Configure your ABAP Development System for Development of HDI Objects using the SAP HANA XS Advanced Cockpit

Version 1.3
## Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>First official release of this document</td>
</tr>
<tr>
<td>1.1</td>
<td>Updated <a href="#">Introduction and Preparation</a></td>
</tr>
<tr>
<td>1.2</td>
<td>Minor Corrections</td>
</tr>
<tr>
<td>1.3</td>
<td>Minor Corrections</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

1 INTRODUCTION AND PREPARATION ................................................................. 5
  1.1 General Concepts ................................................................................... 5
  1.1.1 Development Process ...................................................................... 7
  1.1.2 Configuration in a Nutshell ............................................................. 8
  1.2 Prerequisites ......................................................................................... 9
  1.3 Users ................................................................................................... 10
  1.4 Further Information ............................................................................ 10
2 CONFIGURATION .......................................................................................... 11
  2.1 Checking if HDI is Enabled ................................................................. 12
  2.2 Preparing the Communication from XSA to ABAP ............................ 13
    2.2.1 Creating Database Procedures Required by AS ABAP .................. 13
    2.2.2 Creating an SAP HANA Role ..................................................... 16
  2.3 Preparing the Communication from ABAP to XSA ............................ 18
    2.3.1 Creating Role Collections in XSA ............................................. 18
    2.3.2 Create User HTA_ADMIN_<ABAP_SID> in XSA ...................... 21
    2.3.3 Creating Destinations in ABAP .................................................. 25
    2.3.4 Configuring the Communication from ABAP to XSA ................. 32
  2.4 Configuring the Space ......................................................................... 37
    2.4.1 Creating XSA Space ABAP_SAP<SID> ....................................... 37
    2.4.2 Enable Development in Space ABAP_SAP<ABAP_SID> ............ 41
  2.5 Maintaining the Web IDE URL on ABAP ........................................... 42
  2.6 Creating the Shared Workspace ......................................................... 45
  2.7 Enable an ABAP Developer for HDI Object Development .............. 47
    2.7.1 Assigning Authorizations to the Developer on ABAP ............... 47
    2.7.2 Creating a Developer User on XSA ........................................... 47
    2.7.3 Assigning Developer Role in XSA ............................................ 49
    2.7.4 Adding Developer User to Space .............................................. 50
    2.7.5 Assigning the XSA User to the Web IDE Shared Workspace ....... 52
  3 TROUBLESHOOTING ................................................................................. 53
    3.1 Cannot open SAP HANA XS Advanced Cockpit .............................. 53
    3.2 Cannot open di-space-enablement-ui .............................................. 54
    3.3 Cannot Select ANONYM SSL Client (Anonymous) from the Drop-Down for SSL Certificate ......................................................... 55
    3.4 Error when Executing Connection Test for Destination .......... 55
      3.4.1 HTTPIO_PLG_CANCELED ......................................................... 55
      3.4.2 SSL Handshake Fails ................................................................. 57
      3.4.3 Connection to <host>:<port> failed: NIEHOST_UNKNOWN(-2) .... 57
      3.4.4 Connect to <host>:<port> failed: NIECONN_REFUSED(-10) ....... 57
      3.4.5 No Logon-pop-up .................................................................. 58
      3.4.6 Communication from ABAP to XSA failed ............................. 58
    3.5 Activate service / enable transaction OA2C_CONFIG .................. 59
    3.6 Error when Requesting OAuth 2.0 Tokens ..................................... 60
      3.6.1 HTTP Failure, Processing Failed, Invalid State, Invalid Timeout or Others ................................................................. 60
      3.6.2 Client Authentication Failed (e.g., Unknown Client, no Client Authentication Included, ...) ................................. 61
      3.6.3 The Sever is Refusing to Respond to the Request (HTTP 403 – Forbidden) .......................................................... 61
      3.6.4 Requested Resource could not be found (HTTP 404 – not found) .................................................................. 61
3.7 Error in existence check for XSA space
3.7.1 Error when executing GET_SPACE_ID command (http code is 0)
3.8 Space Enablement UI Does not Open, Error Message: Forbidden
3.9 Error when Creating Shared Workspace
3.9.1 Action CREATE HTTP REST CLIENT was canceled (http status code 0)
3.9.2 Action CREATE WORKSPACE(POST) was Canceled (Http Status Code 404)
3.9.3 Action CREATE WORKSPACE(POST) was Canceled (Http Status Code 500)
3.9.4 Invalid action
3.9.5 OAuth 2.0 Client Profile SAP_HTA_HDI is not Assigned to an OAuth 2.0 Client
4 AppENDIX
4.1 Configuring Secure Connection from ABAP to XSA
4.2 Using the program Define parameters of HTA-for-HDI Configuration
4.3 Abbreviations
This guide provides an overview about the process of developing HDI objects using SAP HANA Transport for ABAP for SAP HANA Deployment Infrastructure (HTA for HDI). The guide explains the prerequisites that must be fulfilled on your development systems before you can develop HDI objects using HTA for HDI, and the configuration that is required.

1 INTRODUCTION AND PREPARATION

This guide is only relevant for development systems (so systems where you would like to create objects and export them). None of the configuration steps described below has to be executed on test or production systems.

To be able to develop and transport HDI objects in the context of ABAP, you only need a SAP Web IDE for SAP HANA (Web IDE) connected to the ABAP development system. For this, you only need one XSA server that has the Web IDE installed. ABAP creates the HDI containers on the target systems that are part of the transports requests. Therefore, XSA is not needed on any target system (as long as you do not develop native XSA applications – to be able to transport these native XSA applications, you would have to configure CTS+ as HTA for HDI objects).

Please check SAP Note 2569651 - Configure your ABAP Development System for Development of HDI Objects before you start the configuration. The note contains updates and information that you need during the configuration.

This guide makes use of the SAP HANA XS Advanced Cockpit when it explains administration tasks that have to be done in XSA.

SAP HANA Platform 2.0 SPS 03 introduces the new and enhanced version of SAP HANA XS Advanced Administration tool called SAP HANA XS Advanced Cockpit.

The SAP HANA XS Advanced Administration tools component version 1.6.x is now deprecated. It is replaced by the SAP HANA XS Advanced Cockpit version 1.1.x. Please note that these version numbers are reflecting the version of SAP HANA on the XSA server – not the version of SAP HANA that you use on the ABAP system.

1.1 General Concepts

HDI (SAP HANA Deployment Infrastructure) is a service layer of the SAP HANA database that simplifies the deployment of SAP HANA Database artifacts.

HDI artifacts are developed using Web IDE that is available on premise as an SAP HANA XS, Advanced Model (XS advanced / XSA) application. Each ABAP development system that is used to develop or change HDI objects needs an instance of Web IDE assigned to it. Multiple ABAP systems can share an instance of the Web IDE. At runtime, no Web IDE instance is required. Therefore, test or production systems do not need a Web IDE and therefore no configuration.

To be able to develop HDI objects, you have to use the Web IDE (in the past – if you worked with SAP HANA Repository – the SAP HANA Studio or SAP HANA Web-based Development Workbench was used for this purpose).

As you might know ABAP developers work on one system (not locally). In Web IDE, this concept is realized using a shared workspace where all developers connect to and where all developed objects are stored.

NOTE

Using an ABAP system running on SAP HANA is absolute basic prerequisite for using HDI in ABAP. If your ABAP system does not run on SAP HANA, you cannot use any of the functionality (HDI or HTA for HDI) described in this guide.

The following picture shows the basic elements of ABAP and SAP HANA that are involved in HTA (SAP HANA Transport for ABAP) for HDI including the system IDs that we will use as example in this guide. The figure shows the systems (ABAP and XSA) that are involved in the development activities. Target systems (like test or production) only consist of an AS ABAP (like AT5) and its SAP HANA server (like HDB) used as database for ABAP. The next two chapters will explain the role and usage of these elements in the development process and how they have to be configured.
Destination required to create a shared workspace, add users to this workspace, create projects, and copy HDI objects from the shared workspace to HTA Repository in ABAP and vice versa. Later in the configuration, you will create this destination and name it XSA DEVX.
Destination needed by ABAP to automatically create external services (user-provided services) – see also connection. Later in the configuration, you will create this destination and name it XSA_ADMI.

These connections are used by:

1. DI Core (DevX) when a build in Web IDE is triggered to deploy the HDI artifacts to the HDI container in the database of the ABAP system.
2. Modeling Tools in Development Perspective in Web IDE (e.g. Calculation View Editor or Synonym Editor) to retrieve and display the definition of existing catalog objects (e.g. views, tables, …) of a schema/HDI container in the database of the ABAP system.
3. The Database Explorer of SAP Web IDE for SAP HANA so that Developers can test their HDI objects (e.g. Calculation Views)

These connections are created automatically by HTA as an external service (user-provided service) in the XSA space ABAP_SAP<SID> when a developer creates the first HDI namespace for a container in ADT or when a developer does the first check out of existing HDI objects of a container in ADT.

NOTE

In the picture above, ZDM and HDB run on two different SAP HANA servers. This is just an example to make clear what runs on which server. ZDM and HDB can also run on the same HANA server or in other words, ABAP on HANA and XSA can run on the same HANA server.”.

1.1.1 Development Process

In this guide, you will configure your system so that a developer can afterwards develop native SAP HANA artifacts (via HDI design-time objects) and use them in the context of ABAP. This is needed if the development options offered by AMDP (ABAP Managed Database Procedures - framework for managing and calling stored procedures or database procedures in AS ABAP) or ABAP CDS (ABAP Core Data-Services make it possible to define semantic data models on the central database of the AS ABAP) are not sufficient and you need to develop other objects like e.g. calculation views in SAP HANA that you need to use in ABAP.

Once the system is configured for HDI-object development, the development process for HDI-objects looks like this:

1. The developer (or an administrator) creates an HDI container in transaction SCTS_AMHC on the system AT5 in our example (there should not be that many containers in one system, so this should be done somehow centrally and not by each developer). A container is a triple of schema/user pairs: for runtime objects, design-time object storage, and object owner.
2. The developer creates a namespace in an HDI container using ABAP development tools (ADT / also known as ABAP in Eclipse – AiE). This will automatically create a project in the shared workspace on XSA (system ZDM on the left side in our example), this communication is illustrated by in the picture above (the destination is named XSA_DEVX). All users develop in the shared workspace. In addition, an external service that allows the communication from XSA to the HDI container in the SAP HANA database of AS ABAP is automatically created. Destination XSA_ADMI will be used for this. in the picture above.
3. The developer develops HDI objects in SAP Web IDE (a browser application on his client) and deploys them via the deployment infrastructure (DI Core in XSA) into the HDI container in the SAP HANA database on the ABAP server. This is done via the external service that was created in the previous step.
4. The developer checks in his objects into ABAP using ADT. During the check in the developer is also asked to collect the objects in a transport request in CTS. The objects are now copied from the shared workspace on XSA to the HTA repository on ABAP. Destination XSA_DEVX is used for reading the files from the shared workspace. As a result, the HTA repository contains HDI objects and ABAP managed HDI containers.
5. The developer releases the transport request.
6. (not part of the picture above) The administrator imports the transport request into the target system. Now, the objects that are part of the transport request are imported into the HTA repository during the import step Import and deployed from the HTA repository to the HDI container during the import step SAP HANA Deployment (in older releases, the step is named SAP HANA Repository Deployment). For the import process, there is no SAP Web IDE for SAP HANA required on the target system.
Before you can start developing, you have to set up all the required communications described above. This guide explains in detail how this is done.

### 1.1.2 Configuration in a Nutshell

In this guide, we will use the landscape shown in the picture in chapter 1.1 General Concepts as an example. We will use one system named ZDM. On this SAP HANA system, XSA and the SAP Web IDE for SAP HANA are installed.

In addition, we need AS ABAP which runs on an SAP HANA database. In our example, this system is named AT5.

HDI runs on SAP HANA of AT5 – not on ZDM.

The transaction SCTS_HTA_TOOLS provides links to many programs and documentation that you might find helpful when using HTA for HDI. In there, you can find e.g. links to tools that you need when configuring your system or checks that can help fixing issues.

The program Check HDI configuration that can be accessed from within SCTS_HTA_TOOLS, guides you through the configuration. It includes tests for most of the configuration steps described in this guide. We recommend that you use this program to check which configuration steps have to be done and whether they were successfully completed. All configuration chapters in this guide describe how to use the program for the respective step and how to deal with possible errors.

#### NOTE

The transaction SCTS_HTA_TOOLS is available if your system has at least 7.52 SP 4, 7.53 SP 2 und 7.54 SP 0 installed. If your system is not on any of these releases, yet, you can use SA 38 to execute the checks. Run the program SCTS_HTA_CHECK_CONFIGURATION. SCTS_HTA_TOOLS provides links to many other programs. If required, you can always use SA38 to execute the programs. You can find the names of the programs in the in this guide in the chapters where they are needed.

First step is to check that HDI in general is enabled in your system. In most cases, this should have been done by SUM/SWPM – see chapter Checking if HDI is Enabled.

As a next step, you have to execute two scripts that will create procedures to be able to create technical users and a role in SAP HANA database of AS ABAP. These are required later on during development to automatically create external services in XSA. These external services are later used by DI Core to communicate with the HDI container in the SAP HANA database of AS ABAP. Each external service contains a technical user that needs a dedicated role in HANA database of AS ABAP. This configuration is done in the chapter Preparing the Communication from XSA to ABAP.

After that, you need to make sure that the communication between ABAP (AT5 in our example) and XSA / SAP Web IDE (ZDM in our example) is possible. This is explained in the chapter Preparing the Communication from ABAP to XSA. Two http(s)-destinations need to be established:

- **XSA_ADMI**: ABAP needs to be able to communicate with the XSA controller to create external services. This service is created automatically by HTA when you do the first check out of objects from an existing HDI namespace or when you create the first HDI namespace for a container. The service contains the connection data for the HDI container of the SAP HANA database of AS ABAP including a technical user.
- **XSA_DEVX**: ABAP needs to be able to communicate with DI Core (DevX) to create a shared workspace, add users to this workspace, create projects, retrieve objects from the shared workspace to ABAP (Check-in) and push objects from ABAP to the shared workspace (Check-out).

The destinations XSA_ADMI and XSA_DEVX are configured in chapter Creating Destinations in ABAP. As XSA only allows connections with OAuth2 authorization the communication from ABAP to XSA requires the configuration of an OAuth2 client on AS ABAP This is done in chapter Configuring the Communication from ABAP to XSA.

The communication via these destinations requires a user. The respective technical user is created in chapter Create User HTA_ADMIN_<ABAP_SID> in XSA.

The shared workspace is created in chapter Creating the Shared Workspace.
The program *Check HDI configuration* is designed to guide you through the configuration steps. It shows if a configuration has already been completed and what has to be done next. In case of errors, the program gives hints how to solve them. The different tests in the program are named like the chapters in this guide.

All configuration actions described in this guide need to be performed on development systems only. That is, for systems where you want to develop and change HDI objects with SAP Web IDE. Transports are later on executed via the known mechanisms of TMS / CTS. You do not need to configure any additional transport routes or systems in transaction STMS.

**NOTE**
In this guide, we assume that the transport routes between the different ABAP systems are already in place. The required configurations are not described in this guide. If you need information on how to do this, refer to the SAP Help Portal at 'Configuring TMS'.

### 1.2 Prerequisites

- In this guide, we assume that you have configured transport routes for your ABAP systems in the Change and Transport System (transaction STMS). There are no additional configurations required concerning transport routes if you like to use HTA for HDI. Details about the change and transport system and how to set up transport routes are available on the SAP Help Portal at [https://help.sap.com/viewer/4a368c163b08418890a406d413933ba7/7.52.0/en-US](https://help.sap.com/viewer/4a368c163b08418890a406d413933ba7/7.52.0/en-US).

- For development systems in general, the following prerequisites apply:
  - You have an ABAP system (7.52 SP2 or higher) on an SAP HANA database version 2.0 SP01 or higher.
  - You have installed an XSA server 2.0 SP02 or higher. It is not required that XSA runs on the same system as the SAP HANA DB of your ABAP development system. You can use any XSA server that you already have (if it fulfills the release prerequisite) or install a new one with required version on a new/existing SAP HANA DB.
  - You have installed a Web IDE in the space SAP on the XSA system with a version that corresponds to your XSA version. Check the central release note for your SAP Web IDE to make sure that it fits with your XSA version (the SAP Web IDE is not needed for further configuration in this guide, but it is needed as soon as you start developing)
  - To be able to develop, you need to create a space (see chapter Creating XSA Space ABAP_SAPAT5). Use the default organization and only develop in spaces in the same organization where the SAP Web IDE is installed. SAP Web IDE for SAP HANA supports just one organization. See SAP Note [2642384](https://help.sap.com/viewer/2642384) for details.

- For the execution of this guide, we assume that you have installed / access to the following tools:
  - SAP GUI
  - SAP HANA Studio or SAP HANA database explorer (in SAP HANA Cockpit or SAP Web IDE for SAP HANA) to execute the SQL scripts. This guide uses the SAP HANA Studio. More information about executing SQL statements in SAP HANA Studio is provided in here: [https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/latest/en-US/e0c984fbbb5710149c7be707c74ad231.html](https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/latest/en-US/e0c984fbbb5710149c7be707c74ad231.html).
  - More information about the SAP HANA Database Explorer is available on the SAP Help Portal at [https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/latest/en-US/4d6d3c181a43a499e0aa964ee9c6a3d348.html](https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/latest/en-US/4d6d3c181a43a499e0aa964ee9c6a3d348.html) and [https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/latest/en-US/7a981c81f1b4196b243faeb4af5739.html](https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/latest/en-US/7a981c81f1b4196b243faeb4af5739.html).
  - XS Advanced Command-Line Client
  - SAP HANA XS Advanced Cockpit
  - Google Chrome is recommended for all tasks in this guide that use a browser-based UI.
1.3 Users

For this guide, you will need different users:

- One user with system administrative authorization on the ABAP side. We do not recommend that you use user DDIC, but similar authorizations are required. By using a named user, you can make sure that, later on, you can find out which user did which configurations. In this guide, the user is called MY_ABAP_ADMIN or is referred to as ‘your user’

For using the program Check HDI configuration, the user must have the following authorizations:
  - S_TRANSPORT: ACTVT=03
  - S_OA2C_USE:PROFILE=*:ACTVT=*;

For the OAuth configuration, the user must have the following authorizations:
  - S_TCODE:TCODE=OA2C_CONFIG;
  - S_OA2C_ADMIN:ACTVT=*;
  - S_OA2C_USE:PROFILE=*:ACTVT=*;

Go to your role management in ABAP (Transaction PFCG) to add the authorizations to the role or profile that is assigned to you (the user who does the configuration). If you need more information about transaction PFCG, please take a look at the documentation in the SAP Help Portal at https://help.sap.com/doc/7bc79ea6726810148a4b1a83b0e91070/latest/en-US/nwbc_110_guide_en.pdf. Search for the chapter ‘Role Maintenance in PFCG’

\* NOTE

The OAuth configuration has to be done in each client of the ABAP system that you want to use to develop HDI objects. So, you need this authorization in each client to be able to do the configuration.

- You need the password of the user SYSTEM for the relevant tenant database of your SAP HANA system which is used by ABAP. These credentials are used in some SQL scripts that you will execute later in this guide.

- One user with administrative authorizations on the SAP HANA XSA side. We do not recommend that you use XSA_ADMIN. But similar authorizations are required. By using a named user, you can make sure that later on, you can find out which user did which configurations. In this guide, this user is called MY_XSA_ADMIN.

- One technical user used for communication between ABAP and DI Core (DevX) / XSA. In this guide, the user is called HTA_ADMIN_AT5. It will be created in XSA in chapter Create User HTA_ADMIN <ABAP SID>. This user is required for the destinations 1 and 2 in the picture in chapter 1.

- In addition, everyone who shall develop HDI objects, needs users in ABAP and on XSA. These users do not have to exist before you can start the configuration. You can create the user-ids later on. In this guide, the user on XSA is called DEVELOPER. But you might need more than one developer user – one per developer. For more information about configurations required for these users, see Enable an ABAP Developer for HDI Object Development.

1.4 Further Information

In this chapter, you will now execute all the configuration steps that will enable the HDI development in ABAP for your developers. To learn more about developing HTA for HDI, take a look at the SAP Help Portal at Using HTA for HDI. When you finished the configuration, you can also check our tutorials.

In some steps, you will be asked to use the program Check HDI configuration. This program is designed to help you with the configuration. It is not required that you use it for the configuration. Nevertheless, we recommend that you do so. The program contains checks for all configuration tasks that have to be done on the ABAP side. The checks are named like the chapters in this guide.

NOTE
We recommend that you always execute the program Check HDI configuration first if any issue comes up with using HTA for HDI.
If e.g. a developer reports that he cannot check out or check in any more, run the program to find out if the configuration is still ok. As a last check the program also creates some objects and deletes them again – many developer tasks are thereby included in this program and the test result can give you hints what might be wrong.

We recommend that you execute the configuration tasks in the order that is used in this guide. The test program executes the tests in the same order. So, checks in the program should turn into green from top to bottom if you execute the configuration tasks as described in this guide. The error messages and the corresponding long texts in the program will also offer help if the configuration did not succeed.
In addition, you can use the program if you cannot finish the configuration at once to find out where you stopped or to hand over the configuration to someone else to finish.
The program does not support or check the configuration parts that have to be done on XSA side.
When you call the program for the first time, it will look like this:
You can click on the callouts in front of each Test and error message to get more information on what is tested and what needs to be done. The first line for each test shows the title of the chapter in this guide that is relevant for the respective test.

2.1 Checking if HDI is Enabled

HDI should have been enabled by SWPM (SAPINST) / SUM during your system setup or update for all systems that are afterwards on SAP NetWeaver 7.52. By this, the service diserver should have been started and the database schema _SYS_DI should have been created.

To be able to use HTA for HDI, a container group is needed. By default, SWPM(SAPINST) / SUM uses the same name for the container group as the ABAP schema name, e.g. SAPABAP1 or SAP<SID> in older releases.

To check if this is the case in your system, do the following:

1. Log on to your ABAP system with a user who has assigned the authorizations to use the program Check HDI configuration. See chapter Users for details.

2. Open transaction SCTS_HTA_TOOLS

If the transaction SCTS_HTA_TOOLS does not exist, start the program SCTS_HTA_CHECK_CONFIGURATION in transaction SA38 and continue with step 4. Use this program whenever the guide talks about „Test HDI Configuration“

3. Execute the program Test HDI Configuration.
4. The first check Test: ‘Checking if HDI is Enabled’ should be green. You can open the details for Check if HDI is Enabled to get more details about what it means that HDI is enabled.

![Image of HDI configuration test]

If this is not the case, check SAP Note 2603193 Enable HTA for HDI on ABAP systems after the installation with SWPM. If this doesn’t solve the issue, open an incident for SAP in the component BC-INS-SWPM, if you newly installed the system, and in component BC-UPG-TLS-TLA, if the system was upgraded.

2.2 Preparing the Communication from XSA to ABAP

You need to create SAP HANA procedures on the SAP HANA database of AS ABAP that will be used later to create technical users for the External Services (User-Provided Services) that are required for the communication from Web IDE on XSA to the HANA DB of AS ABAP – even if they run on one SAP HANA database. The technical users will be created automatically later, as soon as developers start developing HDI artifacts. This is illustrated by the destination [3] in the picture above. The technical users that will be created (later on) with the help of the SAP HANA procedures need a dedicated role which you will create on the SAP HANA that is used by ABAP (not on the SAP HANA of the XSA system if it is a different one). You will perform both actions by executing SQL statements on the tenant database of the SAP HANA system that is used by ABAP.

2.2.1 Creating Database Procedures Required by AS ABAP

To create the required SAP HANA procedures on SAP HANA database of AS ABAP, you have to execute SQL statements on the relevant tenant database of your SAP HANA system which is used by ABAP. Before execution, adapt the script with your own data by following the instructions at the beginning of the script.

1. Execute the program Test HDI Configuration from transaction SCTS_HTA_TOOLS. The test Test: ‘Creating Database Procedures Required by AS ABAP’ should show an error.
2. Open the SAP HANA Studio and log on to the tenant DB and not the system DB of your SAP HANA system underlying ABAP.

3. Right-click the tenant database and choose Open SQL Console.

4. Download the file CreatingDatabaseProceduresRequiredByASABAP.SQL from SAP Note 2569651 and save it to your PC.

5. Load the script into the SQL console – right-click in the empty space of the SQL console and choose Open File… and select the file that you saved in the previous step or copy the code into the SQL console.

6. Adapt the script according to the instructions at the beginning of the script:
   a. Set the correct password for user SYSTEM by replacing <manager> with the password of the SYSTEM user.

   **NOTE**
   If the user SYSTEM is deactivated in your system, either activate it to execute the SQL scripts or replace user SYSTEM by another user with similar privileges.
b. Set the correct name for the user SAPSID by replacing <USER>.
To get the correct user name, go to your ABAP system. Go to System → Status. In the Database Data area, it is displayed as value for the User field.

![Database Data](image)

Set the correct name for the ABAP schema by replacing <SCHEMA> with the name of the ABAP schema in this system.
To get the correct user name, go to your ABAP system. Go to System → Status. In the Database Data area, it is displayed as value for the Schema field.

![Database Data](image)

7. Select all lines of the script (CTRL+A) and execute them at once.

8. Make sure that no errors are reported. If errors are reported which are not related to the replacements that you did, check SAP Note 2569651 Configure your ABAP Development System for Development of HDI Objects.

9. Execute the program Test HDI Configuration from transaction SCTS_HTA_TOOLS. The test Test: 'Creating Database Procedures Required by AS ABAP' should now be ok.
2.2.2 Creating an SAP HANA Role

The technical users that will be created (later on) with the help of the procedures of the previous chapter need a dedicated role. In this chapter, you are going to create this role on the SAP HANA system that is used by ABAP (not on the XSA system if it is a different one).

To do this, you proceed as above (download the file with the SQL statements and open it in an SQL console of the tenant database of your SAP HANA system and execute it after having adapted it with your own data by following the instructions at the beginning of the script).

\[\text{NOTE}\]
In our example, this needs to be done on HDB (which is the SAP HANA DB underlying the ABAP system) and not on ZDM (which is the database for XSA).

1. Execute the program Test HDI Configuration from transaction SCTS_HTA_TOOLS. The test Test: ‘Creating an SAP HANA Role’ should show an error.

2. Use an empty SQL-Console (Right-click the tenant database and choose Open SQL Console to open a new one):

3. Download the file CreatingAnSAPHANARole.sql from SAP Note 2569651 and save it to your PC or copy the code from the Appendix: Script for creating SAP HANA role.
4. Load the script into the SQL console — right-click in the empty space of the SQL console and choose Open File… and open the file that you saved in the previous step or copy the code into the SQL console.

5. Adapt the script according to the instructions at the beginning of the script: Set the correct password for user SYSTEM by replacing <manager> with the password of the SYSTEM user.

6. Select all (CTRL+A) and execute the script. If errors are reported for GRANT statements, you can ignore them. Depending on the HANA version there are more/less views/table/procedures available. The GRANT statements might therefore try to set permissions for views/table/procedures that do not exist on your system. For other errors, or if you are in doubt, create a ticket and assign it to component BC-CTS-HTA.
7. Execute the program *Test HDI Configuration* from transaction SCTS_HTA_TOOLS. The test *Test: 'Creating an SAP HANA Role'* should now be ok.

![Test HDI Configuration](image)

2.3 Preparing the Communication from ABAP to XSA

For the communication from ABAP to DevX / XSA a technical user is needed on XSA with dedicated roles. We will now at first create the required role collections. After that, we will create the user (see chapter Create User HTA_ADMIN <ABAP_SID>) and assign the role collections to the user.

2.3.1 Creating Role Collections in XSA

**NOTE**

This configuration step has to be executed on XSA. As the program *Test HDI Configuration* from transaction SCTS_HTA_TOOLS is testing configurations on AS ABAP only, it cannot test whether this configuration is already done and is working fine.


1. Open the XS start page. The URL is [https://<host>:<port>](https://<host>:<port>) where port is 3<XSA/HDB instance>30" (e.g. 30230 for XSA/HDB instance 02). Click on xsa-cockpit. If this does not work, check the chapter Cannot open SAP HANA XS Advanced Cockpit to open the SAP HANA XS Advanced Cockpit.
2. Open a browser window and enter the URL for the XSA cockpit
3. Log in with your XSA admin user.

![XS start page](image)
4. Go to Security → Role Collections.

5. Click on New Role Collection.

6. Enter a name, e.g. WEBIDE_DEVELOPER, and a description, e.g. WEBIDE_DEVELOPER and choose Save.

**NOTE**
If you already have your own role collections you do not have to create new ones, but you can assign the application roles mentioned in the following steps to your own role collections.

7. The new role collection appears in the list. Click on the Name of your newly created role collection.

8. Click on Add Role

9. From the dropdown menus in the Add Role dialog box, choose the following entries:

<table>
<thead>
<tr>
<th>Application Identifier</th>
<th>Role Template</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>webide!1</td>
<td>WebIDE_Developer</td>
<td>WebIDE_Developer</td>
</tr>
</tbody>
</table>

You have to do this top down – the entries in the second and third drop-down will only appear after something was selected for the first and second drop-down. Click Save.

The result looks like this.

10. You have to add a second role to this role collection. Click Add Role

11. From the dropdown menus in the Add Role dialog box, choose the following entries:

<table>
<thead>
<tr>
<th>Application Identifier</th>
<th>Role Template</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>sap-xsac-hrtt!1</td>
<td>xsac_hrtt_developer_template</td>
<td>xsac_hrtt_developer_template</td>
</tr>
</tbody>
</table>
12. The Role Collection now looks like this:

<table>
<thead>
<tr>
<th>Application Identifier</th>
<th>Role Name</th>
<th>Role Template</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>sap-xsec-http11</td>
<td>xsac_httpt_developer_template</td>
<td>xsac_httpt_developer_template</td>
<td></td>
</tr>
<tr>
<td>webidei1</td>
<td>WebIDE_Developer</td>
<td>WebIDE_Developer</td>
<td></td>
</tr>
</tbody>
</table>

13. Click on Home

14. Repeat steps 5-8 to create the role collection WEBIDE_ADMINISTRATOR. Use the following values as Name and Description:

<table>
<thead>
<tr>
<th>Role Collection Name</th>
<th>WEBIDE_ADMINISTRATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Collection Description</td>
<td>WEBIDE_ADMINISTRATOR</td>
</tr>
</tbody>
</table>

From the dropdown menus in the Add Role dialog box, choose the following entries:

<table>
<thead>
<tr>
<th>Application Identifier</th>
<th>Role Template</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>webidei1</td>
<td>WebIDE_Administrator</td>
<td>WebIDE_Administrator</td>
</tr>
</tbody>
</table>

The result looks like this:

2.3.2 Create User HTA_ADMIN_<ABAP_SID> in XSA

1. NOTE

This configuration step has to be executed on XSA. As the program Test HDI Configuration from transaction SCTS_HTA_TOOLS is testing configurations on AS ABAP only, it cannot test whether this configuration is already done and is working fine.

1. Go to User Management. (If you closed the XSA Admin Cockpit, check Creating Role Collections in XSA to find out how to open this UI.)
2. Choose New User.

3. Fill in the required data for the user HTA_ADMIN_<ABAP_SID> (HTA_ADMIN_AT5, in our example, as this user is used on the ABAP server to communicate with DevX / XSA). Note that the password that you enter here, is an initial password that you will need to change in a later step. Choose Create.
4. Click on the Assign Role Collections for the user that you just created.

5. Click on Add.

6. In the Role Collections dialog box, select the WEBIDE_DEVELOPER and the WEBIDE_ADMINISTRATOR role collections that you created before, and the role collection
7. The three role collections were assigned to the user. The role XS_USER_PUBLIC is assigned to each user automatically. Choose Save.

8. Since this user is a technical user that is only used in the background, log on once with this user to the SAP HANA XS Advanced to change the initial password. You will need the new password in a later step so make sure that you remember it or note it down in a password safe-tool.
2.3.3 Creating Destinations in ABAP

ABAP is the leading system when developing HDI objects for ABAP on HANA applications. Therefore, ABAP needs to be able to connect to XSA / DevX and ABAP needs to configure XSA to connect back to the SAP HANA database of AS ABAP. In detail, the destination XSA_ADMI is needed by ABAP to create user-provided services and XSA_DEVX is required to create a shared workspace, add users to this workspace, create projects, and copy objects from the shared workspace to ABAP and vice versa.

In this chapter, we will use a standard naming for these two destinations. If you stick to this naming, no further action is required. If you choose other names, you have to make the system aware of these names by entering the destination names in the program Define parameters of HTA-for-HDI Configuration. Check the chapter Using the program Define parameters of HTA-for-HDI Configuration for details.
1. Go to transaction SCTS_HTA_TOOLS and execute the program *Test HDI Configuration*. The test *Test: ‘Creating Destinations in ABAP’* should show an error.

2. Go to transaction SM59.

3. Click on *Create* to create a new destination.
4. Enter XSA_ADMI as RFC Destination. Enter Connection Type G and e.g. Destination to XSA Controller as Description 1.

Press <enter>.

5. Confirm the pop-up HTTP connections may not be secure if it is shown.
6. Execute the command `xs -v` to get host and port of the `controllerEndpoint`.

![Command output]

7. Enter the values as `Host` and `Port` on the tab `Technical Settings` of your destination `XSA_ADMI`. Use the values that you got as output when executing the command `xs -v`. Make sure that you enter the server-name and domain as `Host` (but leave out `https`).

Usually, the port looks like the following: 3<XSA/HDB Instance>30 (e.g. 30230 for XSA/HDB instance 02)
Set SSL to Active. 
If a pop-up is shown saying that user and password are going to be deleted, confirm it.

Choose **ANONYM SSL Client (Anonymous)** from the drop-down for SSL Certificate.

If you get a warning **SSL client PSE ANONYM does not exist**, or if you can’t select the **SSL Certificate ANONYM SSL Client (Anonymous)**, the SSL-client configuration was not yet done for this ABAP system. See troubleshooting section **Cannot Select ANONYM SSL Client (Anonymous) from the Drop-Down for SSL Certificate** to do this configuration.

9. Save your settings.
10. Execute the Connection Test.

11. The value for Status HTTP Response should be 200.

If the result is not 200 or if you receive a pop-up showing an error message check the troubleshooting section Error when Executing Connection Test for Destination.

12. You need a second destination named XSA_DEVX. To create it, you are now going to copy the destination XSA_ADMI.
If you don’t see the details of destination XSA_ADMI any more, double-click on this destination.
Choose Connection \(\rightarrow\) Copy.

13. Enter XSA_DEVX as Destination and click OK.
14. Switch to change mode.

15. To get the host and the port, execute the command `xs target -s SAP` first to make sure that you work in the SAP space and then execute the command `xs app di-core`. Copy the value that you can see for `urls`.

If you get the error message **App "di-core" not found in space "SAP"**, switch to the space where you have installed the Web IDE using the command `xs target -s <your Space>` and then execute the command `xs app di-core`.

16. Enter a description. Check that host fits to the value for `urls` that you copied. Enter the port and save the destination. Leave all other settings unchanged.
17. Execute the connection test for this destination. The following dialog popup will appear.

On this dialog popup choose **Cancel**. You will get the following response:

'401' is not an error in this case.
If the above dialog popup does not appear, see chapter **Error when Executing Connection Test for Destination**.

18. Go to transaction SCTS_HTA_TOOLS and execute the program **Test HDI Configuration**. The test **Test: 'Creating Destinations in ABAP'** should now be ok.

### 2.3.4 Configuring the Communication from ABAP to XSA

Since the communication between AS ABAP and SAP HANA XS advanced model uses the OAuth 2.0 protocol, you have to configure this communication between AS ABAP and the backend of SAP Web IDE (DevX) and AS ABAP and the XS controller. This is done by maintaining the OAuth2 client profile **SAP_HTA_HDI**.

This configuration has to be done in each client that you use for developing HDI objects.
1. Go to transaction SCTS_HTA_TOOLS and execute the program Test HDI Configuration. The test Test: ‘Configuring the Communication from ABAP to XSA’ should show errors.

If you see the error message ICF service OA2C_CONFIG is not active, you have to activate the service in transaction SICF. Take a look at chapter Activate service / enable transaction OA2C_CONFIG.

2. In the XS advanced command-line client, execute `xs create-service-key devx-uaa devx-uaa-ABAP`.

   You could see OK or failed: Service key with name “devx-uaa-ABAP” already exists as result of this step. Both are fine.

3. Execute `xs sk devx-uaa devx-uaa-ABAP`.

   Copy the values for `clientid` and `clientsecret` and paste them to e.g. notepad because you will need them for the following steps.
4. Go to your SAP GUI for system AT5. Open transaction OA2C_CONFIG.

NOTE

OAuth Clients created in transaction OA2C_CONFIG are specific to the ABAP client that you are logged on to. So, make sure that you do this configuration in the correct ABAP client (the one that you will use for developing HDI objects – the same that you use also for developing ABAP workbench objects). If you use more than one client of your ABAP system for developing HDI objects, you have to do this configuration in OA2C_CONFIG in each client.

NOTE

If a new browser window opens up showing Forbidden and Service cannot be reached as title, check the chapter Activate service / enable transaction OA2C_CONFIG.

5. Click on Create.

6. Choose SAP_HTA_HDI as OAuth 2.0 Client Profile, enter this value as Configuration Name, as well. Add the OAuth 2.0 Client ID that you found in step 3.

Depending on your release, it might be that you cannot enter a Configuration Name. This is also fine.
7. Fill in the details in the order listed here – make especially sure that you fill in the Client Secret at last.
   a. Authorization Endpoint: Execute the command `xs -v` to get the URL.
   
   ![CAUTION: Make sure that you enter the URL without 'https://'. Add /oauth/authorize at the end.]

   ![Client version: xs v0.1718-cons
   Server version information:
   name = XS Controller
   support = http://service.sap.com/message
   build = v0.1718-cons
   version = 1
   softwareVersion = 0.1718.1729.49
   contentVersion = 0.1718.1729.49
   overallState = READY
   user = <not set>
   description = SAP XS Runtime on premise
   configuration:
<table>
<thead>
<tr>
<th>configuration</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorizationEndpoint</td>
<td><a href="http://30232/usa-security">http://30232/usa-security</a></td>
</tr>
<tr>
<td>allowDebug</td>
<td>true</td>
</tr>
<tr>
<td>acceptEncoding</td>
<td>gzip, x-gzip</td>
</tr>
<tr>
<td>limits</td>
<td>memory:&lt;not set&gt;, apps:&lt;not set&gt;, app uris:&lt;not set&gt;, services:&lt;not set&gt;</td>
</tr>
<tr>
<td>databaseType</td>
<td>HANA MULTI</td>
</tr>
<tr>
<td>addRole</td>
<td>false</td>
</tr>
</tbody>
</table>
   | addRole | false |](image)

   ![b. Token Endpoint: Use the same URL as for Authorization Endpoint but replace authorize at the end by token.

c. Access Settings: Set Client Authentication to Basic, set Resource Access Authentication to Header Field and Selected Grant Type to Resource Owner Password Credentials.

d. Client Secret: Enter the value that you found in step 4.](image)
8. Click Save.

9. As a result, you should see a message that OAuth 2.0 Client sb-webide/i1 is saved successfully. The list of existing OAuth 2.0 Clients should now contain the client that you just created.

10. Scroll down until you can see the button Request Tokens… Click on it.

11. Enter the user HTA_ADMIN_<ABAP_SID> (HTA_ADMIN_AT5 in our example) and the user’s password. Click OK.

12. As a result, the circle in front of Request Tokens should change from red to green. Now the communication between ABAP and XSA is established.

NOTE
If you use more than one client of your ABAP system for developing HDI objects, use the same user and password for each client where you configure HDI. Reason is that there will be only one shared workspace created per ABAP system with the user that you enter in here and only this user has admin privileges in this shared workspace.

CAUTION: the token is only valid for 90 days (30 days in older versions of Web IDE). You have therefore to request tokens every thirty days. If the token has expired, the developer will see the error message ‘Refresh Token has Expired at the Server (Execute OAuth 2.0 Access Token Request Again)’ in the HDI namespace editor in ADT. Check Guided Answers for details: https://gad51588421.us2.hana.ondemand.com/dtp/viewer/#/tree/2311/actions/30100:30571:30106/?version=current
13. Go to transaction SCTS_HTA_TOOLS and execute the program Test HDI Configuration. The test Test: ‘Configuring the Communication from ABAP to XSA’ should now be ok.

2.4 Configuring the Space

More information on creating and managing spaces is also available on the SAP Help Portal at https://help.sap.com/viewer/1a8e7ab05a2e4119b02b702121142215/2.0.02/en-US/a6a17b3bb99d4280b5c354d0d74dab.html ‘Managing Spaces for Development’

2.4.1 Creating XSA Space ABAP_SAP<SID>

In this chapter, you are going to create a space on your XSA system. For each ABAP development system, exactly one space must be configured in XSA. In this space, ABAP will create all external services (user-provided services) that will later be used by di-core (DevX) or other Web IDE tools/services to connect to HDI containers / schemas in the database of the ABAP system. This space will be assigned to each Web IDE project that is created by the ABAP system. The default naming of the space is ABAP_SAP<SID>. AT5 is the SID of the ABAP system that we use as an example in this guide.

NOTE
On your system, <SID> is replaced by the SID of your ABAP system. If you decided to use another naming than the recommended standard for your space, then this test will later show the name of your space. Additional configuration steps are required if you decide to go for another space name – see step 4 in this chapter for details.

1. Go to transaction SCTS_HTA_TOOLS and execute the program Test HDI Configuration. The test Test: ‘Creating XSA Space ABAP_SAP<SID>’ should show errors.
2. Go to the home page of SAP HANA XS Advanced Cockpit. Make sure that you logged in as MY_XSA_ADMIN. (If you closed the UI, check the beginning of the chapter Creating Role Collections in XSA to find out how to open this UI.)

Click on Organizations and then on the existing organization, AT5ORG in our example.

3. Click on New Space.

4. In the New Space dialog box, enter ABAP_SAP<ABAP_SID>, ABAP_SAPAT5 in our example, and choose Save.

\* NOTE

We recommend that you use this standard naming. If you do not use this standard space name ABAP_SAP<ABAP_SID> in here, you have to add the space name to the configuration of HTA for HDI in the program Define parameters of HTA-for-HDI Configuration. See Using the program Define parameters of HTA-for-HDI Configuration.
5. The new space is shown. Click on it.

6. Choose Members.

7. Choose Add Members
8. Enter `HTA_ADMIN_<ABAP_SID>` as **User IDs**. Make sure that **Space Developer** is selected in the section **Assign Roles** and choose **OK**.

9. You can assign additional users to this space according to your needs. Every XSA user who should be able to develop in Web IDE for the ABAP system (AT5 in our example) has to become a member of the space. So, if you already know the user names of your developers in XSA, you can also add them right now. You can also add users later on at any point in time.

10. Go to transaction `SCTS_HTA_TOOLS` and execute the program **Test HDI Configuration**. The test **Test: 'Creating XSA Space ABAP_SAP<SID>'** should now be ok.
2.4.2 Enable Development in Space ABAP_SAP<ABAP_SID>

1. **NOTE**
   This configuration step has to be executed on XSA. As the program Test HDI Configuration from transaction SCTS_HTA_TOOLS is testing configurations on AS ABAP only, it cannot test whether this configuration is already done and is working fine.

1. Open the XS start page. The URL is https://<host>:<port> where port is 3<XSA/HDB instance>30” (e.g. 30230 for XSA/HDB instance 02). Click on di-space-enablement-ui. If this does not work, check the chapter Cannot open di-space-enablement-ui.
2. You should see just the space that you created before. Click Enable for this space.

3. Wait until you can see a message The builder has been deployed successfully. It might be that you have to scroll down in the Log area.

4. Come back to this UI after each update of Web IDE or DevX and check whether you have to update the space: if the status for the space is Outdated, execute an Update.
2.5 Maintaining the Web IDE URL on ABAP

You need to define the URL to your Web IDE to allow your developers later on to easily navigate to Web IDE: the URL is used in ADT to enable a button named Open in Web IDE in the namespace editor. With this, the developers can easily reach the right workspace in the Web IDE from ABAP development tools.

1. Go to transaction SCTS_HTA_TOOLS and execute the program Test HDI Configuration. The test Test: ‘Maintaining the Web IDE URL on ABAP’ should show an error.

2. We assume that you use the Web IDE on the system where XSA is installed. Execute the command xs app webide on your system where XSA is installed (ZDM in our example) and copy the value for urls.

   If you get the error message xs app webide not found in space <xyz>, you may have to switch first to the space in which the Web IDE is installed (default space is SAP) before you can execute the command xs app webide. To switch to the space SAP, execute the command xs target -s SAP.

3. Go back to transaction SCTS_HTA_TOOLS.

   If the transaction SCTS_HTA_TOOLS does not exist, start the program SCTS_HTA_CONFIGURATION in transaction SA38 and continue with step 5.
4. Click on the program Define parameters of HTA-for-HDI Configuration.

5. The program shows the standard values for all parameters except for the Web IDE URL. As URL of SAP Web IDE for SAP HANA enter the urls value that you copied in step 2. Leave all other input fields as they are. The default values are shown – you only need to change them if you did not follow the standard configuration and naming as described in this guide. Details about when else to use this program are described in chapter Using the program Define parameters of HTA-for-HDI Configuration.

6. If you did not use standard naming in the previous configuration steps, you have to change the values for the respective parameters as well. (For Name of XSA space, see Creating XSA Space ABAP_SAP<ABAP_SID>, for destinations, see Creating Destinations in ABAP. CreateUserProcedure is part of the SQLscript that you executed in Creating Database Procedures Required by AS ABAP.)

7. Execute the program after you filled in the data that you need.
8. You should see a result similar to the following (depending on the data that you maintained in the previous steps). For each parameter, you can see the current value and the default value.

![HTA-for-HDI Configuration](image)

9. Go to transaction ‘SCTS_HTA_TOOLS’ and execute the program 'Test HDI Configuration'. The test: 'Maintaining the Web IDE URL on ABAP' should now be ok.

![Test HDI Configuration](image)
2.6 Creating the Shared Workspace

When working with HTA for HDI, all developers use a shared workspace. With this, they can work on the same objects. They don't use any local workspace when working in SAP Web IDE for SAP HANA.

1. Go to transaction SCTS_HTA_TOOLS and execute the program Test HDI Configuration. The test Test: ‘Creating the shared Workspace’ should show an error.

2. In transaction SCTS_HTA_TOOLS, execute Create Workspace in SAP Web IDE / Add User Rights in Workspace.

Transactions and Utility Programs for SAP HANA Transport for ABAP

3. Choose Get existing workspace/create workspace as Action. You do not have to fill in any value for User ID. A User ID is not needed to execute this Action. Click on Execute.
4. The workspace is created, and the *Workspace id* is shown. Click OK.

If you do not get the workspace shown but get errors, see [Error when Creating Shared Workspace](#).  

5. Go back again and execute the program *Test HDI Configuration*. This was the last configuration step. The program *Test HDI Configuration* will now also execute a complete functional test for HTA for HDI. Some objects will be created and deleted in the background. Therefore, it will take some time until the program is executed, this time.

6. The tests *Test: ‘Creating the shared Workspace’* and *Test: ‘Checking the Configuration’* should now also be ok – you can open the details of *Test: ‘Checking the Configuration’* to find out what had been tested including which objects had been created for test purposes and had been deleted afterwards again.
2.7 Enable an ABAP Developer for HDI Object Development

Each developer who works on both parts (ABAP and XSA) has to have a user on ABAP and on XSA. The user-IDs of the developers can be different on both sides (ABAP and XSA). The ABAP Developer must have authorizations to use the destination from ABAP to XSA to be able to Check-in / Check-out HDI objects in ADT. The XSA developer user must be assigned as developer to the space that you created before, he has to have Web IDE developer rights (using the role collection that you created before), and he must have authorizations on the Web IDE shared workspace.

As it is not possible to set up SSO between HTA for HDI and XSA, all the assignments have to be done for every user who shall have developer permissions in XSA and ABAP.

2.7.1 Assigning Authorizations to the Developer on ABAP

A new authorization object enables your developers to use the destinations that you just configured.

Go to your role management in ABAP (Transaction PFCG) to add the authorization S_OA2C_USE:PROFILE=*;ACTVT=* to the role or profile that is assigned to your developers.

If you need more information about transaction PFCG, please take a look at the documentation in the SAP Help Portal at „Role Maintenance in PFCG“

https://help.sap.com/viewer/9737050ef01843f19572591b42128f1b/7.latest/en-US/4c5bdc2a97817511e10000000a42189b.html

2.7.2 Creating a Developer User on XSA

If accounts for your developers already exist on XSA, continue with chapter Assigning Developer Role in XSA.

To create a new developer user on XSA, proceed as follows:

1. Open the SAP HANA XS Advanced Cockpit on the system where XSA is installed (in our example, this is system ZDM) and go to User Management.

Go to the beginning of chapter Creating Role Collections in XSA to learn how to get the URL of the SAP HANA XS Advanced Cockpit.
2. Click on New to create a new user.

![Image of user management interface]

**NOTE**
If the user already exists in SAP HANA, you can click on *Migrate SAP HANA User* to transform this user into an XSA user.

3. Enter all required fields in the dialog box. In our example, the user is called DEVELOPER. Afterwards, choose *Create*.

![Image of user creation dialog]

The DEVELOPER user was created.
### 2.7.3 Assigning Developer Role in XSA

Each user who shall develop has to have the role collection `WEBIDE_DEVELOPER` assigned that you created in chapter Creating Role Collections in XSA.

1. You are in the User Management in SAP HANA XS Advanced Cockpit on the system where XSA is installed (in our example, this is system ZDM).
2. Click on Assign Role Collections for the user that you just created.

3. Click on **Add**.

4. Select the role collection `WEBIDE_DEVELOPER` and click **OK**.
5. The role collection WEBIDE_DEVELOPER is now in the list. Click on Save.

2.7.4 Adding Developer User to Space

Each developer has to be assigned to the correct space.

1. In the SAP HANA XS Advanced Cockpit, go to Organizations and click on your organization.

2. Click on your space (ABAP_SAPAT5 in our example)
3. Click on Members.

4. Click on Add Members.

5. Enter the User IDs of the user that you just created. Make sure that Space Developer is selected in the section Assigned Roles and click OK.
6. The user appears in the list of Members.

7. Repeat these steps for any user who will develop. You do not have to do this for all users right now. Just re-execute the steps described in this chapter whenever you need to add a user.

2.7.5 Assigning the XSA User to the Web IDE Shared Workspace

You can repeat this step whenever a new developer user needs to develop HDI objects and thus needs to work on the shared workspace. It is not required to assign all the possible developers right now.

1. Open transaction SCTS_HTA_TOOLS.

2. Execute the program Create Workspace in SAP Web IDE / Add User Rights in Workspace.
3. Choose the **Action Add user with collaborator right**. Enter the **User ID** of your developer on XSA (**DEVELOPER** in our example) and click **Execute**.

![Create Workspace/Add User Rights in Workspace](image)

**CAUTION**
The **User ID** is case sensitive. User IDs in SAP HANA (and therefore XSA) are usually stored in upper case. So, make sure that you enter the **User ID** in same case writing as in SAP HANA.

4. As a result, you should see the message **Action completed** at the bottom left.

![Action completed!](image)

A tutorial on the SAP Help Portal guides you through the steps for creating some basic HDI objects:

**Tutorial: Developing and Consuming HDI Objects in ABAP.**

3 **TROUBLESHOOTING**

In this chapter, you can find possible errors and their solutions. The title of the sub-chapter is the error message that you can see – or the most ‘speaking’ part of it. This chapter contains issues that could come up when you configure HTA for HDI.

If you encounter issues during development, please refer SAP Note 2682272.

**1 NOTE**

We recommend that you always execute the program **Test HDI Configuration** first if any issue comes up with using HTA for HDI.

If e.g. a developer reports that he cannot check out or check in any more, run the program to find out if the configuration is still ok. As a last check the program also creates some objects and deletes them again – many developer tasks are thereby included in this program and the test result can give you hints what might be wrong.

3.1 **Cannot open SAP HANA XS Advanced Cockpit**

To open the SAP HANA XS Advanced Cockpit, do the following:

1. **Open the URL** https://<server>:<port of app>. To find the correct port, you can use the xs advanced command-line client of your XSA installation.

2. **Log on to the XSA server with an OS user.**

   Call the command **xs login -u <MY_XSA_USER> -s SAP**
   (replace `<MY_XSA_USER>` with the ID of your XSA admin user).

   Enter the password for your XSA Admin user.

   With this, you are logged in to XSA with your admin user and work in the space SAP.

   If you are asked for an API URL: you don’t have to enter anything if you work directly on the XSA server.
3. Use the command `xs -v`.

   Look for the Registered service URLs for `xsa-cockpit` and copy the URL to a browser window.

   ![Registered service URLs](image)

   If you don’t know this command line tool or can’t find the `xsa-cockpit`-URL, ask your XSA administrator.

### 3.2 Cannot open di-space-enablement-ui

To find the URL for the di-space-enablement-ui and to open it, do the following:

1. Log on to the XSA server with an OS user.

   Call the command `xs login -u <MY_XSA_USER> -s SAP` (replace `<MY_XSA_USER>` with the ID of your XSA admin user).

   Enter the password for your XSA Admin user.

   With this, you are logged in to XSA with your admin user and work in the space SAP.

   If you are asked for an API URL: you don’t have to enter anything if you work directly on the XSA server.

2. Use the command `xs -v`.

   Look for the URL `di-space-enablement-ui` and copy it.

   ![Registered service URLs](image)

   If you don’t know this tool or cannot find the URL, ask your administrator.

3. Open a new browser window and enter the URL that you just copied. Log on with the user `MY_XSA_ADMIN`.

   ![SAP HANA XS Advanced](image)

   If you get an error message when logging on to the DI Space Enablement UI, your user may not have the appropriate authorization. See troubleshooting section *Space Enablement UI Does not Open, Error Message: Forbidden*. 
3.3 Cannot Select ANONYM SSL Client (Anonymous) from the Drop-Down for SSL Certificate

ISSUE: you get a warning SSL Client PSE Default is not available or you are not able to select ANONYM SSL Client (Anonymous) from the drop-down for SSL Certificate when you create the destinations in chapter Creating Destinations in ABAP.

SOLUTION: check transaction STRUST and make sure that the section SSL client SSL Client (Anonymous) is created (shown as a folder) and certificates are ok. See Appendix Configuring Secure Connection from ABAP to XSA for details.

3.4 Error when Executing Connection Test for Destination

3.4.1 HTTPIO_PLG_CANCELED

ISSUE: when you execute the connection test for the destinations created in chapter Create Destinations in ABAP You get the message HTTPIO_PLG_CANCELED
SOLUTION: Check whether http(s)-services are started correctly:
Open transaction SMICM

Choose Goto → Services.

Check the both HTTP and HTTPS-Protocol are marked as Active.
3.4.2 SSL Handshake Fails

ISSUE: When checking the destination XSA_ADMI in chapter **Creating Destinations in ABAP** you get the following error message:

![SSL handshake failure](image)

SOLUTION: Make sure that SSL is set up correctly. For details have a look at chapter **Configuring Secure Connection from ABAP to XSA**.

3.4.3 Connection to <host>.<port> failed: NIEHOST_UNKNOWN(-2)

ISSUE: When executing the connection test for destination XSA_DEVX in chapter **Creating Destinations in ABAP** you get the following error message:

![Connection failure](image)

SOLUTION: Make sure that the target host and port number is set correctly. Check especially that you did not copy https:// into the **Target**-field.

3.4.4 Connect to <host>:<port> failed: NIECONN_REFUSED(-10)

ISSUE: When executing the connection test for destination XSA_DEVX in **Creating Destinations in ABAP** you get the following error message:

![Connection refused](image)

SOLUTION: Make sure that the target host and port number is set correctly.
3.4.5 No Logon-pop-up

ISSUE: when you execute the connection test for the destination XSA_DEVX, there is no log-on pop-up shown. The Status HTTP Response is 404 (and not 401)

SOLUTION: Check that you entered the correct port.

3.4.6 Communication from ABAP to XSA failed

ISSUE: after you configured the destination XSA_DEVX, the program Test HDI Configuration shows the error message Communication from ABAP to XSA failed for the test Test: ‘Configuring the Communication from ABAP to XSA.’

SOLUTION: execute the command xs app di-core and make sure that the port shown is the one that you entered in the destination XSA_DEVX in SM59. It can happen that the port changes if you re-installed the Web IDE.
ISSUE after having configured the communication from ABAP to XSA, you get the following error message when executing the program Test HDI Configuration:

SOLUTION: (only if you are on Windows) implement SAP Note 2677819.

3.5 Activate service / enable transaction OA2C_CONFIG

ISSUE: When the test ‘Test: Configuring the Communication from ABAP to XSA’ is executed, you receive an error message that the service OA2C_CONFIG is not active.
Or when you try to open transaction OA2C_CONFIG, you receive the error message Forbidden in the browser window (with the title Service cannot be reached)
SOLUTION: Check whether the required service is activated
Log on to your ABAP system and open transaction SICF. Check that the service OA2C_CONFIG is activated. If it is not activated yet, activate it.

Take a look on the SAP Help Portal at Active Services in ICF for more details.
3.6 Error when Requesting OAuth 2.0 Tokens

3.6.1 HTTP Failure, Processing Failed, Invalid State, Invalid Timeout or Others.

ISSUE:

SOLUTION: check that you entered the URLs for the endpoints correctly. Especially, make sure that you did not enter http://, https:// or spaces at the beginning of the URL and that you used correct port of the UAA of your XSA installation. Especially check what is marked in the following screenshot:
3.6.2 **Client Authentication Failed (e.g., Unknown Client, no Client Authentication Included, ...)**

Issue: You get the following error popup when you execute Request Token for an OAuth client.

![Error Popup](image)

Solution: This error is either related to a wrong password or Client Secret. First try to execute Request Token again and make sure to use correct password. If this still fails, edit the OAuth 2 client configuration by reentering the Client Secret. Then save the configuration and retry Request Token.

3.6.3 **The Server is Refusing to Respond to the Request (HTTP 403 – Forbidden)**

ISSUE: You get the following error popup when you execute Request Token for an OAuth client.

![Error Popup](image)

SOLUTION: Check whether the user HTA_ADMIN_AT5 needs to change his password by trying to logon to XS Advanced Administration and Monitoring Tools (Admin UI) and change the password there. Then retry to request OAuth 2.0 Tokens in transaction OA2C_CONFIG. This has to be done in each client where you develop.

3.6.4 **Requested Resource could not be found (HTTP 404 – not found)**

ISSUE: You get the following error popup when you execute Request Token for an OAuth client

SOLUTION: Check the URLs for Authorization Endpoint and Token Endpoint. Make sure that they follow the following format:

```
<server>:<port>/uaa-security/oauth/authorize
<server>:<port>/uaa-security/oauth/token
```
3.7 Error in existence check for XSA space

3.7.1 Error when executing GET_SPACE_ID command (http code is 0)

ISSUE: after having configured what is described in chapter Creating XSA Space ABAP_SAPAT5, the program Test HDI Configuration shows the error message 'Error when executing GET_SPACE_ID command'.

SOLUTION: Check that the Host and the port for XSA_ADMI are set correctly. Check chapter Creating Destinations in ABAP (especially step 5) for details.

3.8 Space Enablement UI Does not Open, Error Message: Forbidden

ISSUE: When you try to open the Space Enablement UI, you get the error message 'Forbidden'.

SOLUTION: To log on to the DI Space Enablement UI, the logon user needs the WEBIDE_ADMINISTRATOR authorization. If you get the error message Forbidden, assign the WEBIDE_ADMINISTRATOR authorization to the logon user.

3.9 Error when Creating Shared Workspace

When creating the shared workspace (Chapter Creating the Shared Workspace) you get a sequence of Information Popups with error messages. Usually the first popup looks like this:

The following popups depend on the error situation and are shown in the following sub chapter.

3.9.1 Action CREATE HTTP REST CLIENT was canceled (http status code 0)

ISSUE: When creating the shared workspace, you get the following popup:

SOLUTION: Check that you entered the correct host name and port number for destination XSA_DEVX. Make sure you used the port that is shown when you execute the command xs app di-core on your system. Do not use the port shown in this guide. Ports differ from system to system. See chapter Creating Destinations in ABAP, especially steps 15 and 16.
3.9.2 Action CREATE WORKSPACE(POST) was Canceled (Http Status Code 404)

ISSUE: When creating the shared workspace, you get the following popup:

SOLUTION: Check that you entered the correct host name and port number for destination XSA_DEVX. Make sure you used the port that is shown when you execute the command `xs app di-core` on your system. Do not use the port shown in this guide. Ports differ from system to system. See chapter Creating Destinations in ABAP, especially steps 15 and 16.

3.9.3 Action CREATE WORKSPACE(POST) was Canceled (Http Status Code 500)

ISSUE: When creating the shared workspace, you get the following popup:

SOLUTION: Check the OAuth Client configurations whether all settings are correct. As first step open transaction OA2C_CONFIG, in the table select the line with `Configuration Name = SAP_HTA_HDI` and `OAuth 2.0 Client ID = sb-webide/i1` and check that the button Request Token shows a green circle. If this is not the case switch to edit mode and press Request Token and provide username as `HTA_ADMIN_<ABAP_SID>` with the correct password. See also chapter Configuring the Communication from ABAP to XSA.

3.9.4 Invalid action

ISSUE: You receive a message like the following:

SOLUTION: Check that you entered the correct host name and port number for destination XSA_DEVX.
3.9.5 OAuth 2.0 Client Profile SAP_HTA_HDI is not Assigned to an OAuth 2.0 Client.

ISSUE: When creating the shared workspace, you get the following popup:

SOLUTION: Check the OAuth Client configurations whether the required clients were created. In SAP GUI start transaction OA2C_CONFIG and make sure you find the OAuth 2.0 Client IDs, sb-webide11 and sb-admin.

If one or the other is missing, see chapter Configuring the Communication from ABAP to XSA and create them.

4. APPENDIX

4.1 Configuring Secure Connection from ABAP to XSA

Your ABAP server needs to connect to the XSA server with secure http connections and therefore needs to accept the SSL certificate of the XSA server. To enable ABAP to trust XSA certificates you need to either import the XSA certificate or one of his parents in the certification path. If this is already done for your system, you can skip this chapter.

1. Log on to your XSA server via the command line tool xs. Use the database user that was used when installing XSA.
2. Execute the command `xs login -u <MY_XSA_ADMIN> -s SAP` (replace `<MY_XSA_ADMIN>` with the ID of your XSA admin user)

Enter the password of your user.

```
badadm$ host/sap/HDB/HDB02> xs login -u SAP
API_URL: https://<server>:<port>/sapbans
USERNAM: XSA_ADMIN
PASSWORD: 
Authenticating...
ORG: ATSGG
SPACE: SAP
API endpoint: https://<server>:<port> (API version: 1)
User: XSA_ADMIN
Org: ATSGG
Space: SAP
```

3. Execute the command `xs -v`. Search for the URL for `authorizationEndpoint` and copy it.

```
badadm$ wsdlimporter://usr/sap/HDB/HDB02> xs -v
Client version: xs v0.1718-cons
Server version information:
  name = XS Controller
  support = http://service.sap.com/message
  build = v0.1718-cons
  version = 1
  softwVersion = 0.1718.1718.289443
  contentVersion = 0.1718.1718.289443
  overallState = READY
  user = <not set>
  description = SAP XS Runtime on premise
  authorizationEndpoint = https://<server>:<port>/uaa-security
  appService: <not set>
  allowDebug = true
  acceptEncoding = gzip, x-gzip
  limits = memory: <not set>, apps: <not set>, app urls: <not set>, services: <not set>
  usage = memory: <not set>, apps: 35, app urls: <not set>, services: 33
  databaseType = HANA Multi
  databaseInfo = HDB 2.0.0.0.0.0.1492839354

Registered service URLs:
  deploy-service = https://<server>:<port>/uaa-security/deploy-service
  xsa-admin = https://<server>:<port>/uaa-security/xsa-admin
  hrtt-service = https://<server>:<port>/uaa-security/hrtt-service
  hrtt-core = https://<server>:<port>/uaa-security/hrtt-core
  webide = https://<server>:<port>/uaa-security/webide
```

4. Extend the URL with `/oauth/token`. The final URL looks like this: `https://<server>:<port>/uaa-security/oauth/token`

In the reminder of this chapter, we use google chrome. If you use another browser, some features (like finding the certificate) work differently.

Scroll down to the end of this chapter to learn how to get the certificate when using another browser.
5. Copy the URL to a browser window. Click **Cancel** on the authentication pop-up.

6. Press F12. Click on **Security**, then on **View certificate** and then in the pop-up window on **Certification Path**. Make sure that the **Certificate Status** says 'The certificate is ok'.
7. Click on Details and choose Copy to File…

8. Go through the wizard. Click Next.
9. Leave the settings as they are (or choose the format you need) and click Next.

10. Browse for the location where you would like to store the file and name it `<file>.cer`.
11. Check that everything is correct and click Finish.

12. Click ok

13. Log on to your ABAP system and open transaction STRUST
14. If the section *SSL client SSL Client (Anonymous)* is marked with a red cross, continue with the following steps (15 to 17). If SSL in general is already configured (and you can see a folder symbol in front of *SSL client SSL Client (Anonymous)*) continue with step 18.

15. Switch to change mode (Choose Display <> Change). Right click on *SSL client SSL Client (Anonymous)*. Choose Create
16. Leave everything as it is in the following pop-up or change the settings according to your needs. Check SAP Note 510007 for details. Click on Continue.

17. You should see a success message at the bottom of the screen. The section SSL client SSL Client (Anonymous) should now show a folder.
18. Now the folder SSL client SSL Client (Anonymous) should exist but there is still no certificate shown in the Certificate List area.

19. Choose Certificate → Import
20. Browse for the location where you stored the certificate before and click on *Continue*.

21. Allow the system to access the file.
22. Choose Add to Certificate List.
23. Now, the certificate is shown in the *Certificate List*. Click on *Save*.

24. Make sure the node *SSL client SSL (Standard)* is shown as a folder and has sub-nodes. If not, switch to change mode (Choose Display <-> Change). Right click on *SSL client SSL Client (Standard)*. Choose *Create* and *Confirm* the popup.

**NOTE**

To download the certificate in different browsers, do the following:

- **Internet Explorer** (Run as administrator):
  In the same tab, where you opened the link: Start the navigation path *Security report*.
Click on Certificate
Choose the tab Details
Click on Copy to File…
Choose Next
On the next screen, leave the settings as they are or choose the format that you need
Choose Next
Browser for the folder where you would like to store the certificate. Enter <your file name>
Choose Next
Choose Finish
Confirm the pop-up that the file was saved successfully
Click OK

- **Firefox**
  In the same tab, where you opened the link, start the navigation pathShow site information
  Click on the Arrow
  Click on More information
  Click on Security
  Click on View Certificate
  Click on Details
  Click on Export…
  Click on Save

- **Safari**
  In the same tab, where you opened the link, choose the padlock symbol in the URL bar
  Choose Show Certificate.
  Drag the certificate symbol to your desktop.
4.2 Using the program Define parameters of HTA-for-HDI Configuration

You can define several HDI parameters by using the program Define parameters of HTA-for-HDI Configuration. In a standard configuration, you only need this program to define the URL for the Web IDE (see chapter Maintaining the Web IDE URL on ABAP).

If you did use other names for the destinations (chapter Creating Destinations in ABAP), the XSA space (chapter Creating XSA Space ABAP_SAP<ABAP_SID>) or when creating the database procedures (chapter Creating Database Procedures Required by AS ABAP) than the ones suggested in this guide, you have to enter these names in the program as well.

To open the program, first open transaction SCTS_HTA_TOOLS. Then choose Define Parameters for HTA-for-HDI Configuration.

Make sure that you enter the parameters in the correct case.

4.3 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUM</td>
<td>Software Update Manager</td>
</tr>
<tr>
<td>SWPM</td>
<td>Software Provisioning Manager</td>
</tr>
<tr>
<td>XSA</td>
<td>SAP HANA XS Advanced</td>
</tr>
<tr>
<td>IDE / Web IDE</td>
<td>SAP Web IDE for SAP HANA</td>
</tr>
<tr>
<td>AMHC</td>
<td>ABAP Managed HDI Container</td>
</tr>
<tr>
<td>HDI</td>
<td>SAP HANA Deployment Infrastructure</td>
</tr>
<tr>
<td>HTA</td>
<td>HANA Transport for ABAP</td>
</tr>
<tr>
<td>ADT</td>
<td>ABAP Development Tools</td>
</tr>
<tr>
<td>AIE</td>
<td>ABAP in Eclipse</td>
</tr>
<tr>
<td>CDS</td>
<td>Core Data Services</td>
</tr>
<tr>
<td>AMDP</td>
<td>ABAP managed database procedures</td>
</tr>
</tbody>
</table>