Note that such users additionally need a so-called **Controller role** to gain access to a specific organization or space. If you want a user to have only read privileges, for example to audit some parts of the system, assign the role collection `XS_CONTROLLER_AUDITOR`.

i Note

Having the role collections `XS_CONTROLLER_USER` or `XS_CONTROLLER_AUDITOR` assigned is just the prerequisite for making a user a Controller user. As these role collections do not scope the Controller resources that the user may access, an additional Controller role is required to fill the gap (see *Controller Role Model*).

The following table gives an overview of all available standard role collections for the Controller, supplemented by the corresponding scopes and the permitted operations.

| Role Collection | Application | Scope(s) | Permitted Operations |
|-----------------------|-------------|---|---|
| XS_CONTROLLER_ADMIN | Controller | <ul style="list-style-type: none"> cloud_controller.admin cloud_controller.write cloud_controller.read | Unlimited access to Controller |
| XS_CONTROLLER_USER | Controller | <ul style="list-style-type: none"> cloud_controller.write cloud_controller.read | Read or write Controller resources (global resources excluded from modifications) |
| XS_CONTROLLER_AUDITOR | Controller | <ul style="list-style-type: none"> cloud_controller.read | Read access to Controller resources |

Related Information

[Authorization Management Tools \[page 326\]](#)

[Controller Role Model \[page 322\]](#)

16.3.3 Controller Role Model

Controller role collections are associated with essential authorizations like read and write permissions, but they do not control the permission to access a specific resource.

Resources in the Controller are entities such as:

- Applications
- Service instances and bindings
- Domains and routes
- Services and plans making up the marketplace of a service broker
- Spaces and organizations
- Service brokers
- Buildpacks

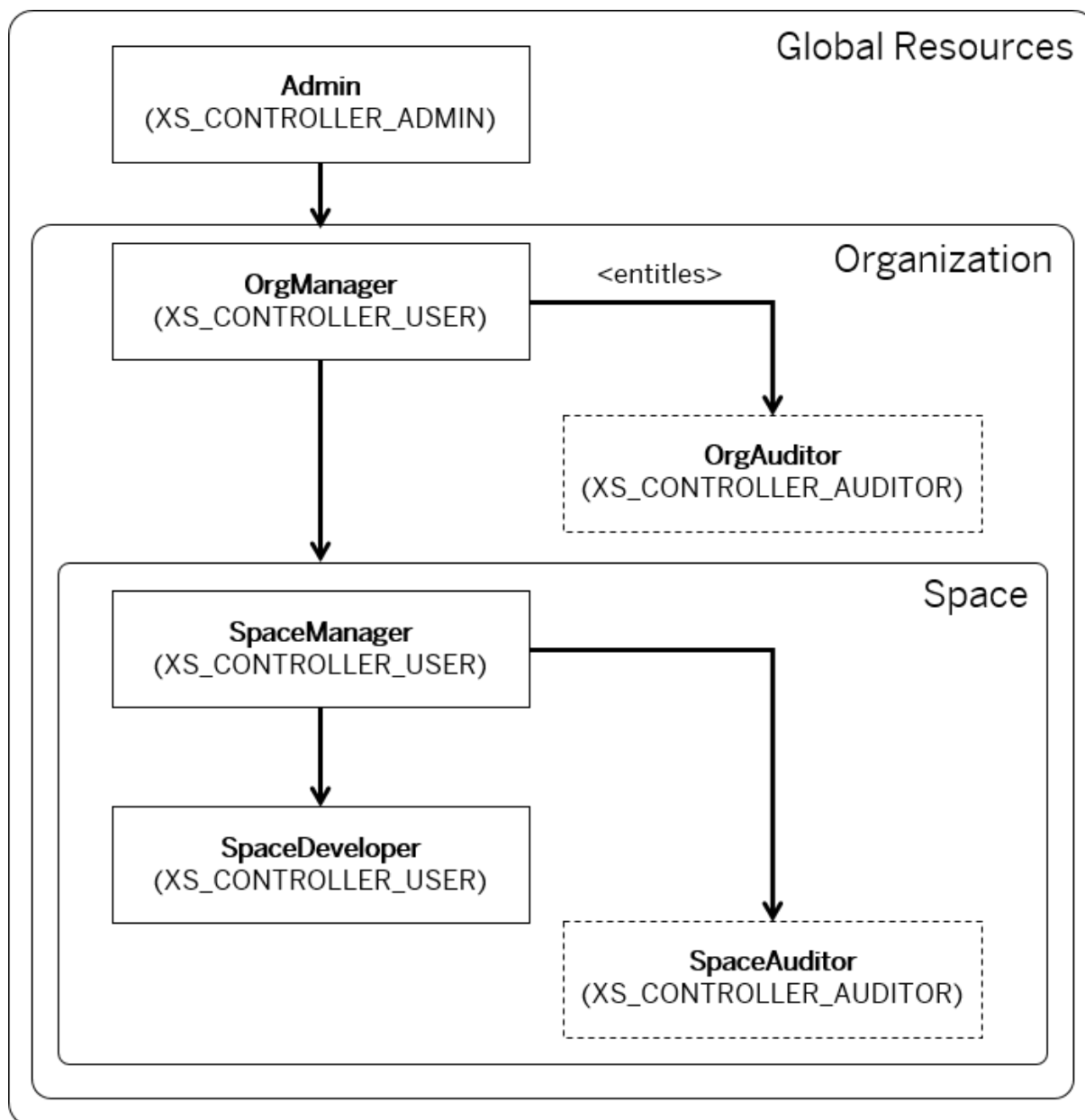
Spaces are a central concept for grouping applications that are tightly coupled and can run in a shared trust zone. Organizations simply embrace spaces at a higher level. Spaces not only determine the trust zones during runtime, but also provide a way to define the XS advanced users that should be allowed to manage the space's resources collectively. Which resources can be assigned to a space?

Each Controller resource has either a reference to a space (applications, service instances, bindings, and so on) and therefore is scoped to this space, or it is designated as a global resource (service broker, buildpacks and so

on). This is where Controller roles come into play. A Controller role is granted to a Controller user for a specific space or organization. Information about which roles are granted to a Controller user is not stored in the UAA, but attached to the space or organization entities in the Controller model. Five different categories of Controller roles are defined.

| Controller Role | Resource Scope | Permitted Operations Within Scope |
|-----------------|----------------|---|
| OrgManager | Organization | <ul style="list-style-type: none"> • Create a new space or update an existing space within the organization • Create or change a domain for the organization (not a shared domain) • View all Controller resources within the organization like spaces, apps and so on (including credentials) • Grant OrgManager, OrgAuditor, SpaceManager, SpaceDeveloper or SpaceAuditor to other platform users |
| OrgAuditor | Organization | <ul style="list-style-type: none"> • View all Controller resources within the organization like spaces, apps etc. (excluding credentials) |
| SpaceManager | Space | <ul style="list-style-type: none"> • Grant SpaceManager, SpaceDeveloper, SpaceAuditor to other platform users |
| SpaceDeveloper | Space | <ul style="list-style-type: none"> • Create or change resources within the space like apps, service instances or service bindings • Authorized to push an application to the space |
| SpaceAuditor | Space | <ul style="list-style-type: none"> • View the resources of a space like apps, application logs, application environments (excluding credentials) |

Controller roles are hierarchical as depicted in the following diagram (dotted lines denote read access only):



Controller Role Model

When a user submits a request to the Controller, his or her user ID and scopes are extracted from the OAuth2 access token as a first step. If a non-global resource is requested, the Controller then checks the user list for the resource's space (or organization).

Having extracted the Controller scopes and space or organization role, the Controller finally allows authorization according to following rules:

- Global resources without reference to a space or organization can only be modified by users with scope `cloud_controller.admin`. `cloud_controller.read` is sufficient for viewing global resources.
- A Controller user (that is a user with some Controller scope) who has been granted a Controller role to a specific space (or organization) may access all resources in this space (or organization) according to the assigned Controller role (for example, Space Developers may start an application in the space, Space

Auditors may only view this application). If the user has only `cloud_controller.read` scope, no resource may be modified.

The following list shows the scope of some resource types together with xs CLI commands.

| Controller Resource | Resource Scope | xs Commands | Minimum Authorization |
|--------------------------------|----------------|--|-----------------------------|
| Organization | Organization | <code>orgs, create-org, delete-org</code> | <code>admin</code> |
| Domain | Organization | <code>domains, create-domain, update-domain, ...</code> | <code>OrgManager</code> |
| User | Organization | <code>set-org-role, unset-org-role</code> | <code>OrgManager</code> |
| Space | Space | <code>create-space, update-space, ...</code> | <code>OrgManager</code> |
| Application | Space | <code>apps, push, scale, delete, start, stop, ...</code> | <code>SpaceDeveloper</code> |
| Route | Space | <code>create-route, delete-route, ...</code> | <code>SpaceDeveloper</code> |
| Service instance | Space | <code>create-service, delete-service, ...</code> | <code>SpaceDeveloper</code> |
| User-provided service instance | Space | <code>create-user-provided-service, delete-user-provided-service, ...</code> | <code>SpaceDeveloper</code> |
| Service key | Space | <code>create-service-key, ...</code> | <code>SpaceDeveloper</code> |
| Service binding | Space | <code>bind-service, unbind-service, ...</code> | <code>SpaceDeveloper</code> |
| User | Space | <code>set-space-role, unset-space-role</code> | <code>SpaceManager</code> |
| Buildpack | <global> | <code>create-buildpack, ...</code> | <code>admin</code> |
| Runtime | <global> | <code>create-runtime, ...</code> | <code>admin</code> |
| Service broker | <global> | <code>create-service-broker, ...</code> | <code>admin</code> |

| Controller Resource | Resource Scope | xs Commands | Minimum Authorization |
|---------------------|----------------|--|-----------------------|
| Service URL | <global> | register-service-url, unregister-service-url | admin |

For more information about xs commands, see *The XS Command-Line Interface Reference* in the SAP HANA Developer Guide (For SAP HANA XS Advanced Model).

16.3.4 Authorization Management Tools

Various command line and UI tools can be used for user and authorization management.

SAP HANA XS Advanced Cockpit

The SAP HANA XS Advanced cockpit is an XS Advanced application that is provided as additional content on the SAP HANA installation medium. Among other things, it provides a comfortable way to handle the following tasks:

| Task | Navigation |
|--|---|
| Create or delete users | User Management |
| Assign existing role-collections to users | User Management |
| Create custom role-collections based on deployed role-collection templates | ▸ Security ▸ Role Collections ▸ |
| Manage organizations and spaces | ▸ Organizations ▸ <organization>/<space> ▸ Members ▸ |
| Assign Controller roles (SpaceDeveloper, SpaceManager, etc.) to Controller users | User Management |

i Note

Users who need full access to the [User Management](#) tool need the role collection XS_USER_ADMIN. To view user settings only, XS_USER_DISPLAY is sufficient. Please note that the initial administrative user XSA_ADMIN has both role collections after installation.

For more information about the [XS Advanced Administration and Monitoring Tools](#), see the section on maintaining the SAP HANA XS advanced model runtime in the *SAP HANA Administration Guide*.

SAP HANA SQL Interface

If the UAA's identity provider is SAP HANA itself (default), the established SAP HANA user-management tools can be used to create XS advanced users for both business users and system users. The SQL statement `CREATE RESTRICTED USER <HANA_USER>` creates a user without initial privileges in SAP HANA. Restricted SAP HANA users are sufficient as these users won't access SAP HANA artifacts directly: applications access SAP HANA with technical users on XS advanced users' behalf.

You can make a standard SAP HANA user into a Controller user with the following SQL statement (assuming you have the corresponding privileges):

```
ALTER USER <HANA_USER> SET PARAMETER XS_RC_XS_CONTROLLER_USER='XS_CONTROLLER_USER'
```

You can revoke Controller role collections with:

```
ALTER USER <HANA_USER> CLEAR PARAMETER XS_RC_XS_CONTROLLER_USER
```

xs Command Line Client

This tool is available in each standard XS advanced installation and is located in the `/xs/bin` directory of the installation (`/hana/shared/<sid>/xs/bin/xs` by default). You cannot use this tool to create new XS advanced users, but you can use it to view and manage Controller roles of users that have already been granted Controller role collections. Controller users must first be created using another tool, preferably the [User Management](#) application.

The following table provides a list of all relevant xs client commands for user management:

| Command | Description |
|-------------------------------|--|
| <code>users</code> | List all users |
| <code>purge-users</code> | Delete all users in Controller who are not known to the User Account and Authentication (UAA) service [-f] |
| <code>space-users</code> | Show space users by role |
| <code>set-space-role</code> | Assign a space role to a user |
| <code>unset-space-role</code> | Revoke a space role from a user |
| <code>org-users</code> | Show organization users by role |
| <code>set-org-role</code> | Assign a organization role to a user |
| <code>unset-org-role</code> | Revoke a organization role from a user |
| <code>roles</code> | Display a list all existing application roles |
| <code>role-collections</code> | Display a list of all existing application role collections |

| Command | Description |
|--|--|
| <code>role-collection</code> | Display details of a specific application role collection |
| <code>create-role-collection</code> | Create a new application role collection |
| <code>update-role-collection</code> | Modify an existing application role collection |
| <code>assigned-role-collections</code> | Display a list of the application role collections currently assigned to a specific user |
| <code>assign-role-collection</code> | Assign an application role collections to a specific user |
| <code>unassign-role-collection</code> | Remove an assigned application role collection from a specific user |

Note

The `xs` command line client cannot be used to change the initial password of SAP HANA users. So, the first time you log on with a newly created SAP HANA user, you get a warning message demanding a password change. The password can be changed on the UAA login page. The URL of UAA's login page can be extracted with the command `xs version`.

For more information about `xs` commands, see the XS command-line interface reference in the *SAP HANA Developer Guide (For SAP HANA XS Advanced Model)*.

16.4 Network and Communication Security with SAP HANA XS Advanced

Security mechanisms are applied to protect the communication paths used by the SAP HANA XS advanced server infrastructure. SAP provides network topology recommendations to restrict access at the network level.

Related Information

[Security Areas \[page 329\]](#)

[Public Endpoints \[page 330\]](#)

[Single-Host Scenario \[page 331\]](#)

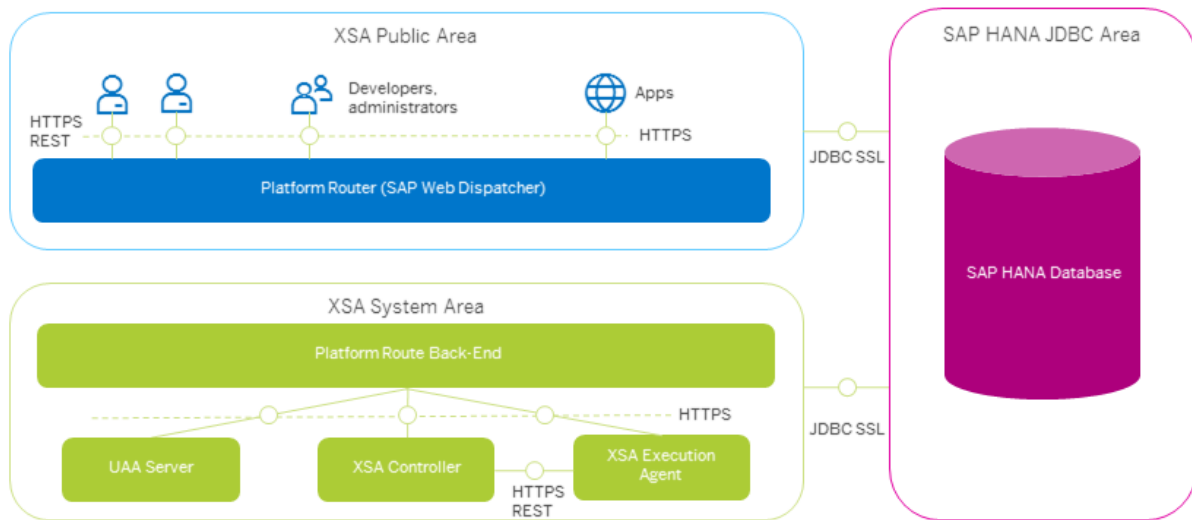
[Multiple-Host Scenario \[page 332\]](#)

[XS Advanced Certificate Management \[page 334\]](#)

16.4.1 Security Areas

Three different security areas can be identified in the XS advanced application server infrastructure. These cover communication channels that are all secured by TLS/SSL.

The different areas are characterized in the way how certificate management is done. As you can see in the figure below, the areas are referred to as the **XS Advanced Public** area, the **XS Advanced System** area, and the **SAP HANA JDBC** area.



XS Advanced Security Areas

The XS Advanced Public area is managed by an XS advanced administrator who configures the connections between clients and the public endpoints of applications and server components like the Controller or UAA. Inter-application communication is also related to this area because when another application is called, the request has to be sent to the public endpoint of the application exposed by the Platform Router. The XS advanced administrator is able to deploy custom SSL certificates for application domains or the system's administration domain. By default, the platform provides self-signed certificates for these endpoints. For more information about how this is accomplished, see *Certificate Management*.

The XS Advanced System area covers all internal communication channels between application server components like Controller, Execution Agents, UAA, Platform Router back-end, and so on. These channels are secured with TLS/SSL based on a system public key infrastructure (PKI), which provides mutual authentication. Note that the back-end of the Platform Router is also placed in the XS advanced system area so it can be managed by the XS advanced infrastructure for internal communication. The SSL certificates are only used for internal purposes and are never exposed to other areas, neither to applications nor to any XS advanced clients.

The SAP HANA JDBC area includes TLS/SSL-secured connections between the SAP HANA database and XS advanced applications, as well as between the database and server components.

⚠ Caution

The JDBC connection to the SAP HANA database is **not** encrypted by default. To activate JDBC TLS/SSL, custom SSL certificates need to be configured as described in section *Certificate Management*.

Related Information

[XS Advanced Certificate Management \[page 334\]](#)

16.4.2 Public Endpoints

The number of public endpoints directly depends on the configured routing mode.

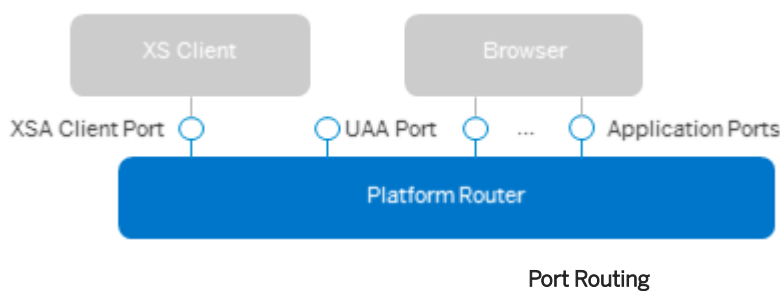
The XS advanced application server supports two routing modes: port routing and hostname routing. With port routing, the endpoints need an own port. In contrast, hostname routing requires only a single port but additional domain name server (DNS) entries. As URLs in the hostname routing scenario are user friendly and there is only a single public port, this mode is recommended for production usage.

→ Recommendation

To provide a high level of protection to your system, a firewall should reject all client requests against non-public ports.

Public Endpoints with Port Routing

With port routing, the endpoint's URLs are composed of the shared domain along with a dedicated port: `https://<shared-domain>:<port>`. Routing to the back-end component is solely based on this port. To avoid clashes with other systems, the port numbers are derived from the instance number of the system. Port routing is suitable if you don't want to or can't edit the DNS configuration. It works without additional manual effort and is therefore the installation default.



The table below shows the public ports of the Platform Router as they are exposed to clients.

| Endpoint | Protocol | Authentication | Port(s) |
|-----------------------|----------------|----------------|------------------|
| Controller (public) | HTTPS REST | Server | 3<instance_no>30 |
| UAA (public) | HTTPS REST/WEB | Server | 3<instance_no>32 |
| Applications (public) | HTTPS | Server | 51000 – 51500 |

⚠ Caution

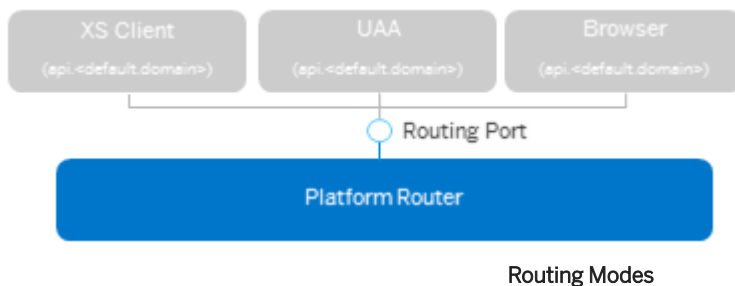
Some browser versions do not distinguish between different server ports in cookie handling. Port-based routing is therefore not recommended for use in production systems. Moreover, this mode occupies many more ports compared to hostname routing mode.

⚠ Caution

Some browsers are not able to distinguish between the cookies for different applications. Depending on the scenario, this might be a security issue.

Public Endpoints with Hostname Routing

With hostname routing, only the single public port 3<instance_no>33 of the Platform Router is required. In contrast to port routing, the routing to the internal back-end components is entirely based on the URL's host specification provided with the request. The host names are derived from the routes that are created during application deployment. A route contains the application's name by default, which is complemented by the default domain as suffix. Administrators may add custom shared domains, Org Managers are allowed to create domains for their specific organization.



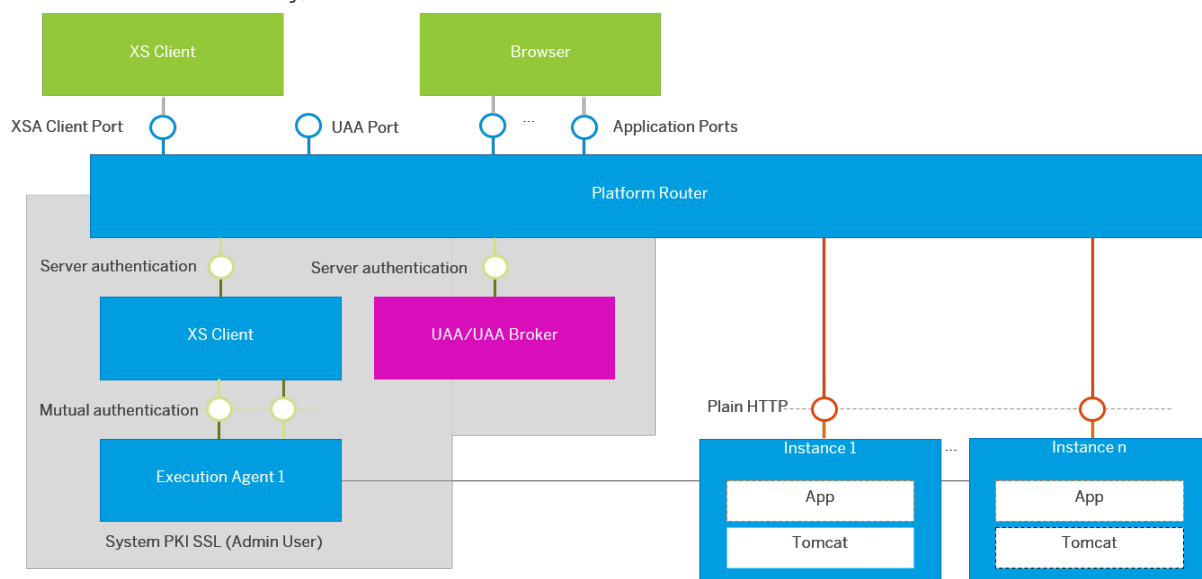
The following table lists the URLs of system components:

| Endpoint | Authentication | URL |
|-----------------------|----------------|--|
| Controller (public) | Server | <code>https://api.<default-domain>:3<instance_no>33</code> |
| UAA (public) | Server | <code>https://uaa-server.<default-domain>:3<instance_no>33"</code> |
| Applications (public) | Server | <code>https://<hostname>.<domain>:3<instance_no>33</code> |

16.4.3 Single-Host Scenario

In a single-host system, internal communication between the Platform Router back-end and the application instances is not secured. Since application instances are not part of the internal system public key

infrastructure (PKI), the Platform Router cannot authenticate them. However, given that the communication is bound to the local host only, the data transfer can be considered secure.



XS Advanced Single-Host Scenario

Private Endpoints

Server components and application instances expose ports that are not designed for external communication. The following tables lists all private ports:

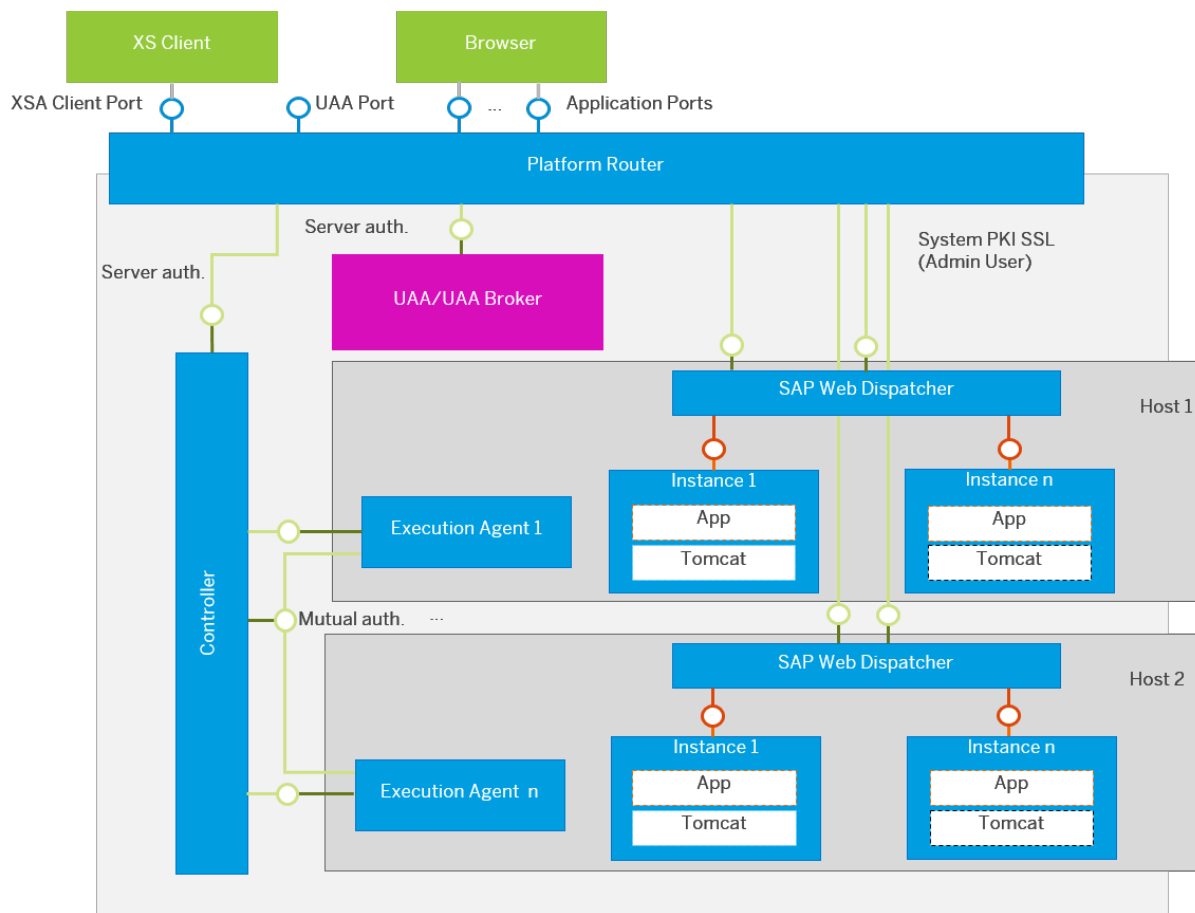
| Endpoint | Protocol | Authentication | Port(s) |
|----------------------------------|----------------|---------------------|-----------------------------------|
| Controller | HTTPS REST | Server (system PKI) | Within 51000 – 51500 (only local) |
| Controller (for Execution Agent) | HTTPS REST | Mutual (system PKI) | 3<instance_no>29 |
| Execution Agent(s) | HTTPS REST | Mutual (system PKI) | free |
| Application instances | HTTP | Client (OAuth2) | 50000 – 50999 |
| UAA | HTTPS REST/WEB | Server (system PKI) | 3<instance-no>31 (only local) |

16.4.4 Multiple-Host Scenario

In a multiple-host scenario, the Platform Router and the application instances typically run on different hosts. Due to the fact that the connection is based on HTTP, the data transferred to the application instances could be compromised. To solve this problem, each Execution Agent manages an additional SAP Web Dispatcher instance, which bridges the gap between the Platform Router host and the host of the application instance.

Since both the Platform Router and the SAP Web Dispatcher instances are integrated into the system public key infrastructure (PKI), the channel between them is protected.

The following diagram gives an overview of this setup:



XS Advanced Multiple-Host Scenario

As you can see above, there are slightly more private ports in use than in the single-host scenario. Especially the number of application ports needs to be doubled because each application port has to be exposed firstly, by the instance itself and secondly, by the additional SAP Web Dispatcher on each host with an Execution Agent.

Private Endpoints

In contrast to the single-host scenario, the internal port range for application is shared by the host router and application instances. Similarly to the single-port scenario, protecting private ports with an adequate firewall is strongly recommended.

| Endpoint | Protocol | Authentication | Port(s) |
|----------------------------------|----------------|-----------------|-----------------------------------|
| Controller | HTTPS REST | Server | Within 51000 – 51500 (only local) |
| Controller (for Execution Agent) | HTTPS REST | Mutual | 3<instance_no>29 |
| Execution Agent(s) | HTTPS REST | Mutual | free |
| Application instances | HTTP | Client (OAuth2) | 50000 – 50499 |
| Host Web Dispatcher | HTTPS | Server | 50500 – 50999 |
| UAA | HTTPS REST/WEB | Server | 3<instance_no>31 (only local) |

16.4.5 XS Advanced Certificate Management

The three security areas in the XS advanced server infrastructure have a slightly different certificate management. The Controller is the central instance that performs global certificate management, providing the necessary trust certificates for the corresponding components.

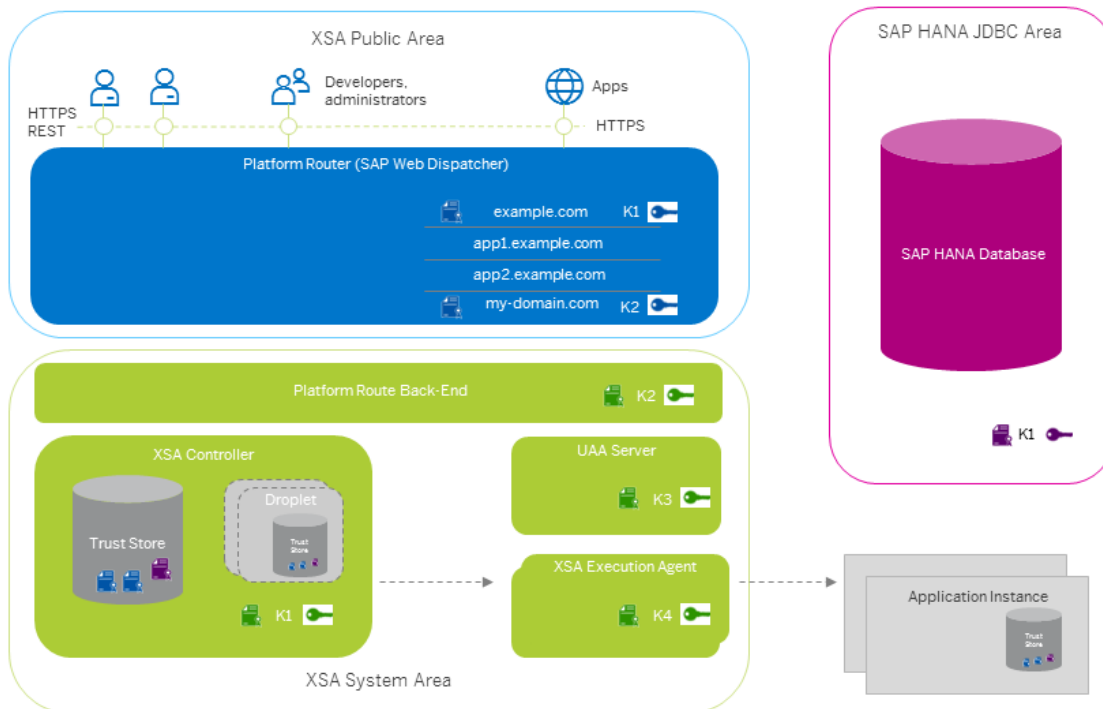
XS Advanced Public Area

In the XS advanced Public area the XS advanced administrator is responsible for deploying the domain-specific certificates. These can be either self-signed or issued by the global certificate authority (CA). The certificates can be deployed in the xs client using the `set-certificate` command. This is explained in detail in SAP Note 2243019. However, by default, the system generates self-signed certificates that the administrator can manually and securely distribute among the clients.

Note

In production XS advanced installations, replace the self-signed certificate with one trusted in your organization.

The distribution of private keys and their certificates in the XS advanced server environment is illustrated in the figure below. The Platform Router is totally managed by the Controller, which means that each time the administrator deploys a certificate for the specified domain (for example, by submitting `xs set-certificate DOMAIN -k KEY_FILE -c CERT_FILE`), the Controller adapts the Platform Router configuration accordingly. Due to this approach, the Controller is aware of all custom certificates and is therefore able to authenticate all external endpoints exposed by the Platform Router. On the other hand, the Controller uses its trust store for passing it to the application instances in order to allow them to authenticate the Platform router endpoints as well.



Certificate Management in XS Advanced Public Area

A security-critical scenario can arise if the certificate expires. In this case, the Controller cannot authenticate the Platform Router anymore and aborts start-up. To solve this problem, the administrator has to restart the Controller with option `--reset-certificate` and a new self-signed certificate is generated.

For more information about `xs` commands, see *The XS Command-Line Interface Reference* in the *SAP HANA Developer Guide (For SAP HANA XS Advanced Model)*.

XS Advanced System Area

The system administrator does not need to perform any configuration steps for the XS advanced system area. The internal system PKI is responsible for certificate management in this area. Each component within this infrastructure gets its own private certificate, which is signed by the root CA of the system. In this case, mutual authentication of these components is assured. This kind of certificate is never exposed to external clients.

XS Advanced JDBC Area

The XS advanced administrator is also responsible for the certificate management in the SAP HANA JDBC area. The XS advanced platform does not encrypt the JDBC connections to SAP HANA out of the box.

Therefore, custom certificates must be configured as explained in SAP Note 2300943. The following steps are required:

- Deployment of the custom certificate in the SAP HANA database in order to provide the certificate to all index server instances
- Publishing of the certificate to the XS advanced application server components (`xs trust-certificate ...`)
- Enabling the JDBC SSL in the platform settings and restart the system

More Information

For more information about:

- Setting up system-wide JDBC TLS/SSL connections, see SAP Note 2300943
- Setting up certificates in an XS advanced landscape with reverse proxy, see the section on installing XS advanced in the *SAP HANA Server Installation and Update Guide*
- Configuring allowed TLS versions and cipher suites for internal and external communication, see the section on configuring the XS Advanced Platform router in the *SAP HANA Administration Guide*

Related Information

[SAP Note 2243019](#)

[SAP Note 2300943](#)

16.5 Data Storage Security

Security mechanisms are applied to protect critical data managed by the SAP HANA XS advanced model infrastructure.

Most system components need to persist data provided by end users. The Controller stores application-related data (for example, its design time artifacts), as well as staged droplets ready for execution. Moreover, the Controller model, which is structured by organizations and spaces, may be modified by privileged users. Bindings to service instances typically bear credentials that also need special attention as they are not expected to be stored as plain text in SAP HANA tables. Similarly, the central UAA instance makes usage of a storage to which user information and credentials are written in a secure manner.

Finally, some system components, or to be more precise the standard service brokers, provide database and file system storage for applications at deployment time.

Related Information

[System Component Storage \[page 337\]](#)

[Application Storage \[page 338\]](#)

16.5.1 System Component Storage

Controller Storage

As part of deployment, application files are uploaded to the Controller and subsequently written to the **BlobStore**. The BlobStore is optimized to store file contents (Blobs) in an efficient manner, avoiding redundancy in cases where the very same file content is used in different contexts (for example, the same JAR file is shared by different Java-based applications). This file store is located in the SAP HANA schema, which is owned by the Controller's technical SAP HANA user `SYS_XS_RUNTIME`. When a resource is requested, only the Blobs that are referenced by this resource are available for download. As different Controller resources may share the same Blob in this store, a modification will result in a new Blob.

i Note

Only administrative Controller users (for example, `XSA_ADMIN` or users with role collection `XS_CONTROLLER_ADMIN`) have full BlobStore access. Some commands in the `xs` command line client therefore need administrative privileges.

Controller resources, such as application or buildpack metadata, are written to the ConfigStore, which also resides in the Controller's database schema. User requests that need to fetch data from ConfigStore are authorized by the mechanisms described in the section *Controller Role Model*.

The following sensitive Controller data is stored encrypted in SAP HANA secure store:

| Data | Content |
|--------------------------------|------------------------------------|
| Service broker | <code>broker credentials</code> |
| Service binding | <code>service credentials</code> |
| Service key | <code>service credentials</code> |
| User-provided service instance | <code>instance data</code> |
| Application environment | <code>environment variables</code> |

To access the API of an (external) service broker, for example when a new service instance is requested, basic authentication is required (`auth_username` and `auth_password`). Service binding entities keep the credentials to access the offered services created by a service broker. A prominent example is the credentials of the technical SAP HANA user for a HDI container that has been created by the HDI Broker. In the case of user-provided service instances, you may want to make explicit credentials available for applications.

UAA and UAA Broker Storage

Similar to Controller storage, information stored by the UAA or the UAA Broker is separated according to security relevance. User information, user secrets, and tokens are written encrypted to an SAP HANA secure

store with the technical user `SYS_XS_UAA_SEC`. However, common metadata like the scope, role, and attribute definitions are kept in standard SAP HANA tables in the schema of `SYS_XS_UAA`.

Related Information

[Controller Role Model \[page 322\]](#)

16.5.2 Application Storage

By default, applications may consume two different kinds of storage provided by the XS advanced application server: SAP HANA storage and file-system storage. Both are requested during application deployment when the HDI Broker or the FileSystem Broker are used.

HDI Broker

The HDI Broker offers different service plans to match various customer needs:

- `hdi-shared` provides a full HDI container with a technical user
- `schema` provides a plain SAP HANA schema with a technical user
- `securestore` provides an SAP HANA secure store to write encrypted data
- `sbss` provides an SAP HANA schema with procedures to generate secure passwords

FileSystem Broker

When a service instance of the FileSystem Broker is created (for example, by calling `xs create-service`) within a specific space, a new directory on a dedicated file system is created. This directory is configured with exclusive access rights for the OS user attached to the space of the service instance. In this way, it is guaranteed that the directory is only visible to applications within the same space (or to applications within a space having the same OS user).

16.6 Security-Relevant Logging and Tracing

Auditing makes it possible to trace who has performed which kinds of operations in the XS advanced system and applications.

System audit logs may help you to detect undesired modifications that could be the result of a misconfiguration in the user authorization setup. It could also uncover attempts to breach the system security.

For application auditing, the audit-log service writes audit log events received from applications to SAP HANA auditing.

16.6.1 Audited Operations

Several operations are automatically audited.

By default, the system logs all operations submitted to the Controller endpoint:

- Read operations for all Controller resources like applications, spaces, and buildpacks
- Update operations for all Controller resources
- Create and delete operations for all Controller resources
- Starting and stopping of application instances
- Settings like tracing, backup requests, SSL certificate uploads and so on

The UAA service additionally logs:

- Login activities
- Issuing of access tokens
- All UAA Broker operations, for example, creating a new service instance or binding an application
- Configurations like changing the identity service provider

For each requested operation, a new log line will appear in the audit log containing:

- The name of the user who triggered the operation
- The time stamp the operation was requested
- A short description of the operation containing the affected resources

16.6.2 Audit Trails

Audit logs are written to `<sid>adm` user's tracing file system.

| Service | Audit Log Location |
|-----------------|---|
| Controller | <code>/trace/hdbxscontroller_audit.log</code> |
| Execution Agent | <code>/trace/hdbxsexecagent_audit.log</code> |
| UAA | <code>/trace/uaa-audit.log</code> |

Audit trails can be easily inspected in the SAP HANA studio.

Syslog Support

The Linux operating system provides a comprehensive logging system called syslog. It is highly flexible, provides integration possibilities, and is therefore also provided for audit logs in the XS advanced system. In

order to make the system components write their audit logs to syslog, set the parameter `syslog` in the section `audit` of the Controller's ini file `xscontroller.ini` to `true`.

For more information about how to configure syslog, refer to the documentation of your operating system.

16.6.3 Application Auditing

Deployed applications can be configured to use the audit-log service of the SAP HANA XS advanced platform. The audit-log service writes audit log events received from applications to SAP HANA auditing.

To activate SAP HANA auditing for all events from an XS advanced application bound to the audit-log service, create an audit policy with the following audit actions on all objects:

- CONFIGURATION CHANGE
- PERSONAL DATA ACCESS
- PERSONAL DATA MODIFICATION
- SECURITY EVENT

You can create an audit policy using SQL as shown in the following example:

Sample Code

```
CREATE AUDIT POLICY xsa_policy AUDITING ALL SECURITY EVENT, PERSONAL DATA  
ACCESS, PERSONAL DATA MODIFICATION, CONFIGURATION CHANGE LEVEL INFO;  
ALTER AUDIT POLICY xsa_policy ENABLE;
```

Alternatively, you can use the *Auditing* app of the SAP HANA cockpit.

The audit log entries written by XS advanced applications can be viewed by querying the system view `XSA_AUDIT_LOG`. System privilege `AUDIT OPERATOR` or `AUDIT ADMIN` is required.

Related Information

[Auditing Activity in SAP HANA Systems \[page 240\]](#)

[Audit Trail Layout for Trail Target Database Table \[page 252\]](#)

16.7 Data Protection and Privacy in SAP HANA XS Advanced

As an application platform server, the SAP HANA extended application server, advanced model (XS advanced), provides operators with a selection of tools and functions to conform to the legal requirements of data protection concerning data processed by deployed applications, as well as the server infrastructure itself.

Introduction to Data Protection

Data protection is associated with numerous legal requirements and privacy concerns. In addition to compliance with general data privacy regulations, it is necessary to consider compliance with industry-specific legislation in different countries. SAP HANA XS advanced provides tools and functions to support compliance with regard to relevant legal requirements, including data protection.

This section and any other sections in this Security Guide do not give any advice on whether these features and functions are the best method to support company, industry, regional, or country-specific requirements. Furthermore, this guide does not give any advice or recommendations regarding additional features that would be required in specific IT environments; decisions related to data protection must be made on a case-by-case basis, taking into consideration the given system landscape and the applicable legal requirements.

i Note

In the majority of cases, compliance with data privacy laws is not a product feature. SAP software supports data privacy by providing security features and specific functions relevant to data protection, such as functions for the simplified blocking and deletion of personal data. SAP does not provide legal advice in any form. The definitions and other terms used in this guide are not taken from any given legal source.

Glossary

| Term | Definition |
|-----------------------------------|--|
| Blocking | A method of restricting access to data for which the primary business purpose has ended. |
| Business purpose | A legal, contractual, or in other form justified reason for the processing of personal data. The assumption is that any purpose has an end that is usually already defined when the purpose starts. |
| Consent | The action of the data subject confirming that the usage of his or her personal data shall be allowed for a given purpose. A consent functionality allows the storage of a consent record in relation to a specific purpose and shows if a data subject has granted, withdrawn, or denied consent. |
| Deletion | Deletion of personal data so that the data is no longer available. |
| End of purpose (EoP) | End of purpose and start of blocking period. The point in time, when the primary processing purpose ends (e.g. contract is fulfilled). |
| End of purpose (EoP) check | A method of identifying the point in time for a data set when the processing of personal data is no longer required for the primary business purpose . After the EoP has been reached, the data is blocked and can only be accessed by users with special authorization (for example, tax auditors). |

| Term | Definition |
|--------------------------------|---|
| Personal data | Any information relating to an identified or identifiable natural person ("data subject"). An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural, or social identity of that natural person |
| Residence period | The period of time between the end of business and the end of purpose (EoP) for a data set during which the data remains in the database and can be used in case of subsequent processes related to the original purpose. At the end of the longest configured residence period, the data is blocked or deleted. The residence period is part of the overall retention period. |
| Retention period | The period of time between the end of the last business activity involving a specific object (for example, a business partner) and the deletion of the corresponding data, subject to applicable laws. The retention period is a combination of the residence period and the blocking period. |
| Sensitive personal data | A category of personal data that usually includes the following type of information: <ul style="list-style-type: none"> • Special categories of personal data, such as data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, genetic data, biometric data, data concerning health or sex life or sexual orientation, or personal data concerning bank and credit accounts. • Personal data subject to professional secrecy • Personal data relating to criminal or administrative offenses • Personal data concerning insurances and bank or credit card accounts |
| Where-used check (WUC) | A process designed to ensure data integrity in the case of potential blocking of business partner data. An application's where-used check (WUC) determines if there is any dependent data for a certain business partner in the database. If dependent data exists, this means the data is still required for business activities. Therefore, the blocking of business partners referenced in the data is prevented. |

Scope of This Documentation

The XS advanced platform functions presented in this section only deal with aspects of protecting data that is managed by the platform itself, for example residing in server trace files or application logs. The XS advanced platform may be used to process data stored in other systems by leveraging communication capabilities with other data sources.

For aspects of data protection and privacy related to the SAP HANA database as basis of the platform stack, see the section *Data Protection in SAP HANA*.

General security requirements that are mandatory for data protection and privacy are covered elsewhere in this section on security for XS advanced. These include for example data-access control and logging, as well as communication security.

Personal Data in SAP HANA XS Advanced Applications

Applications deployed to the SAP HANA XS advanced server may process and store information related to an identified or identifiable natural person. Depending on the way a person interacts with the system, different server artifacts might be affected. For example, personal data of an internal platform user (for example, a Controller user) might be traced to platform trace logs or audit logs that are necessary to generally meet specific quality standards of the platform; whereas personal data of business users might appear in application logs controlled by the platform or even be stored to application-controlled services like an SAP HANA schema.

There are three categories of artifact to which personal data may be related:

- **Category 1: Platform-controlled artifacts like audit logs and server trace files**
Personal data originates from platform users. The data are required to guarantee operation of the platform.
- **Category 2: Platform-controlled artifacts like application log files or audits**
Personal data originates from business users and is transparent for the platform. These services are provided by the platform to support the operation of applications. With respect to personal data, they may be used by applications only for this purpose.
- **Category 3: Application-controlled artifacts like a persistence service or file system service**
Personal data originates from business users and is transparent for the platform.

Compliance with any existing data privacy law is not a product feature of the XS advanced application server. However, the XS advanced application server provides a variety of security-related features to implement general security requirements that can also be used by administrators for data protection and privacy.

By default, potential personal data stored in any of the three artifact types mentioned above is protected from unauthorized access. Additionally, for platform-controlled artifacts, there are retrieval functions in place for authorized users to query the stored data. A general deletion function is also available (for platform administrators only).

Caution

From the perspective of applications, the platform acts as a container that is not aware of the type or structure of the potentially processed or stored personal data even if the services are provided by the platform. Therefore, applications are fully responsible for providing the features to fulfill personal data requirements in application-controlled artifacts.

Related Information

[Processing of Personal Data in Platform-Controlled Artifacts \[page 344\]](#)

[Processing of Personal Data in Standard XS Advanced Applications and Services \[page 347\]](#)

[Data Protection and Privacy in SAP HANA \[page 270\]](#)

16.7.1 Processing of Personal Data in Platform-Controlled Artifacts

Personal data originating both from platform users and business users may be present in server artifacts.

As a general rule, platform users are typically internal users who operate the application services and server infrastructure (for example, Controller users like SpaceDeveloper or administrators). Business users, on the other hand, interact with deployed applications.

Server artifacts that may contain personal data originating from the server infrastructure (category 1)

Various interaction sequences with the XS advanced server initiated by **platform users** may leave personal data in server artifacts. These are listed in the table below.

All server logs are written to a rotating file, which means that older log lines are removed from the log history when a specified file size threshold is met. The size and number of all rotated application log files may be adjusted to suit customer needs. Data is generally discarded sooner if low file-size limits and a small number of history files are configured.

| Platform Artifact | Location | Type | User Type | Content |
|---|---|--------------------------------|-----------------------------|------------------------------|
| xscontroller_ <i>i</i> .log | Trace directory of the SAP HANA instance (alias 'cdtrace') and <code><xs-base>/controller_data/controller/tracing/log/</code> | Log file (rotated) | Platform user | User name |
| xsexeagent_ <i>i</i> .log | Trace directory of the SAP HANA instance (alias 'cdtrace') and <code><xs-base>/controller_data/controller/tracing/log/</code> | Log file (rotated) | Platform user | User name |
| hdbxscontroller_audit_ <i>i</i> >.log | Trace directory of the SAP HANA instance (alias 'cdtrace') | Log file (rotated) | Platform user | User name, client IP address |
| hdbxsexecutionagent_audit_ <i>i</i> >.log | Trace directory of the SAP HANA instance (alias 'cdtrace') | Log file (rotated) | Platform user | User name, client IP address |
| uaa.log | Trace directory of the SAP HANA instance (alias 'cdtrace') | Truncated and rotated log file | Platform and business users | User name, client IP address |
| uaa-audit.log | Trace directory of the SAP HANA instance (alias 'cdtrace') | Truncated and rotated log file | Platform and business users | User name, client IP address |

| Platform Artifact | Location | Type | User Type | Content |
|--|---|----------------------------|-----------------------------|------------------------------|
| access_log-controller-route-api.log | <xs-base>/controller_data/controller/router/webdispatcher/logs/ | Log file (rotated) | Platform user | Client IP address |
| access_log-external-uaa-route-uaa-server.log | <xs-base>/controller_data/controller/router/webdispatcher/logs/ (in xs base directory alias 'cdxs') | Log file (rotated) | Platform and business users | client IP address |
| STOREDEVENT table | Schema SYS_XS_RUNTIME | (Truncated) persistence | Platform users | User name |
| UAA shadow user | Schema SYS_XS_UAA (only if an external identity providers is configured) | persistence | Platform and business users | User name, email and so on |
| syslog | /var/log/messages | According to UNIX settings | Platform and business users | User name, client IP address |

Note

<xs-base> denotes the XS advanced base directory (alias 'cdxs').

The following platform-provided functions can be used on the artifacts listed above:

- To retrieve content of the log files, either access the file from the UNIX command line as <sid>adm, or view them with the SAP HANA database explorer ([Database Diagnostics Files](#))
- To retrieve the content of stored events (table STOREDEVENT), use the `xs events` command.
- To delete all entries in the listed log and audit files older than a specified date, use the command `xsa delete-personal-data <date> --exclude APP, USERS` executed as <sid>adm.

Caution

Deleted data cannot be recovered.

The `exclude` option allows you to prevent the deletion of specific artifacts:

- PLATFORM
Platform logs including the event log in the database and the trace files
- AUDIT
Audit log files of XS advanced
- APP
Application log files
- ACCESS
WebDispatcher access logs
- USERS
Shadow users in the UAA

i Note

The size and number of all rotated log files may be adjusted to suit customer needs. Note that the lower the file-size limit and the shorter the length of time for which files are retained, the sooner logged data will be discarded.

i Note

If audit events are written to the UNIX syslog, the system operator must configure the retention policy, and so on.

For more information about the `xs` and `xsa` commands mentioned here, see the section on maintaining the XS advanced runtime environment in the *SAP HANA Administration Guide*.

Server artifacts that may contain personal data originating from applications (category 2)

Various interaction sequences with applications deployed to the XS advanced server may also leave personal data related to **business users** in artifacts that are managed by the server.

⚠ Caution

There may be additional relevant data managed by the application itself, for example, data that is retained in an application-specific persistence. For this kind of data, it is the responsibility of the application to provide ways to fulfill data-protection and privacy requirements.

| Artifact | Location | Type | User Type | Content |
|---|---|--------------------|--------------------|------------------------------|
| Application logs (stdout, stderr) | <code><app_working>/host/executionroot/<app-instance>/app-logs</code> | Log file (rotated) | All types of users | Application specific |
| HTTP access logs of application instances | <code><xs_base>/controller_data/controller/router/webdispatcher/logs</code> | Log file (rotated) | All types of users | Client IP address |
| Auditlog service | SAP HANA audit log | Database table | All types of users | User name, client IP address |

The following platform-provided functions can be used on the artifacts listed above:

- To retrieve the full content of application logs, execute the command `xs logs <app> --all` to retrieve all log lines of the application you are interested in.
- To delete all application log entries older than a specified date, execute the command `xsa delete-personal-data <date> --exclude PLATFORM, AUDIT, USERS as <sid>adm`.

⚠ Caution

Deleted data cannot be recovered.

- To view the entries written to the auditlog-service, use the available SAP HANA tools to view and manage the data.

Related Information

[Audit Trails \[page 246\]](#)

16.7.2 Processing of Personal Data in Standard XS Advanced Applications and Services

Personal data originating from business users and written by system applications may be present in server artifacts (category 1 – 3).

| Application Name | Personal Data | Location | Comment |
|-------------------|---|---|--|
| AppRouter | User ID, name, and client IP address in the event of unsuccessful login | Audit log service and/or application log (category 2) | Necessary for application operation |
| deploy-service | User ID, name, client IP of platform user | Application log (category 2), internal persistency scoped to deployment process (Category 3) | Necessary for application deployment |
| product-installer | User ID, name, client IP address of platform user | Application log (category 2), internal log persistency scoped to installed product (category 3) | Necessary for application installation |
| job-scheduler | User ID, name, client IP address of platform user | Application log (category 2), internal log persistency scoped to created service (category 3) | Necessary for service deployment |
| Admin tools | User ID, name, client IP address of platform user | Application log (category 2) | Necessary for platform operation |
| XSA cockpit | User ID, name, client IP address of platform user | Application log (category 2) | Necessary for platform operation |

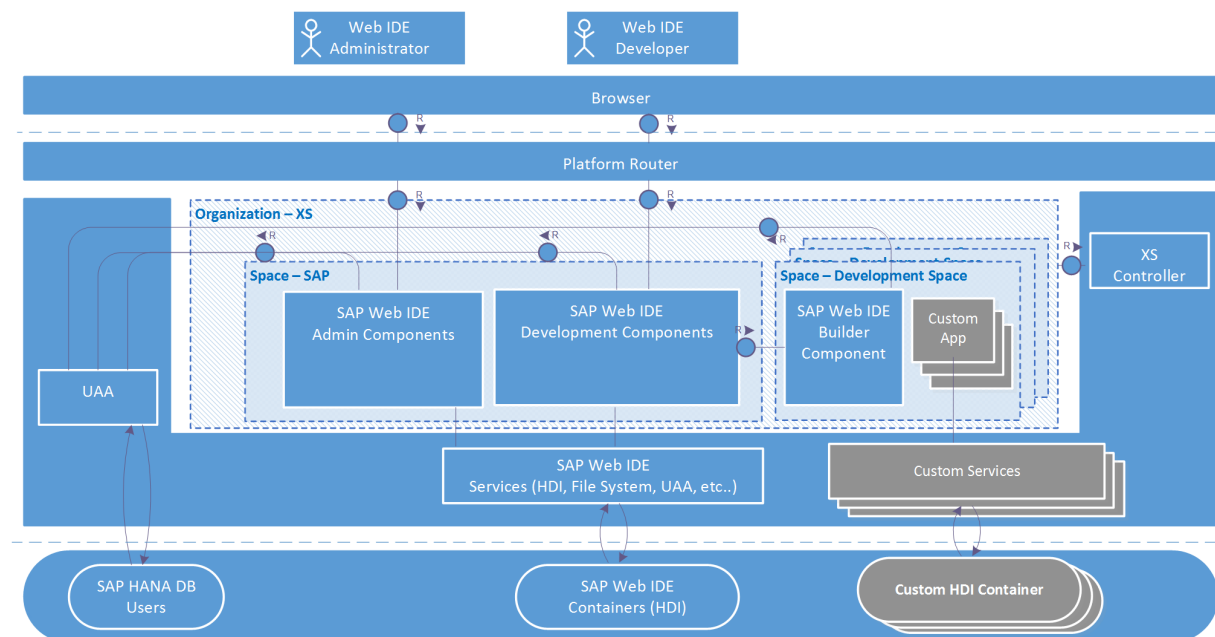
16.8 Security Aspects of SAP Web IDE for SAP HANA

SAP Web IDE for SAP HANA (SAP Web IDE) is a browser-based integrated development environment for the development of SAP HANA-based applications. These applications are comprised of web-based or mobile UIs, business logic, and extensive SAP HANA data models.

SAP Web IDE supports developers who use SAP HANA Extended Application Services, advanced model (XS Advanced), by providing a variety of tools. These tools include syntax-aware editors for code and SAP HANA artifacts, graphical editors for Core Data Services (CDS) data models and calculation views, as well as inspection, testing, and debugging tools.

Architecture

SAP Web IDE is comprised of several application components deployed in XS Advanced application server. The following diagram provides an architectural overview of SAP Web IDE in XS Advanced.



i Note

Since SAP Web IDE is not tenant-aware, customers who use it in an environment with multiple tenant databases should install an instance of SAP Web IDE in each XS Advanced organization that represents a tenant.

Multiple Spaces for Development

SAP Web IDE supports multiple spaces in XS Advanced, in which developers build and run their applications. To ensure the isolation of the development environment, different development teams should use separate spaces.

i Note

We strongly recommend not to use the predefined *SAP* space for development. Doing so compromises the SAP Web IDE security.

SAP Web IDE provides an administration tool to enable XS Advanced spaces for development. For more information about this tool, see *Managing Spaces for Development* in *SAP Web IDE for SAP HANA - Installation and Upgrade Guide*.

i Note

Developers who use the same space can access each other's application artifacts. In fact, any authenticated SAP HANA database user assigned to the *SpaceDeveloper* role for a specific space has full access to all applications in this space, and can potentially cause disruption or misuse of these applications.

All SAP Web IDE components, except the Builder component, are installed in the predefined *SAP* space, whereas the Builder component is installed in each space used for development.

Communication with Remote Systems

SAP Web IDE supports access to remote systems, such as a Git source control system. For remote Git repositories, which issue SSL certificates that are not trusted publicly, SAP Web IDE provides a command-line tool that enables administrators to manage SSL certificates. For more information about this tool, see *Managing SSL Certificates* in *SAP Web IDE for SAP HANA - Installation and Upgrade Guide*.

Related Information

[Security for SAP HANA Extended Application Services, Advanced Model \[page 293\]](#)

[User Authorization and Authentication \[page 350\]](#)

[Known Security-Related Issues \[page 351\]](#)

16.8.1 User Authorization and Authentication

User administration, authorization, and authentication concepts of SAP Web IDE for SAP HANA.

User Administration and Authentication

SAP Web IDE is installed on top of XS Advanced, and uses its User Account and Authorization service (UAA) and the application router to manage user logon and logout requests. The UAA service centrally manages the issuing of tokens for propagating the user identity to application containers and to the SAP HANA database.

Authorization

Authorization grants access to application resources and services based on the defined user permissions. Authorization checks are performed on the levels of the XS Advanced controller and each of the SAP Web IDE components.

SAP Web IDE supports the following user types:

- A developer user in SAP Web IDE for SAP HANA is an SAP HANA database user who has additional permissions to create, modify, build, and run applications in SAP Web IDE. These developers are assigned to personal SAP Web IDE workspaces, where they manage their own application artifacts, such as projects, modules, and files. Workspace access is granted only to its owner.

The following table lists the roles and role collections that are required for a developer user using Web IDE:

| User Type | Role/Role Collections | Description |
|-----------|---|---|
| Developer | <i>XS_CONTROLLER_USER</i> role collection | Grants read-write permissions within the assigned organization or space. |
| Developer | <i>SpaceDeveloper</i> role | Assigned per space in XS Advanced. Enables users to access the shared resources of the space, and to deploy, build, and run applications. |
| Developer | A role collection containing the <i>WebIDE_Developer</i> role | Enables users to develop applications using SAP Web IDE. |

Note

To facilitate the work of developers, the SAP Web IDE npm registry cache component provides Node.js modules, which are otherwise available to developers via the SAP public npm registry. Any user has read-only access to this registry without any authentication or authorization.

- **SAP Web IDE administrator** is an SAP HANA database user who has additional permissions to perform administration tasks for SAP Web IDE.

The following table lists the required roles and role collections for an administrator of Web IDE.

| User Type | Role/Role Collections | Description |
|---------------|--|---|
| Administrator | <i>XS_CONTROLLER_USER</i> role collection | Grants read-write permissions within the assigned organization or space. |
| Administrator | <i>SpaceDeveloper</i> role | Assigned per space in XS Advanced. Enables users to access the shared resources of the space, and to deploy, build, and run applications. |
| Administrator | A role collection containing the <i>WebIDE_Administrator</i> role template | Enables users to access the SAP Web IDE administration tools, such as SSL management and space enablement. |

Authentication and Authorization for Custom Applications

Custom applications developed using SAP Web IDE are standard XS Advanced applications, which are deployed into spaces in XS Advanced. SAP Web IDE does not perform any UAA functions on behalf of the applications. Therefore, application developers should implement their own authentication and authorization support using the platform security functions, which are provided by XS Advanced.

Connections to databases in the SAP HANA database explorer

When you specify the credentials to connect to an SAP HANA database, those credentials are not saved to your browser. To have the credentials persist between sessions, save the credentials to the SAP HANA secure store

Related Information

[User Administration and Authentication in SAP HANA XS Advanced \[page 302\]](#)

[Security for SAP HANA Extended Application Services, Advanced Model \[page 293\]](#)

16.8.2 Known Security-Related Issues

A list of known security-related issues of the SAP Web IDE for SAP HANA.

| Topic | Issue / Impact | Workaround |
|---------------------|---|-----------------------------------|
| Backup and recovery | SAP Web IDE does not support backup and recovery for workspace content (projects, folders, or files). | Use a Git repository as a backup. |

| Topic | Issue / Impact | Workaround |
|-------------|--|---|
| File upload | No automatic virus scan or content validation of files is performed before uploading them to SAP Web IDE. Malicious content can be uploaded. | Use external tools to validate the content of the files and scan them for viruses before uploading them to SAP Web IDE. Alternatively, use a file-based antivirus tool to scan the files on the disk. |

Related Information

[Securing the SAP HANA Database Explorer \[page 286\]](#)

17 SAP HANA Security Reference Information

Security reference information for SAP HANA

i Note

Please also refer to the document *SAP HANA Security Checklists and Recommendations*.

[SQLScript Security Procedures \[page 353\]](#)

SQLScript procedures for security functions

[Security Reference for Tenant Databases \[page 358\]](#)

Reference information for secure configuration of tenant databases

[Components Delivered as SAP HANA Content \[page 363\]](#)

The following sections provide the technical details, key features, and roles of all software components delivered with the SAP HANA platform as SAP HANA XS classic content.

17.1 SQLScript Security Procedures

SQLScript procedures for security functions

For more information about calling SQLScript procedures from clients, see *SAP HANA SQLScript Reference*.

[IS_VALID_USER_NAME \(SYS\) \[page 354\]](#)

SQLScript procedure to validate whether a given user name will be accepted by the SAP HANA database as part of a CREATE USER or ALTER USER statement.

[IS_VALID_PASSWORD \(SYS\) \[page 355\]](#)

SQLScript procedure to validate whether a given password will be accepted by the SAP HANA database as part of a CREATE USER or ALTER USER statement.

[GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION \(SYS\) \[page 356\]](#)

SQLScript procedure to generate a filter condition based on ABAP authorization objects that can be used in the CONDITION PROVIDER clause of an analytic privilege

17.1.1 IS_VALID_USER_NAME (SYS)

SQLScript procedure to validate whether a given user name will be accepted by the SAP HANA database as part of a CREATE USER or ALTER USER statement.

Procedure Call

```
CALL SYS.IS_VALID_USER_NAME (<parameter_list>)
```

Input Parameters

- User name

Output Parameters

- ERROR_CODE
- ERROR_MESSAGE

Examples

❖ Example

Input:

```
CALL SYS.IS_VALID_USER_NAME('bob', ? ,?)
```

Output:

```
Out (1);Out (2)  
0      ;
```

The user name is valid.

❖ Example

Input:

```
CALL SYS.IS_VALID_USER_NAME('billy bob', ?, ?)
```

Output:

```
Out (1);Out (2)
```

```
257 ;sql syntax error: incorrect syntax near \"bob\": line 1 col 19 (at pos 19)
```

The user name is not valid.

17.1.2 IS_VALID_PASSWORD (SYS)

SQLScript procedure to validate whether a given password will be accepted by the SAP HANA database as part of a CREATE USER or ALTER USER statement.

Procedure Call

```
CALL SYS.IS_VALID_PASSWORD (<parameter_list>)
```

Input Parameters

- Password

i Note

If the password contains double quotes ("), use the security function ESCAPE_DOUBLE_QUOTES to return the password with escaped double quotes, and then enter the result as the input parameter. Alternatively, you can use the function directly in the procedure call: CALL

```
SYS.IS_VALID_PASSWORD ( ESCAPE_DOUBLE_QUOTES (?), ?, ? ).
```

For more information about the ESCAPE_DOUBLE_QUOTES function, see the *SAP HANA SQL and System Views Reference*.

Output Parameters

- ERROR_CODE
- ERROR_MESSAGE

Examples

❖ Example

Input:

```
CALL SYS.IS_VALID_PASSWORD('Abcd1234', ?, ?)
```

Output:

```
Out (1);Out (2)  
0      ;
```

The password is valid.

❖ Example

```
'CALL SYS.IS_VALID_PASSWORD(ESCAPE_DOUBLE_QUOTES(?), ?, ? )'
```

❖ Example

Input:

```
CALL SYS.IS_VALID_PASSWORD('asdf', ?, ?)
```

Output

```
Out (1);Out (2)  
412    ;invalid password layout: minimal password length is [8]
```

The password is not valid.

17.1.3 GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION (SYS)

SQLScript procedure to generate a filter condition based on ABAP authorization objects that can be used in the CONDITION PROVIDER clause of an analytic privilege

Procedure Call

```
CALL SYS.GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION(<parameter_list>)
```

Input Parameters

- Schema in which the SAP authorization tables UST12 and USRBF2 reside
- An expression that contains at least one CHECKID

i Note

Multiple CHECKIDs may be concatenated. Expression can contain AND, OR and NOT Boolean operators.

- A JSON-formatted string that contains the information required to translate each ABAP authorization object referenced. For each CHECKID, the following must be specified:
 - `authobj`
This specifies the name of the ABAP authorization object to which the CHECKID is mapped. This may contain an empty value if the CHECKID itself is the name of the ABAP authorization object.
 - `filter`
This allows you to specify fixed conditions that the user's ABAP authorizations need to match. It is a list of required fields (`key`) and their values (`values`).
 - `mappings`
This specifies the mapping of FIELD value (`fieldName`) in the ABAP authorization object to column name (`mappedName`) of the target view to be protected.

Output Parameters

An NCLOB that contains the SQL filter condition

i Note

The procedure returns 1>1 if the user has no permissions and 1=1 if the user has full access.

Examples

❖ Example

Input:

```
CALL SYS.GENERATE_STRUCTURED_PRIVILEGE_PFCG_CONDITION (
'A_TEST_SCHEMA',
'CHECKID1',
'{"data":
  {
    "CHECKID1":
    {
      "authobj": "OBJ1",
      "filter": [{"key": "ACTVT", "valueList": ["03"]}],
      "mappings": [{"fieldName": "SACMTSOID", "mappedName": "SO_ID"},
{"fieldName": "SACMTSOLCS", "mappedName": "LIFECYCLE_STATUS"}]
    }
  }
}
```

```
}  
},  
?)
```

Related Information

[Shared Business Authorizations in SAP HANA \[page 184\]](#)

17.2 Security Reference for Tenant Databases

Reference information for secure configuration of tenant databases

[Restricted Features in Tenant Databases \[page 358\]](#)

To safeguard and/or customize your system, certain features of the SAP HANA database can be disabled in tenant databases.

[Default Blacklisted System Properties in Tenant Databases \[page 361\]](#)

In systems that support tenant databases, there is configuration change blacklist `multidb.ini`, which is delivered with a default configuration.

17.2.1 Restricted Features in Tenant Databases

To safeguard and/or customize your system, certain features of the SAP HANA database can be disabled in tenant databases.

Some features of the SAP HANA database are not required or desirable in certain environments, in particular features that provide direct access to the file system, the network, or other resources. The table below lists those features that you can explicitly disable in tenant databases.

The system view `M_CUSTOMIZABLE_FUNCTIONALITIES` lists all features that can be disabled and their status. This view exists in both the `SYS` schema of every database, where it contains database-specific information, and in the `SYS_DATABASES` schema of the system database, where it contains information about the enablement of features in all databases. For more information, see `M_CUSTOMIZABLE_FUNCTIONALITIES` in the *SAP HANA SQL and Systems View Reference*.

For more information about how to disable features, see in the *SAP HANA Administration Guide*.

i Note

Features are hierarchically structured. If you disable a feature with sub-features, these are also disabled.

| Feature | Feature Description | Why Disable? |
|---|---|--|
| RINTEGRATION | R language | Feature not required in all deployment scenarios |
| XB_MESSAGING_SUBSCRIPTIONS | Subscriptions For External Messaging Providers | |
| AFL | Access to Application Function Libraries (AFL) for business logic in native C++ | Feature not required in all deployment scenarios |
| XB_EXTERNAL_CONNECTIVITY | External Connectivity With XB Message Bus | |
| BACKUP | Backup operations | File system access not wanted |
| BACKUP.IGNOREPATH_RESTRICT | Ignore path restrictions for backup operations | File system access not wanted, safeguard against directory traversal attacks |
| IMPORTEXP | Import and export operations | File system access not wanted |
| IMPORTEXP.IMPORT | Import operations | File system read access not wanted |
| IMPORTEXP.EXPORT | Export operations | File system write access not wanted |
| IMPORTEXP.IGNORE_PATH_RESTRICT | Ignoring of path restrictions for import and export | File system access not wanted, safeguard against directory traversal attacks |
| BUILTINPROCEDURE | Execution of procedures associated with critical and/or optional functions | -- |
| BUILTINPROCEDURE.REORG_GENERATE | Data redistribution operations | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.REORG_EXECUTE | | |
| BUILTINPROCEDURE.REORG_CLEAR_LOGS | | |
| BUILTINPROCEDURE.GEM | Procedure to use the graph engine | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.MANAGEMENT_CONSOLE_PROC | Access to the built-in SAP HANA management console (hdbcons) | Safeguard against leakage of SAP HANA process information |
| BUILTINPROCEDURE.KERNELCALL | Access to rowstore internal maintenance features | Safeguard against leakage of SAP HANA process information |
| BUILTINPROCEDURE.TREXVIADBSL | Operation of an SAP Business ByDesign system | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.COMPRESS_FILE | Compression of trace files before they are transferred | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.GET_FULL_SYSTEM_INFO_DUMP | Triggering of complete information dump of the entire system | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.FULL_SYSTEM_INFO_DUMP_CREATE | Triggering creation of complete information dump of the entire system | Feature not required in all deployment scenarios |

| Feature | Feature Description | Why Disable? |
|---|--|--|
| BUILTINPROCEDURE.FULL_SYSTEM_INFO_DUMP_DELETE | Triggering deletion of complete information dump of the entire system | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.FULL_SYSTEM_INFO_DUMP_RETRIEVE | FULL_SYSTEM_INFO_DUMP_RETRIEVE execution | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.DSO | Creation of and access to DataStore Objects (DSOs) for SAP Business Warehouse (BW) powered by SAP HANA | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.STATISTICS_SERVER_CONFIGCHECKPROC | Validation of the statistics server configuration and its e-mail notification capability | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.BW_PRECHECK_RELEASE_LOCK | Operation of an SAP BW powered by SAP HANA system | Feature not required in all deployment scenarios |
| BUILTINPROCEDURE.BW_PRECHECK_ACQUIRE_LOCK_WITH_TYPE | | |
| BUILTINPROCEDURE.BW_PRECHECK_ACQUIRE_LOCK | | |
| BUILTINPROCEDURE.BW_CONVERT_CLASSIC_TO_IMO_CUBE | | |
| BUILTINPROCEDURE.BW_F_FACT_TABLE_COMPRESSION | | |
| BUILTINPROCEDURE.UPDATE_LANDSCAPE_CONFIGURATION | Changes to system landscape and the available services in a system | Feature not required in all deployment scenarios |
| ALTERSYSTEM | Execution of the statement ALTER SYSTEM RECONFIGURE SERVICE, which re-reads the service configuration | Feature not required in all deployment scenarios |
| ALTERSYSTEM.RECONFIGURE_SERVICE | | |
| SMARTDATAACCESS | Federated access to other database systems through virtual tables | Feature not required in all deployment scenarios |
| DXC | Data acquisition and consumption of data models from the SAP Business Suite | Feature not required in all deployment scenarios |
| DYNAMIC_TIERING | SAP HANA Dynamic Tiering operations | Feature not required in all deployment scenarios |
| DYNAMIC_TIERING.CREATE_EXTENDED_STORAGE | Creation of extended storage | |
| DYNAMIC_TIERING.DROP_EXTENDED_STORAGE | Deletion of extended storage | |
| DYNAMIC_TIERING.ALTER_EXTENDED_STORAGE | Changes to extended storage | |
| DYNAMIC_TIERING.ALTER_TABLE_TYPE | Conversion of a regular database table to an extended table or the reverse | |
| DYNAMIC_TIERING.BULK_INSERT_OPTIMIZATION | Bulk insert optimization that executes large inserts into extended tables using a load statement | |

| Feature | Feature Description | Why Disable? |
|--|---|--|
| DYNAMIC_TIERING.QUERY_PLAN_RELOCATION | Query relocation operation that moves data from SAP HANA and SAP HANA Dynamic Tiering for optimal query performance | |
| ACCELERATOR_FOR_ASE | SAP HANA Accelerator for SAP ASE operations | |
| BOE | SAP BusinessObjects Explorer API | Feature not required in all deployment scenarios |
| SMART_DATA_STREAMING | SAP HANA Streaming Analytics operations | Feature not required in all deployment scenarios |
| GRAPH_ENGINE | SAP HANA Graph Engine | Feature not required in all deployment scenarios |
| LDAP | All operations related to LDAP group authorization including creating/modifying LDAP providers, changing user authorization mode to <code>LDAP</code> , mapping LDAP groups to SAP HANA roles, and so on. | Feature not required in all deployment scenarios |
| <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p>⚠ Caution</p> <p>If the LDAP feature is disabled, existing users with authorization mode <code>LDAP</code> cannot log on. The authorization mode of these users can be changed to <code>LOCAL</code>.</p> </div> | | |
| EPMPANNING | SAP HANA Enterprise Performance Management planning feature | Feature not required in all deployment scenarios |
| PLANNINGENGINE | Planning engine features | Feature not required in all deployment scenarios |

17.2.2 Default Blacklisted System Properties in Tenant Databases

In systems that support tenant databases, there is configuration change blacklist `multidb.ini`, which is delivered with a default configuration.

The table below lists the system properties that are included in the `multidb.ini` file by default. This means that tenant database administrators cannot change these properties. System administrators can still change these properties in the system database in all layers.

You can customize the default configuration change blacklist by changing existing entries in the `multidb.ini` file and adding new ones. For more information about how to prevent changes to specific system properties in tenant databases in the *SAP HANA Administration Guide*.

| File/Section | Properties | Description |
|---|---|--|
| auditing configuration | <ul style="list-style-type: none"> • default_audit_trail_type • emergency_audit_trail_type • alert_audit_trail_type • critical_audit_trail_type • audit_statement_length | Prevents configuration of audit trail targets and the maximum audit statement length |
| communication | * | Prevents configuration of default key and trust stores, as well as other critical communication settings |
| global.ini/customizable_functionalities | * | Prevents disabling of restricted features |
| global.ini/extended_storage | * | Prevents configuration of extended storage (SAP HANA dynamic tiering) |
| global.ini/persistence | <ul style="list-style-type: none"> • basepath_datavolumes_es • basepath_logvolumes_es • basepath_databackup_es • basepath_logbackup_es | |
| global.ini/system_replication | <ul style="list-style-type: none"> • keep_old_style_alert • enable_full_sync • operation_mode | Prevents configuration of certain system replication settings |
| global.ini/system_replication_communication | * | |
| global.ini/system_replication_hostname_resolution | * | |
| global.ini/xb_messaging | * | Prevents configuration of messaging |
| multidb.ini/readonly_parameters | * | Prevents configuration of the multidb.ini file itself |
| indexserver.ini/authentication | SapLogonTicketTrustStore | Prevents configuration of the trust store for user authentication with logon/assertion tickets |
| memorymanager | <ul style="list-style-type: none"> • allocationlimit • minallocationlimit • global_allocation_limit • async_free_threshold • async_free_target | Prevents configuration of memory allocation parameters |
| execution | max_concurrency | Prevents configuration of threading and parallelization parameters |

| File/Section | Properties | Description |
|--------------|---|-------------|
| session | <ul style="list-style-type: none"> • maximum_connections • maximum_external_connections | |
| sql | sql_executors | |

17.3 Components Delivered as SAP HANA Content

The following sections provide the technical details, key features, and roles of all software components delivered with the SAP HANA platform as SAP HANA XS classic content.

For more information about what SAP HANA content is, see *SAP HANA Content*.

[Administration \[page 363\]](#)

SAP HANA content related to system and database administration

[Application Lifecycle Management \[page 369\]](#)

SAP HANA content for application lifecycle management

[Runtime Libraries \[page 371\]](#)

SAP HANA content for runtime libraries

[Configuration \[page 372\]](#)

SAP HANA content for configuration

[Supportability and Development \[page 373\]](#)

SAP HANA content for supportability and development

[User Interface \[page 377\]](#)

SAP HANA content for user interface

[Documentation \[page 380\]](#)

SAP HANA documentation delivered as SAP HANA content

Related Information

[Security of SAP HANA Content \[page 278\]](#)

17.3.1 Administration

SAP HANA content related to system and database administration

- [HANA_ADMIN \[page 364\]](#)
- [HANA_BACKUP \[page 364\]](#)

- [HANA_HDBLCM \[page 364\]](#)
- [HANA_SEC_BASE \[page 365\]](#)
- [HANA_SEC_CP \[page 365\]](#)
- [HANA_SYS_ADMIN \[page 365\]](#)
- [HANA_XS_BASE \[page 365\]](#)

17.3.1.1 HANA_ADMIN

This component is available for downward compatibility reasons. It does not contain any content.

For more information about Web-based administration using the SAP HANA cockpit in SAP HANA 2.0, see the *SAP HANA Administration Guide*.

Only the following roles are available with the HANA_ADMIN component:

- `sap.hana.admin.roles::SolutionMangagerMonitor`
- `sap.hana.admin.roles::RestrictedUserDBSLAccess`

The privileges contained in these roles provide users administrating SAP HANA using SAP Solution Manager and SAP NetWeaver tools with the required authorization in SAP HANA.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. 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.

17.3.1.2 HANA_BACKUP

This component is available for downward compatibility reasons. It does not contain any content.

For more information about Web-based backup using the SAP HANA cockpit in SAP HANA 2.0, see the *SAP HANA Administration Guide*.

17.3.1.3 HANA_HDBLCM

This component is available for downward compatibility reasons. It does not contain any content.

For more information about access to the SAP HANA platform lifecycle management Web user interface from the SAP HANA cockpit in SAP HANA 2.0, see the *SAP HANA Administration Guide*.

17.3.1.4 HANA_SEC_BASE

This component is available for downward compatibility reasons. It does not contain any content.

For more information about Web-based security administration using the SAP HANA cockpit in SAP HANA 2.0, see the *SAP HANA Administration Guide*.

17.3.1.5 HANA_SEC_CP

This component is available for downward compatibility reasons. It does not contain any content.

For more information about Web-based security administration using the SAP HANA cockpit in SAP HANA 2.0, see the *SAP HANA Administration Guide*.

17.3.1.6 HANA_SYS_ADMIN

This component is available for downward compatibility reasons. It does not contain any content.

For more information about Web-based administration of tenant databases using the SAP HANA cockpit in SAP HANA 2.0, see the *SAP HANA Administration Guide*.

17.3.1.7 HANA_XS_BASE

This component provides a Web application for configuring and managing SAP HANA XS classic applications and system-level settings such as SMTP and security-related details (SAML, trust store, and so on).

Technical Details

| | |
|----------------------------|---|
| Delivery unit | HANA_XS_BASE |
| Prerequisites | None |
| Content type | Automated content |
| Content details | SAP HANA XS classic applications (including data model, tables, roles, and user interfaces) |
| Target users | SAP HANA XS classic application administrators |
| Web application URL | <code>http(s)://<host>:<port>/sap/hana/xs/admin</code> |

Key Features

The SAP HANA XS Administration Tool enables users to configure and manage SAP HANA XS classic applications and system-level settings. It provides the following features:

- **SAP HANA XS application configuration**
Supports the configuration of application security (public/private) and user authentication methods (basic, form-based, logon tickets, X509, and SAML). It also supports the management of SQL connection configurations, HTTP destinations, and job schedules.
- **SAML configuration**
Enables the configuration and management of SAML service providers (URLs, metadata) and identity providers (IDP metadata, certificates, destinations)
- **Trust management**
Enables trust store configuration and management, and certificate management
- **SMTP configurations**
Enables the configuration and management of e-mail server settings for outbound e-mail connections. It also supports the management of authentication type with credentials, transport security settings, and socket proxy settings.
- **User self-service administration**
Enables the administration of user self-service requests (acceptance/rejection of user requests). It also support the activation of users, the granting of roles and the management of access lists such as blacklist/whitelist email id/domain/IP range adding constraints to the user self-service process
- **Online Translation Tool**
Enables the user to provide manual translation for text strings used in the application's user interface, error messages, and documentation. Also the user can export and import XLIFF-formatted files into the tool. The tool is integrated with SAP Translation Hub for recommendations of the translated texts.

i Note

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.

- **SAP Web Dispatcher HTTP Tracing application**
HTTP tracing for individual SAP HANA XS applications can be enabled in the SAP HANA Web Dispatcher. The SAP HANA XS Administration Tools include the SAP Web Dispatcher HTTP Tracing application, which you can use to enable and disable HTTP tracing in the SAP Web Dispatcher for SAP HANA XS applications.

Roles

The following roles are available with this component. Users must have the privileges contained in one or more of these roles before they can use the component and its functions.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. 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.

| Role | Description |
|---|--|
| <code>sap.hana.xs.debugger::Debugger</code> | Use of the debugging features of the SAP HANA Web-based Development Workbench |
| <code>sap.hana.xs.admin.roles::HTTPDestAdministrator</code> | Full access to HTTP destination configurations (display and edit) |
| <code>sap.hana.xs.admin.roles::HTTPDestViewer</code> | Read-only access to HTTP destination configurations, which are used to specify connection details for outbound connections, for example using the server-side JavaScript Connectivity API that is included with SAP HANA XS |
| <code>sap.hana.xs.admin.roles::JobAdministrator</code> | Full access to the configuration settings for SAP HANA XS job schedules (defined in .xsjob files); user can specify start/stop times, the user account to run the job, and the locale |
| <code>sap.hana.xs.admin.roles::JobSchedulerAdministrator</code> | Full access to the configuration settings for SAP HANA XS job schedules (defined in .xsjob files); user can specify start/stop times, the user account to run the job, and the locale User can also enable or disable scheduling of jobs. |
| <code>sap.hana.xs.admin.roles::JobViewer</code> | Read-only access to the configuration settings for SAP HANA XS job schedules (defined in .xsjob files). |
| <code>sap.hana.xs.formLogin.profile::ProfileOwner</code> | Management of user profile settings such as date/time format and locale It also allows the changing of user password. |
| <code>sap.hana.xs.admin.roles::RuntimeConfAdministrator</code> | Full access to the configuration settings for SAP HANA XS application security and the related user-authentication providers |
| <code>sap.hana.xs.admin.roles::RuntimeConfViewer</code> | Read-only access to the configuration settings for SAP HANA XS application security and the related user-authentication providers, for example, SAML or X509 |
| <code>sap.hana.xs.admin.roles::SAMLAdministrator</code> | Full access to SAML configurations, including both the service provider and the identity providers User can add new entries and make changes to existing service or identity providers, as well as parse the resulting metadata |
| <code>sap.hana.xs.admin.roles::SAMLViewer</code> | Read-only access to SAML configurations that are used to provide details of SAML service providers and identity providers |
| <code>sap.hana.xs.admin.roles::SMTPDestAdministrator</code> | Read-only access to SMTP configurations that are used to specify configuration settings for outbound mail connections to external mail servers |

| Role | Description |
|---|---|
| <code>sap.hana.xs.admin.roles::SMTPDestViewer</code> | Full access to SMTP configurations (display and edit) User can maintain mail server details, authentication type with credentials, transport security settings and socket proxy settings. |
| <code>sap.hana.xs.admin.roles::SQLCCAdministrator</code> | Full access to SQL connection configurations (SQLCC) |
| <code>sap.hana.xs.admin.roles::SQLCCViewer</code> | Read-only access to SQL connection configurations (SQLCC), which are used to enable the execution of SQL statements from inside your server-side JavaScript application with credentials that are different to the credentials of the requesting user |
| <code>sap.hana.xs.admin.roles::TrustStoreAdministrator</code> | Full access to the SAP HANA XS application trust store that manages the certificates required to start SAP HANA XS applications |
| <code>sap.hana.xs.admin.roles::TrustStoreViewer</code> | Read-only access to the trust store that contains the server's root certificate or the certificate of the certification authority that signed the server's certificate |
| <code>sap.hana.xs.admin.roles::USSAdministrator</code> | Administration of user requests submitted by end users through the User Self-Services application It is also possible to manage access lists such as blacklist/whitelist email id/domains/IP range adding constraints to the user self-service process |
| <code>sap.hana.xs.admin.roles::USSExecutor</code> | Role assigned to technical user to enable user self-service application in the system |
| <code>sap.hana.xs.wdisp.admin::WebDispatcherAdmin</code> | Full access to the SAP HANA Web Dispatcher Administration tool used by administrators to maintain secure inbound communication, for example, to enable SSL/TLS connections between browser front-ends or an ABAP system and an SAP HANA XS application |
| <code>sap.hana.xs.wdisp.admin::WebDispatcherMonitor</code> | Read-only access to the information displayed in the SAP HANA Web Dispatcher Administration tool |
| Translator | Enables an SAP HANA user to maintain translation text strings with the SAP HANA Online Translation Tool |
| <code>WebDispatcherHTTPTracingViewer</code> | Read-only access to the HTTP setting of SAP HANA XS applications running on the selected SAP HANA instance. This role extends the <code>JobViewer</code> role to enable the user to view details of the <code>xsjob</code> configuration (<code>httptracing.xsjob</code>) that starts and stops the HTTP tracing tasks. |
| <code>WebDispatcherHTTPTracingAdministrator</code> | Full access required to maintain HTTP tracing in the SAP Web Dispatcher for SAP HANA XS applications. This role extends the <code>JobAdministrator</code> role to enable the user to maintain the XS job file (<code>httptracing.xsjob</code>) used to configure and enable HTTP tracing for XS applications in the SAP Web Dispatcher. |

17.3.2 Application Lifecycle Management

SAP HANA content for application lifecycle management

- [HANA_XS_LM \[page 369\]](#)

17.3.2.1 HANA_XS_LM

This component provides a Web application for the application lifecycle management of components developed for SAP HANA XS.

Technical Details

| | |
|---------------------|---|
| Delivery unit | HANA_XS_LM |
| Prerequisites | SAPUI5_1 |
| Content type | Automated content |
| Content details | SAP HANA XS classic application |
| Target users | Application developers, content administrators |
| Web application URL | <code>http(s)://<host>:<port>/sap/hana/xs/lm</code> |

Key Features

The SAP HANA Application Lifecycle Management application enables application developers to create products, delivery units, packages, and basic application components. Administrators can use the application to set up the transport of delivery units, start and monitor transports, and upload or download delivery unit archives.

Roles

The following roles are available with the SAP HANA Application Lifecycle Management component. Users must have the privileges contained in one or more of these roles before they can use the component and its functions.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's

security policy. 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.

| Role | 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, Execute Transport, and Transport. |
| <code>sap.hana.xs.lm.roles::Developer</code> | Enables the user to work on a change to which he is assigned and to approve own contributions to the change. This role includes the privileges of the Display role. |
| <code>sap.hana.xs.lm.roles::DevelopmentExpert</code> | Enables the user to perform all actions involved in change recording (for example, create, assign objects to, release, delete, assign other users to a change, 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::Execute Transport</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. A role based on this role template cannot be assigned to normal users; it is granted as part of the Execute Transport 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 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. |

For tasks that require interaction with external tools, the privileges in the following additional roles are required:

| Role | Description |
|--|---|
| <code>sap.hana.ide.roles::EditorDeveloper</code> | Inspect, create, change, delete and activate SAP HANA repository objects A role based on this role template is required when you select the <i>Packages</i> tile in order to maintain SAP HANA repository packages in Web-based Development Workbench. |
| <code>sap.hana.xs.admin.roles::HTTPDestAdministrator</code> | Full access to HTTP destination configurations (display and edit) A role based on this role template is required when you register a system for a transport route. |
| <code>sap.hana.xs.admin.roles::RuntimeConfAdministrator</code> | Full access to the configuration settings for SAP HANA XS application security and the related user-authentication providers A role based on this role template is required when you register a system for a transport route. |

The the privileges in the following roles are required for SAP HANA Application Lifecycle Management 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> | This role includes all previous roles. |

17.3.3 Runtime Libraries

SAP HANA content for runtime libraries

- [HANA_XS_DBUTILS \[page 372\]](#)

17.3.3.1 HANA_XS_DBUTILS

This component provides content for the simplified consumption of SAP HANA database objects for XSJS.

Technical Details

| | |
|---------------------|--|
| Delivery unit | HANA_XS_DBUTILS |
| Prerequisites | None |
| Content type | Automated content |
| Content details | XSJS libraries |
| Target users | Developers using the libraries for more convenient access to SAP HANA database objects |
| Web application URL | None |

Key Features

The SAP HANA XS DB UTILITY LIBS component comes as set of XS JavaScript libraries that wrap the database interface of XS with JavaScript-native access methods and object representations:

- Invocation of SQL procedures as if they were JavaScript functions
- JavaScript CDS client and query builder

These libraries can be consumed only by applications deployed on XS. In other words, they cannot be accessed directly via HTTP from outside the XS container.

Roles

This component does not come with any roles. As the component simply wraps the standard XS database interface, the role definitions and authorizations of that interface directly apply.

17.3.4 Configuration

SAP HANA content for configuration

- [HANA_TA_CONFIG \[page 373\]](#)

17.3.4.1 HANA_TA_CONFIG

This component provides predefined configurations and dictionaries used by the SAP HANA text analysis engine and by text mining.

Technical Details

| | |
|---------------------|---------------------|
| Delivery unit | HANA_TA_CONFIG |
| Prerequisites | None |
| Content type | Automated content |
| Content details | Configuration files |
| Target users | SAP HANA developers |
| Web application URL | None |

Key Features

The Text Analysis Configuration component includes the following:

- Predefined configuration files containing text analysis options to be used when creating a full text index
- Predefined configuration files containing text mining options

Roles

This component does not come with any roles.

The Text Analysis Configuration component contains configuration files. It does not contain any executable software. Any user with permission to execute the SQL statement CREATE FULLTEXT INDEX can use the text analysis engine and text mining, which use the HANA_TA_CONFIG data.

17.3.5 Supportability and Development

SAP HANA content for supportability and development

- [HANA_IDE_CORE](#) [page 374]
- [HANA_XS_IDE](#), [HANA_XS_EDITOR](#) [page 375]
- [HANA_DT_BASE](#) [page 375]

17.3.5.1 HANA_IDE_CORE

This component provides a Web-based integrated development environment (IDE) that can be used to build and test development artifacts in SAP HANA. The SAP HANA Web-based Development Workbench is a quick and easy alternative to the SAP HANA studio for developing native SAP HANA applications in SAP HANA XS classic.

Technical Details

| | |
|---------------------|---------------------------------------|
| Delivery unit | HANA_IDE_CORE |
| Prerequisites | SAPUI5_1, SAP_WATT, HANA_XS_BASE |
| Content type | Automated content |
| Content details | SAP HANA applications |
| Target users | SAP HANA developers and support staff |
| Web application URL | http(s)://<host>:<port>/sap/hana/ide |

Key Features

The SAP HANA Web-based Development Workbench includes the following tools:

- Editor (IDE)
Inspect, create, change, delete, and activate SAP HANA repository objects or development artifacts such as database entities, XS JavaScript code, Web content (HTML, CSS, etc.), OData service definitions
- Catalog
Create, edit, execute, and manage SQL catalog artifacts
- Security
Manage users and roles, assign objects and manage security
- Traces
View and download traces for SAP HANA XS applications and set trace levels

Roles

The following roles are available with the SAP HANA IDE component. Users must have the privileges contained in one or more of these roles before they can use the component and its functions.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. 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.

| Role | Description |
|---|--|
| <code>sap.hana.ide.roles::Developer</code> | A combined user role which incorporates all the following roles and provides access to all tools |
| <code>sap.hana.ide.roles::EditorDeveloper</code> | Provides access to the IDE/Editor tool |
| <code>sap.hana.ide.roles::CatalogDeveloper</code> | Provides access to the Catalog tool |
| <code>sap.hana.ide.roles::TraceViewer</code> | Provides access to the Trace tool |
| <code>sap.hana.ide.roles::SecurityAdmin</code> | Provides access to the Security tool |

17.3.5.2 HANA_XS_IDE, HANA_XS_EDITOR

These components provide browser redirection to the SAP HANA Web-based Development Workbench.

i Note

The components HANA_XS_IDE and HANA_XS_EDITOR are available for downward compatibility reasons. They do not contain any functionality except redirection to the SAP HANA Web-based Development Workbench.

17.3.5.3 HANA_DT_BASE

This component provides the SAP HANA REST application programming interface (API).

The SAP HANA REST API allows development tools to access the SAP HANA repository and database catalog via HTTP(S) in a standard-compliant way. It builds upon the Eclipse Orion server API protocol version 1 on SAP HANA. For SAP- specific tools, the Orion server protocol has been extended with SAP HANA-specific features such as activation, change tracking, and database catalog search

Technical Details

| | |
|-----------------|-------------------|
| Delivery unit | HANA_DT_BASE |
| Prerequisites | None |
| Content type | Automated content |
| Content details | SAP HANA REST API |

| | |
|---------------------|---------------------------------------|
| Target users | SAP HANA developers and support staff |
| Web application URL | None |

Key Features

The SAP HANA REST API includes the following features:

- File and folder operations such as reading, writing, moving and deleting files and folders (packages)
It is possible to read and write file and folder metadata. Examples of SAP-specific metadata are the version, the activation time, and the activating user. In addition to the Orion standard, mass operations are available to get and set the metadata of many files with one request.
- Activation of repository objects
- Change tracking
- Handling of user preference data (for example, the SAP HANA Web-based Development Workbench and other development and support tools)
- Existence checks and search suggestions for metadata
These functions can be used to implement searching in the repository and in the database catalog, with auto-completion. The metadata suggestion request returns all resources that match a specified pattern

Roles

The following roles are available with the REST API component. Users must have the privileges contained in one or more of these roles before they can use the component and its functions. Additionally, users need the appropriate authorization on SAP HANA repository entities and catalog entities to be able to view or change repository or database content.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. 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.

| Role | Description |
|---|-------------------------------------|
| <code>sap.hana.xs.dt.base::restapi</code> | Allows users to access the REST API |

17.3.6 User Interface

SAP HANA content for user interface

- [HANA_UI_INTEGRATION_SVC, HANA_UI_INTEGRATION_CONTENT \[page 377\]](#)
- [SAPUI5_1 \[page 378\]](#)
- [SAP_WATT \[page 379\]](#)

17.3.6.1 HANA_UI_INTEGRATION_SVC, HANA_UI_INTEGRATION_CONTENT

These components provide SAP HANA UI Integration Services (UIS), which is a set of Eclipse-based tools and client-side APIs that enable you to integrate standalone SAP HANA client applications into Web-based application sites.

Technical Details

HANA_UI_INTEGRATION_SVC

| | |
|----------------------------|--|
| Delivery unit | HANA_UI_INTEGRATION_SVC |
| Prerequisites | None |
| Content type | Automated content |
| Content details | Database tables, views, stored procedures, UIs, HTML, JavaScript |
| Target users | Developers (design time) and end users (runtime) |
| Web application URL | None |

HANA_UI_INTEGRATION_CONTENT

| | |
|----------------------------|---|
| Delivery unit | HANA_UI_INTEGRATION_CONTENT |
| Prerequisites | HANA_UI_INTEGRATION_SVC |
| Content type | Automated content |
| Content details | Application sites and catalogs (.xsappsites and .xswidgets files) |
| Target users | XS developers |
| Web application URL | None |

Key Features

- For developers and designers: tools for creating content and designing application sites

- For end users: personalization capabilities and role-based access to application sites and their content

Roles

The following roles are available with the SAP HANA UIS components. Users must have the privileges contained in one or more of these roles before they can use the component and its provided services.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. 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.

| Role | Description |
|---|--|
| <code>sap.hana.uis.db::SITE_DESIGNER</code> | Create and edit standard and Fiori Launchpad applications sites and catalogs Assign permissions to standard and Fiori Launchpad application sites and their content |
| <code>sap.hana.uis.db::SITE_USER</code> | Access standard and Fiori Launchpad application sites and catalogs |

17.3.6.2 SAPUI5_1

This component provides SAP UI5, which is the library used by XSC-based Web applications and tools to implement the specific user interfaces.

All XSC-based Web applications delivered with SAP HANA such as SAP HANA Application Lifecycle Management and the SAP HANA Web-based Development Workbench rely on this delivery unit.

Technical Details

| | |
|---------------|-------------------|
| Delivery unit | SAPUI5_1 |
| Prerequisites | None |
| Content type | Automated content |

| | |
|----------------------------|---|
| Content details | Web content such as HTML, CSS, JavaScript |
| Target users | Used by XS-based Web applications |
| Web application URL | None |

Roles

Since this component provides purely Web content consumed by arbitrary Web applications, it is not protected by any specific mechanisms. Any browser can download the artifacts in this library.

17.3.6.3 SAP_WATT

This component provides the SAP WATT Web library, which is an additional Web library used by the SAP HANA Web-based Development Workbench. It contains additional Web content such as HTML, CSS, and JavaScript libraries to build Web development environments.

All XSC-based Web applications delivered with SAP HANA such as SAP HANA Application Lifecycle Management and SAP HANA Web-based Development Workbench rely on this delivery unit.

Technical Details

| | |
|----------------------------|--|
| Delivery unit | SAP_WATT |
| Prerequisites | None |
| Content type | Automated content |
| Content details | Web content such as HTML, CSS, JavaScript |
| Target users | Used by the SAP HANA Web-based Development Workbench |
| Web application URL | None |

Roles

Since this component provides purely Web content consumed by arbitrary Web applications, it is not protected by any specific mechanisms. Any browser can download the artifacts in this library.

17.3.7 Documentation

SAP HANA documentation delivered as SAP HANA content

- [HDC_* \[page 380\]](#)

17.3.7.1 HDC_*

These components provide product documentation for several Web applications delivered with SAP HANA. Users can access the documentation via a tile on the application homepage and from the Help menu if available.

Technical Details

Delivery unit

- HDC_XS_BASE
- HDC_IDE_CORE
- HDC_XS_LM

i Note

The following documentation DUs are available for downward compatibility reasons. They do not contain any content.

- HDC_ADMIN
- HDC_SYS_ADMIN
- HDC_SEC_CP
- HDC_BACKUP

For Web-based administration using the SAP HANA cockpit in SAP HANA 2.0, see the *SAP HANA Administration Guide*.

| | |
|----------------------------|---|
| Prerequisites | None |
| Content type | Automated content |
| Content details | HTML files, image files |
| Target users | Application users |
| Web application URL | <code>http(s)://<host>:<port>/public/sap/docs/hana</code> |

Features

Product documentation is available for the following Web applications delivered with SAP HANA:

- SAP HANA XS Admin Tools

- SAP HANA Application Lifecycle Management
- SAP HANA Web-based Development Workbench

Roles

These components do not come with any roles. Access to the content is controlled by the standard XS-application security mechanism, the .xsaccess file.

Important Disclaimer for Features in SAP HANA Platform



For information about the capabilities available for your license and installation scenario, refer to the Feature Scope Description (FSD) for your specific SAP HANA version on the [SAP HANA Platform webpage](#).

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

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

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

© 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/about/legal/trademark.html> for additional trademark information and notices.