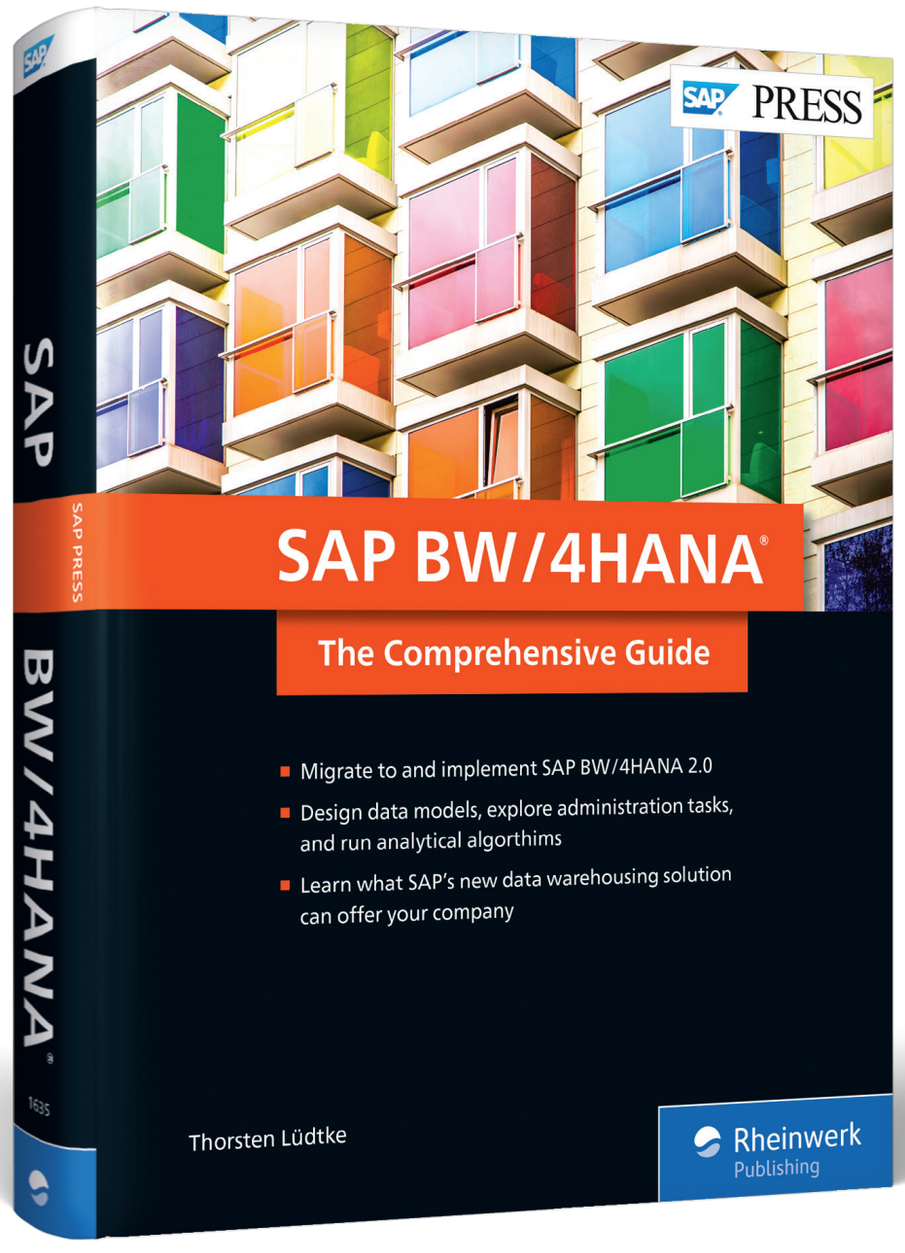
SAP PRESS
First-hand knowledge.
Rheinwerk Publishing



Browse the Book

In this sample chapter, you'll learn to install, configure, and connect essential SAP BW/4HANA tools.

"Tools"

Contents

Index

The Authors

Thorsten Lüdtke, Marina Lüdtke

SAP BW/4HANA 2.0: The Comprehensive Guide

665 Pages, 2021, $89.95
ISBN 978-1-4932-1635-2

www.sap-press.com/4544

Chapter 4
Tools

The design of management information systems most often began with asking what could be done with the information that was already there, not with asking what decisions were being made, and what information would be helpful in making them. [...] The designers had not learned the first lesson of living in an information-rich world: that a major task of an effective information system is to filter information, not proliferate it.

—Herbert A. Simon, The New Science of Management Decision, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

There are a number of tools surrounding SAP BW/4HANA's core components. Developers mostly work in the Eclipse-based SAP HANA Studio (Section 4.1). It contains a number of perspectives for the integrated development of mixed-models at the level of the application server and within SAP HANA. Administrators occasionally use the old SAP GUI frontend, but increasingly work with SAP BW/4HANA cockpit. Third-party developers use the SAP HANA client tools (Section 4.2) to connect their products to the SAP HANA database. More and more analysts use SAP Analytics Cloud as their primary analysis client, which first needs to be connected the SAP BW/4HANA system to retrieve data (Section 4.3). Power users typically run their own models and SAP Data Warehouse Cloud, connected to the SAP BW/4HANA system, provides them with a playground unmanaged by the central IT department (Section 4.4). Data scientists will enjoy SAP Data Intelligence as it provides them with raw access to unaltered data sets and pipelines to automate repetitive tasks (Section 4.5). Finally, operational business users benefit from embedded analytics, partly based on comprehensive models in SAP BW/4HANA (Section 4.6).

In this chapter, you'll learn to install, configure, and connect these tools.

4.1 SAP HANA Studio Installation and Configuration

The SAP HANA Studio is still the major development and administration environment for SAP HANA. Like SAP GUI for SAP BW/4HANA, it provides you with a direct connection to your SAP HANA instance and helps you configure and troubleshoot SAP HANA issues that a web-based frontend cannot guarantee, often for security reasons. With the new *SAP BW/4HANA cockpit*, however, more and more functions are being shifted to a web-based tool. This has the advantage of enabling you to automatically use the latest

frontend tools, and individual clients are no longer responsible for their updates. Currently, however, you are still dependent on SAP HANA Studio for many administration and modeling tasks.

SAP HANA Studio is based on the freely available Eclipse IDE. In the *perspectives,* you can open numerous tools in parallel. For example, you can program in ABAP in one perspective and SQLScript in another. The Eclipse IDE is available for Windows as well as for Mac and Linux. SAP GUI on non-Windows systems can be problematic for graph-based interfaces, as *SAP GUI for Java* has restrictions for displaying them. However, graphical overviews for SAP BW/4HANA, such as process chains and transformations, have been integrated into Eclipse.

The title SAP HANA Studio is a bit misleading because most developers don't install the prepackaged SAP HANA Studio, but its individual components that are required for development purposes. At the time of this book's publication, Eclipse version 2020-06 is the latest supported by the ABAP development environment and the SAP BW modeling tools. We'll demonstrate their installation in the following sections. This methodology hasn't changed over the past few years, and thus, we believe it'll be valid for a few more years to come. Simply visit *http://s-prs.co/v454428* to view the latest supported software stack, as shown in Figure 4.1.

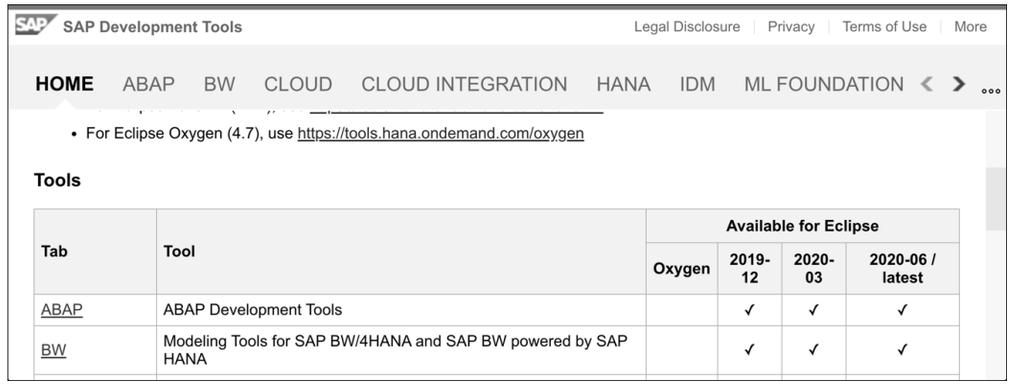


Figure 4.1 SAP HANA Studio Eclipse Download Sites

4.1.1 Connecting the SAP BW/4HANA Server and SAP BW Modeling Tools via a Remote Desktop

You'll learn how to install the SAP BW modeling tools and set up an SAP BW/4HANA project in upcoming sections. In SAP Cloud Appliance Library, however, you can start working with SAP BW/4HANA directly and without prior installation effort. To do this, you connect to the Windows server also provided in the cloud using the remote desktop protocol (RDP). You can find the link **Connect** to this server in SAP Cloud Appliance Library (see Figure 4.2). Then log on to the Windows server as an administrator with your master password.

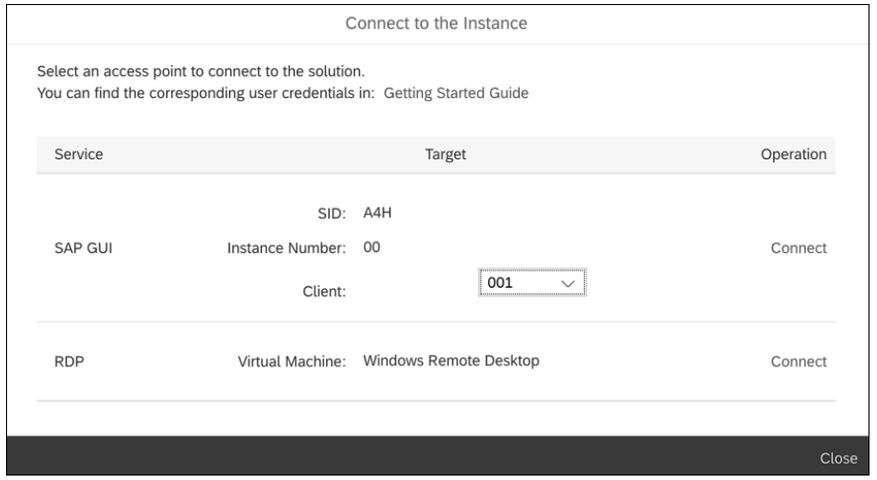


Figure 4.2 Connection to the Windows Frontend Server in SAP Cloud Appliance Library

In this section, we'll first restrict ourselves to describing how to set up a connection to the SAP BW/4HANA server using the SAP BW modeling tools, so that we can then show you how to create SAP BW/4HANA modeling objects.

In SAP HANA Studio, the **BW Modeling** perspective (the SAP BW modeling tools) is connected to the SAP BW/4HANA server, as shown in Figure 4.3.

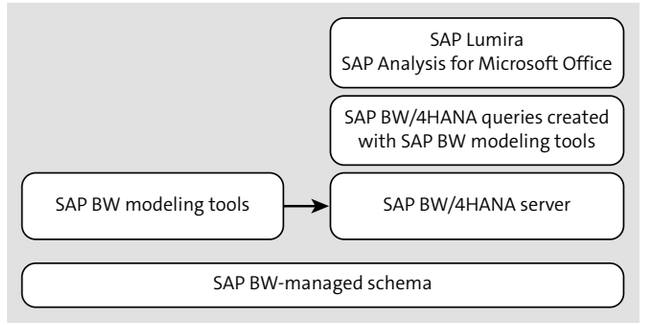


Figure 4.3 SAP BW-Managed Scheme

After you have connected to the Windows server using the remote desktop protocol or have successfully installed Eclipse and the SAP BW modeling tools, you can connect to your SAP BW/4HANA instance:

1. To do this, open the SAP HANA administration console. You can open this perspective in different ways:
   - In the Eclipse main menu, choose **Windows • Perspective • Open Perspective • Other** and scroll down in the window that opens to the **SAP HANA Administration Console**.

– You can also open this window directly by clicking the **Open Perspective** button on the far right of the main menu. Depending on the number of modeling tools installed in Eclipse, this list can become quite long. We therefore prefer the following, third, option.
– To the left of the **Open Perspective** button, there is an input field with the text **Quick Access**. You can use it to start a search for the term you entered. Enter "SAP" here to see all SAP-relevant perspectives and commands, as shown in Figure 4.4.

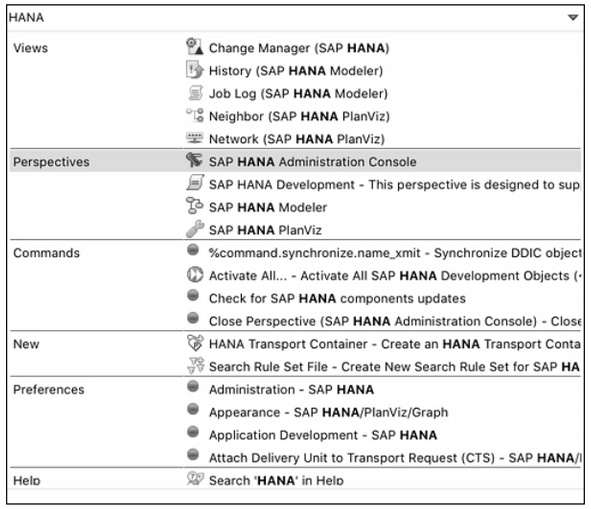


Figure 4.4 Selecting the SAP Administration Console in SAP HANA Studio

2. The **SAP HANA Administration Console** perspective opens. Since you will probably be working on a whole series of parallel developments, it is best to create a development directory right at the start. In this book, the client is the Redalli Holding Company (Redalli, for short). So click the **New** button on the far left of the main menu and select the **Folder** entry. In the field **Folder Name** enter "Redalli" and confirm with **Finish** (see Figure 4.5).

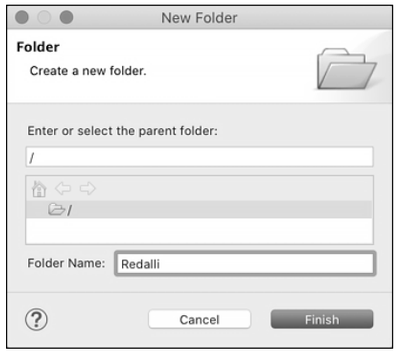


Figure 4.5 Creating a Project Folder in SAP HANA Studio

3. On the far left, under **Systems,** you now have the option of connecting to an SAP HANA system. However, if you want your shortcuts to be added to the correct project folder, right-click on the project folder. You can now connect to your SAP BW/4HANA system using the context menu that appears (see Figure 4.6).

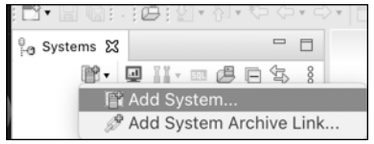


Figure 4.6 Linking a System in SAP HANA Studio

4. In the following window, you specify the connection parameters of your SAP BW/4HANA installation in AWS (see Figure 4.7). These are the external IP address of the Linux server on which the SAP HANA database is running and its instance number. You can simply copy the IP address from the **Information** section of your SAP BW/4HANA instance (see Chapter 1). According to the user guide, the instance number of the SAP HANA database is 02.

Attention: Slash in Product Name

When describing the system, you should always avoid using the slash in the name SAP BW/4HANA, since it is reserved as a character for file paths at the UNIX level (see SAP Note 2354199).

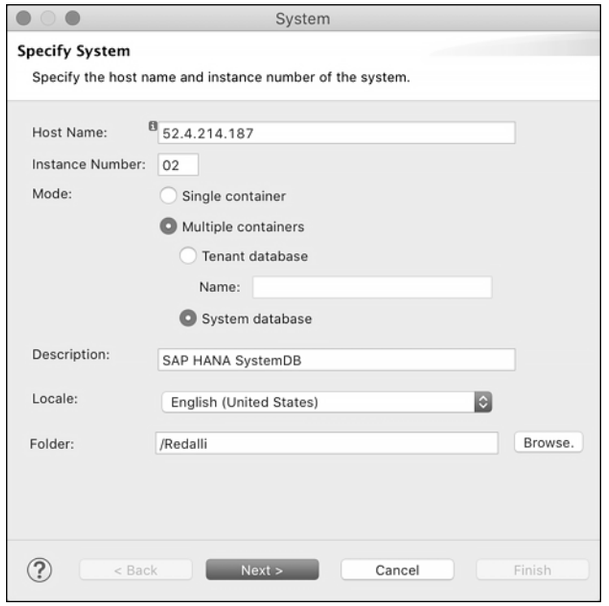


Figure 4.7 Add Connection Parameters for the SAP HANA System Database

5. Choose **Multiple containers** and **System database** as the mode.

> **Multiple Containers**
>
> Multiple containers indicates that, next to the system database, there can be one or more tenant databases. This enables applications to run virtually separated, but in parallel, on different databases. The system database manages the tenant databases.

6. Now confirm your entries by clicking **Next** to enter the ID of the system user SYSTEM of your SAP HANA database in the final screen (see Figure 4.8).

Figure 4.8 Creating Connection Properties to the SAP HANA System Database

7. Also, check the box **Store user name and password in secure storage** to avoid having to enter the system ID again and again later. Leave the SSL checkbox empty for now.
8. Complete the installation of the connection by clicking **Finish**.

You're now connected to the system database managing the other tenant databases. On SAP Cloud Appliance Library your SAP BW/4HANA data dictionary resides in the first tenant. To access it, repeat the steps above for the SAP BW/4HANA tenant database.

1. On the far left, under **Systems,** you now have the option of connecting to your tenant database **A4H**. This time you want your shortcut to be added to the correct project folder. Thus, right-click on the project folder **Redalli**. You can now connect to your SAP BW/4HANA tenant database using the context menu that appears (see Figure 4.9).

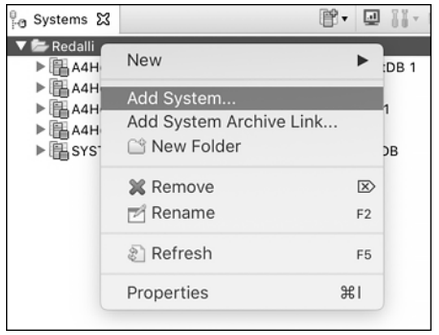


Figure 4.9 Linking a Tenant Database in SAP HANA Studio

2. In the following window, you specify the connection parameters of your SAP BW/4HANA installation in the AWS again (see Figure 4.10). These are the same external IP address of the Linux server on which the SAP HANA database is running and its instance number as for the system database. Simply copy the IP address again from the **Information** section of your SAP BW/4HANA instance. The instance number of the SAP HANA tenant database is the same as for the system database: 02.

Figure 4.10 Add Connection Parameters for the SAP HANA Tenant Database

3. Choose **Multiple containers** and **Tenant database** as the mode again.

Tenant Databases

The schema of your SAP BW/4HANA 2.0 system resides in the first tenant database. We chose the description **TenantDB 1**, as SAP HANA assigns the first available port to a new tenant.

4. Now confirm your entries by clicking **Next** to enter the ID of the system user SYSTEM of your SAP HANA database in the last screen (see Figure 4.11).

Figure 4.11 Creating Connection Properties to the SAP HANA Tenant Database

5. Also, check the box **Store user name and password in secure storage** like you did earlier for the system database. Leave the SSL checkbox empty for now. The **Enable SAP start service connection** checkbox is optional for tenants, as the system database connection takes care of starting SAP HANA services.

Complete the installation of the connection by clicking **Finish**. Next, repeat the steps above for the owner of the SAP BW/4HANA schema, SAPHANADB. Connect it in the same way as the system user for the tenant database A4H. At this time, only SAPHANADB has access to the data dictionary tables.

Once the SAP HANA connections have been established, create a connection to the SAP BW/4HANA server:

1. Remain in the **SAP HANA Administration Console** perspective and select **New • Project**.
2. In the following dialog box, choose the wizard for creating an SAP BW/4HANA project (see Figure 4.12).

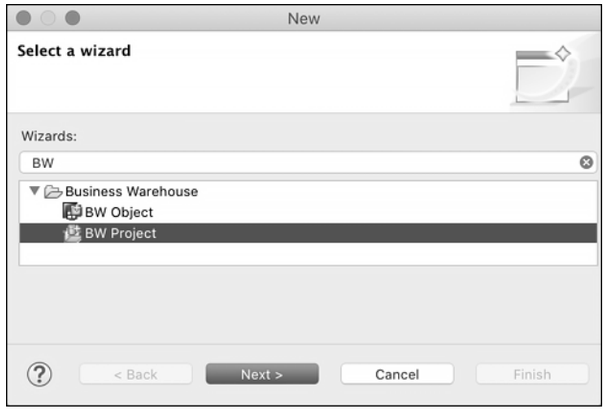


Figure 4.12 Creating a New SAP BW/4HANA Project

3. On the following screen, choose **New System Connection** (or **System Connection for the new BW project**) and enter the system ID and the external IP address of the SAP BW/4HANA server (identical to the data of the SAP HANA server; see Figure 4.13). Then confirm by clicking **Next**.

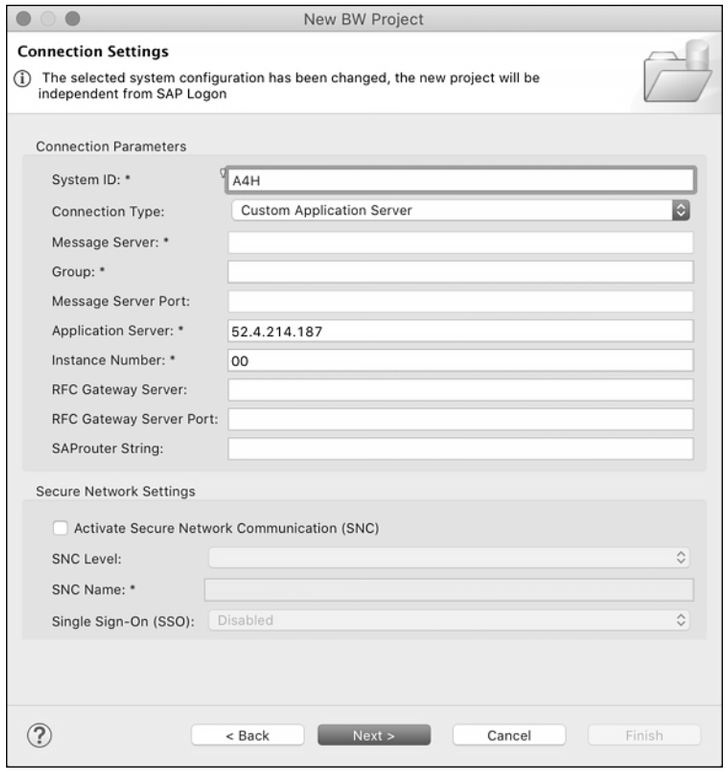


Figure 4.13 Setting Up a New SAP BW/4HANA System Connection in SAP HANA Studio

4. Enter the development user "DEVELOPER" and your master password. Click **Next** again to provide a dedicated project name (see Figure 4.14).

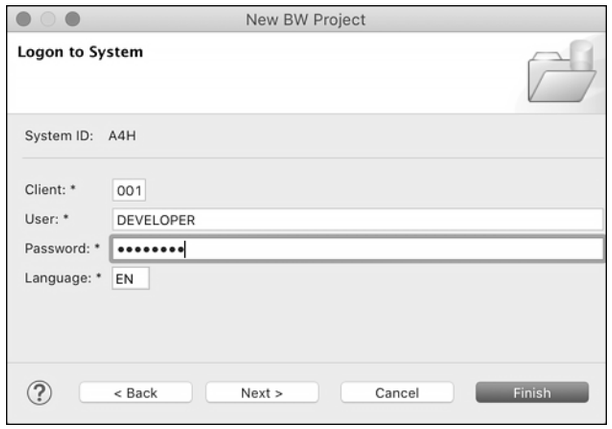


Figure 4.14 Logon as Developer User to the SAP BW/4HANA System

Tip: Assign Your SAP BW/4HANA Project Name in the Wizard

You should assign your own SAP BW/4HANA project name by clicking **Next** first and then **finish** later. This has the advantage that your projects do not receive a generated name according to the pattern *<System ID><Client><User><Logon><Language>*. Assign your SAP BW/4HANA project names so that you can distinguish them from your future ABAP projects, in particular. We'll show you how to set up a connection with the ABAP application server, that is, the SAP BW/4HANA server, in Section 4.1.6.

With these connections, you can now create your first SAP BW/4HANA objects, which we'll show you in Chapter 6.

4.1.2 Installing Eclipse and the SAP BW Modeling Tools

In principle, you're free to choose whichever version of the Eclipse IDE you want, as you can install Eclipse plugins at any time. For developments on SAP Cloud Platform, however, the *Eclipse IDE for Java Developers* is recommended (*http://s-prs.co/v454429*), because Java is an essential development language of SAP Cloud Platform. The download link *https://tools.hana.ondemand.com/latest* points to the latest download page for Eclipse supporting the current SAP BW modeling tools and ABAP Development Tools. This link guides you to the Eclipse download center (Figure 4.15). Select the operating system matching your environment and start the download onto your laptop or cloud server.

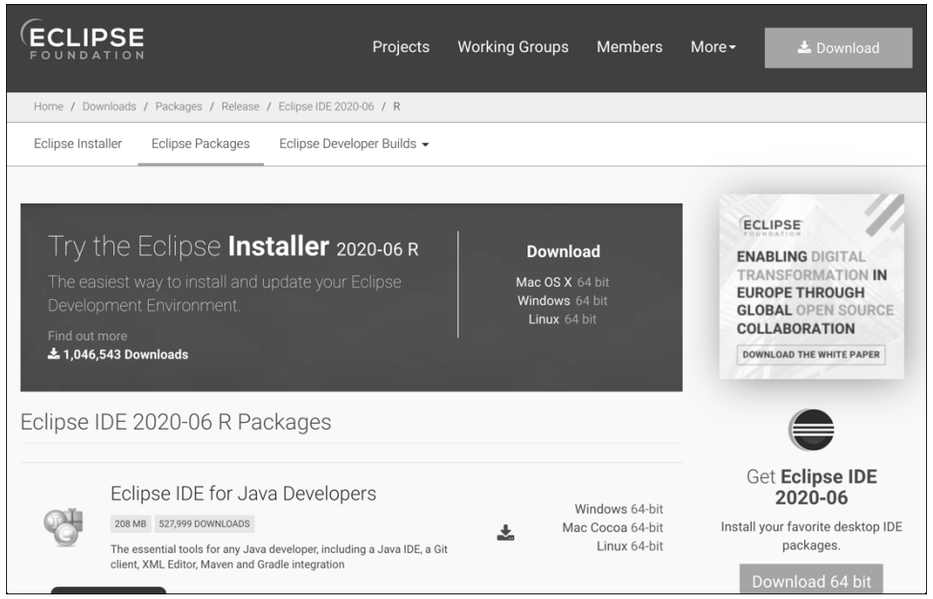


Figure 4.15 Download of the Latest Eclipse IDE

Again, you may choose any type of Eclipse package within the installer window; however, since most development tools are based on Java, most developers choose the Eclipse IDE for Java developers. Once the download has finished, open it and follow the installation guide.

You'll be asked to select a directory as a workspace, as shown in Figure 4.16. If you're working with multiple IDEs at the same time, ensure that their workspaces are separated. Keep in mind that the last section of the path to your workspace later appears like the window title in Eclipse. Once you're done, click **Launch** to start Eclipse.

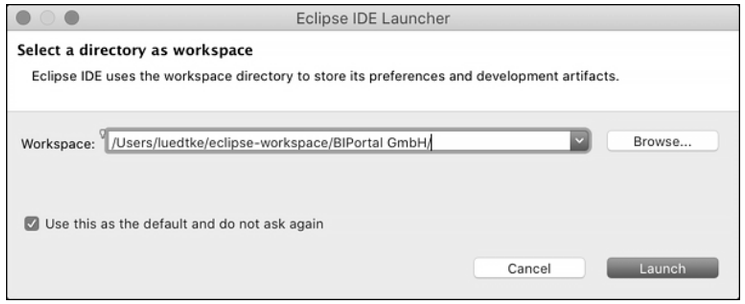


Figure 4.16 SAP HANA Studio Workspace Setup

If you've used the workspace with an earlier version of Eclipse, please keep in mind that updating it will result in an incompatible version with previously installed tools (Figure 4.17). Click **continue** to overwrite.

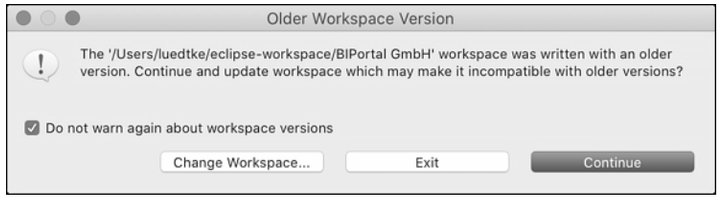


Figure 4.17 Incompatible Older Workspace Versions in SAP HANA Studio

On a Mac you'll probably be asked for access to the directory where you want to place the workspace. Confirm by clicking **OK**. Next, install the SAP BW/4HANA modeling tools along with the ABAP workbench and the SAP HANA modeling and development tools.

1. Open Eclipse and select a directory name for your workspace.
2. If you're behind a firewall or accessing the Internet via a proxy server, enter this proxy for the HTTP and HTTPS protocols on the **Window • Preferences • General • Network Connections** page.
3. Then navigate to **Help • Install New Software** to install the SAP development tools for Eclipse.

   All of this can be done via the menu path **Help • Install New Software** (Figure 4.18).

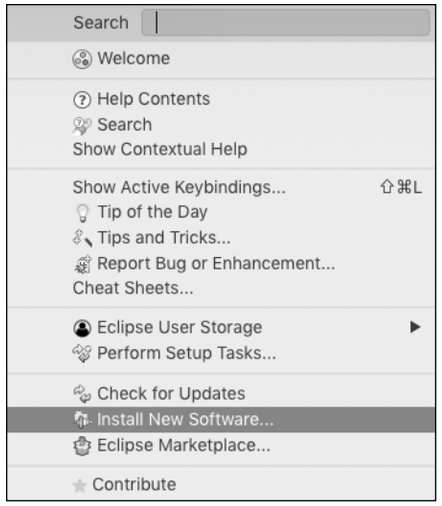


Figure 4.18 SAP HANA Studio Launch of the Software Installer

Select **Add new source** and enter the download path "*https://tools.hana.ondemand.com/latest*" as described previously and confirm by clicking **Add** (Figure 4.19).

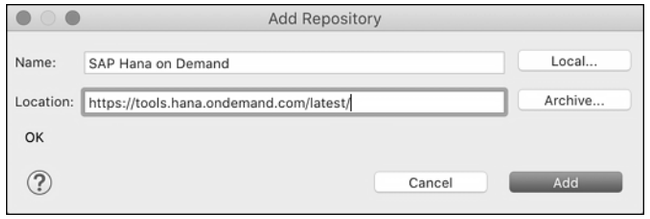


Figure 4.19 SAP HANA Studio Download Site Setup for SAP Development Tools

4. Select all available SAP tools (and therefore also the SAP BW modeling tools) to avoid subsequent installation. By default, the Eclipse installer will only mark items that haven't been installed yet (see Figure 4.20). Confirm by clicking **Next and** installing the entire collection of tools (although the UI Development Toolkit for HTML5 is not required in this book).

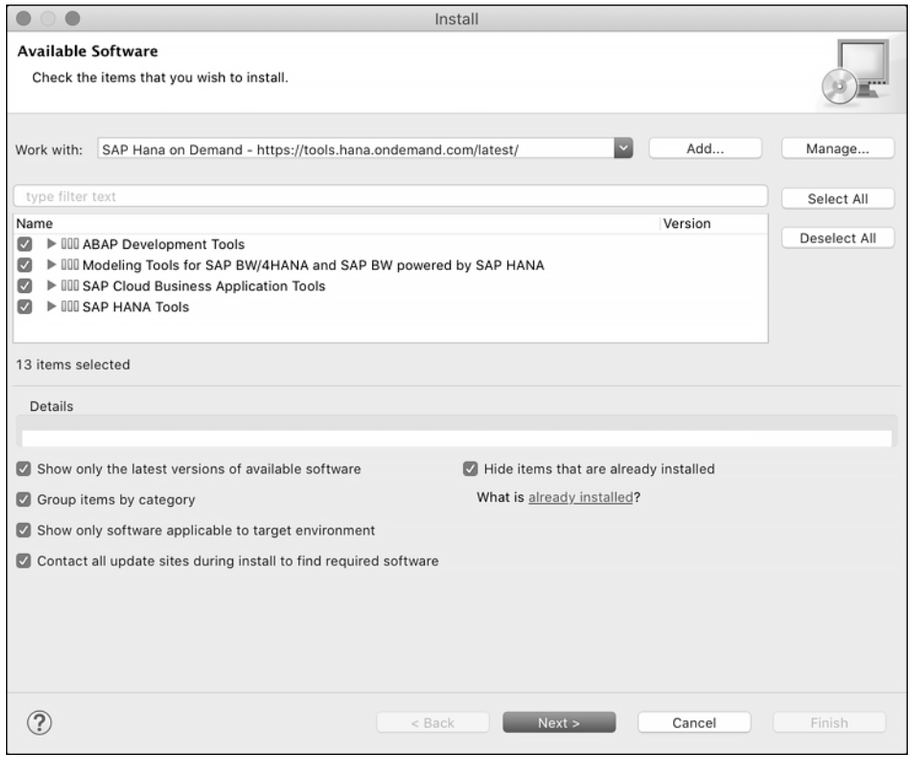


Figure 4.20 Installing the SAP Development Tools for Eclipse

5. The next screen shows you the full contents of the previously selected packages. Confirm by clicking **Next.**
6. Then confirm the installation by clicking **Next**. Accept the license terms by clicking **Finish**.

7. Once the installation of the SAP development tools is complete, you're prompted to restart the Eclipse IDE. Confirm the restart.

Tip: Subscribe to Information about Updates

Navigate to **Window • Preferences • Install/Update • Automatic Updates • Automatically find new updates and notify me**, and click **Apply** to be informed about updates to the SAP development tools.

You can now go ahead and explore your new IDE. Enter "HANA" in the search window at the top and select the SAP HANA administration perspective.

4.1.3 Connecting to SAP HANA

On the far left in Eclipse, under **Systems**, you have the option of establishing a connection with an SAP HANA system. However, if you want your shortcuts to be added immediately to the correct project folder, right-click the project folder to open the shortcut menu. Use this to establish the connection to your SAP HANA system for SAP BW/4HANA. In the following window, enter the connection parameters of your SAP BW/4HANA installation in the AWS. These are the external IP address of the Linux server on which the SAP HANA database is running and its instance number. You can find the IP address directly in the **Information** section of your instance in SAP Cloud Appliance Library (see Figure 4.21). You can find the instance number in the user guide.

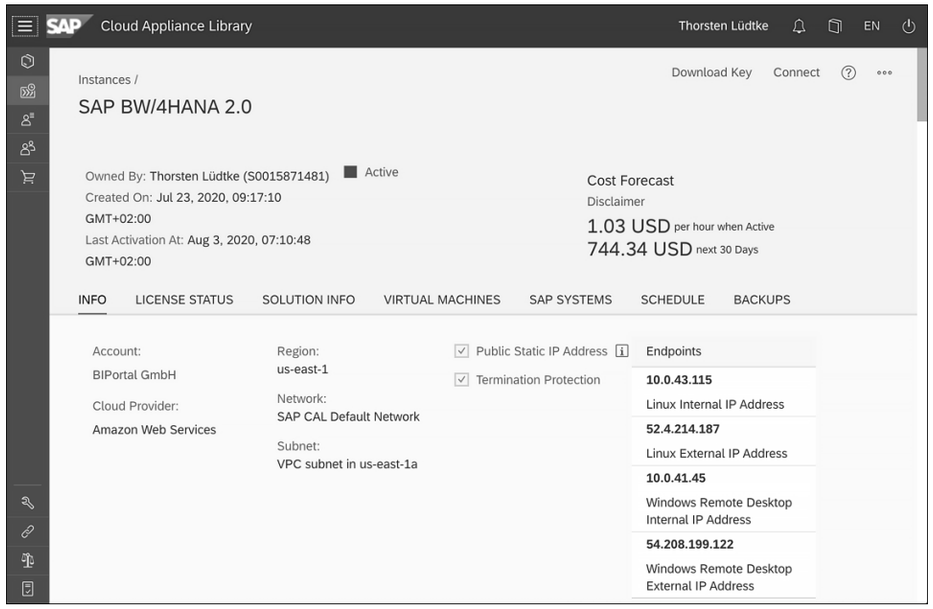


Figure 4.21 Retrieving the External IP Address of the SAP HANA Linux Server (Host) of an SAP BW/4HANA Instance in the AWS Cloud

Also, check the box **Store user name and password in secure storage to** avoid having to enter the system ID again and again later. Leave the other checkboxes blank.

Setting Up SSL Encryption

Encrypted SSL communication is not activated by default on the SAP Cloud Appliance Library instance of the SAP BW/4HANA system. Besides, the HTTPS port is not open. The assumption here is that you do not need encrypted communication for test purposes or that encryption has already been performed by your virtual private network (VPN). For a production system, we recommend SSL encryption. However, the SAP HANA system must also be configured for SSL communication:

1. To set this up, choose **Communication and Monitoring • Open Administration** in the context menu of the HDB system.
2. **On the Configuration** tab, enter "ssl" in the **Filter** field.
3. Under the **communication** node you can now change the value **off** to the value **on** via the context menu. This adapts the *global.ini* file for SSL communication (see Figure 4.22).

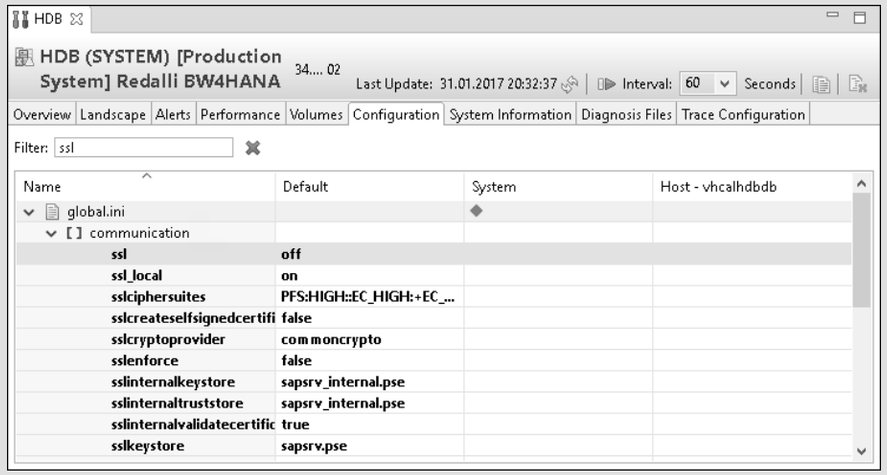


Figure 4.22 Setting Up Encrypted SSL Communication with the SAP HANA Database

For detailed information on SSL configuration, see the *SAP HANA Security Guide* at *http://s-prs.co/v454430*.

For testing purposes, we don't use the quite extensive SSL configuration here. Therefore, leave unencrypted communication at this point and confirm the setup of the link with your SAP BW/4HANA system by clicking **Finish**.

4.1.4 Creating an SAP HANA Archive Link

Now, it would be too cumbersome if every developer or consultant had to maintain the connection parameters individually. Therefore, there is an archiving link in SAP HANA

Studio that allows you to read the connection parameters from a central file so all project members can access it. You can therefore export the link you just created as an XML file to a shared server directory:

1. Select **File • Export** from the main menu. In the window that opens, enter "Landscape" under **Select an export wizard** (see Figure 4.23).

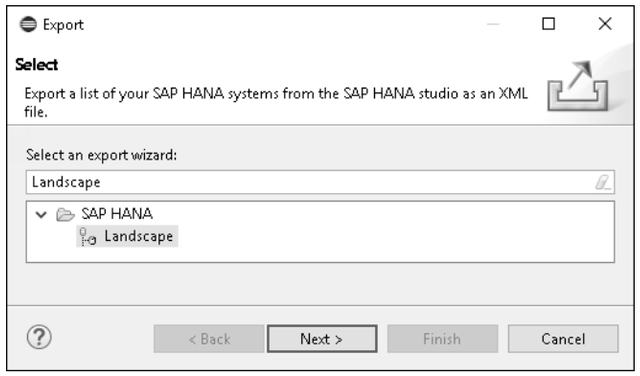


Figure 4.23 Exporting the System Landscape in SAP HANA Studio

2. Click **Next to** select the central directory.
3. Select all systems together with your project folders that you want to export and confirm by clicking **Finish**.
4. You can then insert the generated *landscape.xml* file for your system landscape on the **Systems** tab in a new workspace or client of SAP HANA Studio by clicking on the **Add** icon and then on **Add System Archive Link** (see Figure 4.24).

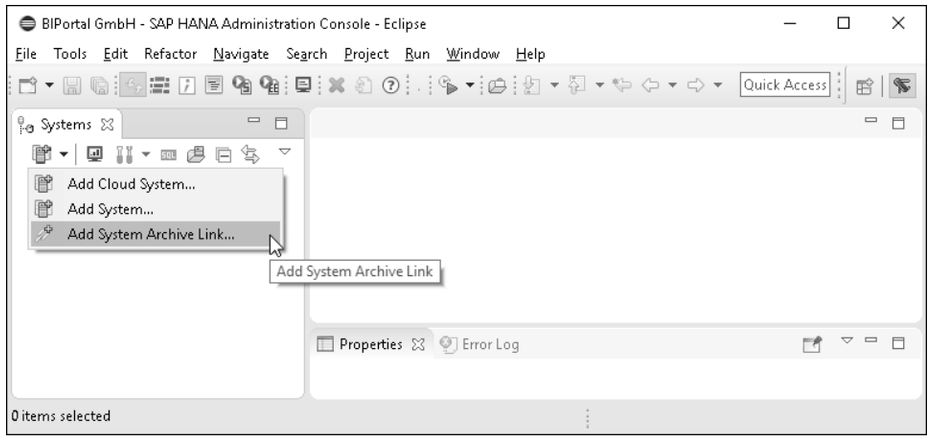


Figure 4.24 Adding an Archive Link in SAP HANA Studio

Of course, the passwords are not exported but must be entered by the user if they are not yet stored in the *secure* area *(secure store)* of the user client.

4.1.5 Configuring the SAP HANA Usage Type

The usage type of the SAP HANA system underlying SAP BW/4HANA is set by default to **production**. Obviously, this does not make any sense in a trial environment. The constant warnings whenever you change a feature mar your user experience. We there encourage you to change the value for the key usage for the parameter **system_information** to "development". You can do this by right-clicking on the usage parameter and selecting **Change**, as shown in Figure 4.25.

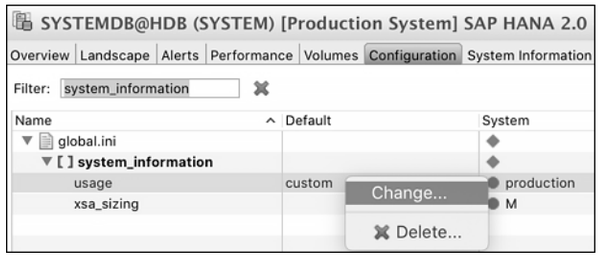


Figure 4.25 Configuring the SAP HANA usage type

Once you've clicked **Save**, the bright yellow background color at the top disappears.

4.1.6 Installing the ABAP Development Tools for Eclipse

In this section, we'll show you how to install the *ABAP Development Tools*, which are a component of SAP HANA Studio. You'll use the ABAP Development Tools in addition to the SAP BW modeling tools. The ABAP Development Tools are a prerequisite for programming transformation rules that must be executed in the SAP HANA database. The SAP BW/4HANA transformations can be embedded into *ABAP managed database procedures* (AMDP).

The installation of the ABAP Development Tools is similar to that of the SAP BW modeling tools:

1. In the opened Eclipse IDE, select **Help • Install New Software**.
2. In the **Work with** window, enter "*https://tools.hana.ondemand.com/latest*" as the source of the latest release of the ABAP Development Tools. Press Enter to display the available features. Select **ABAP Development Tools** and choose **Next**. On the next wizard page, you get an overview of the features to be installed (see Figure 4.26).

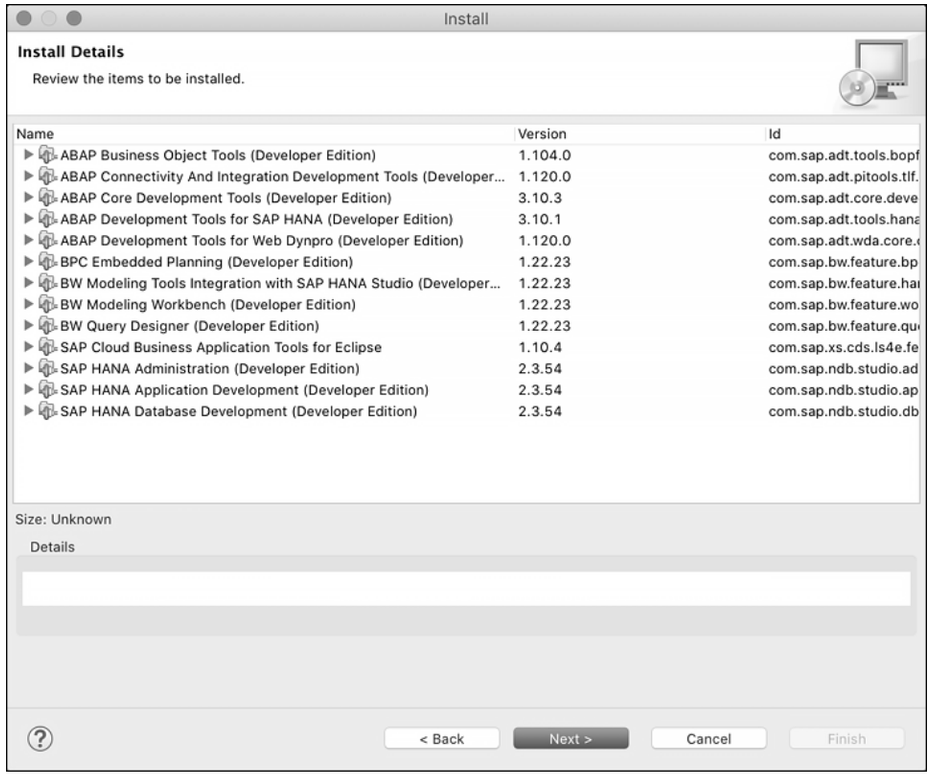


Figure 4.26 Installing the ABAP Development Tools for Eclipse

3. Next, review and accept the SAP license agreement and confirm by clicking **Finish**. You may be prompted to update a previous installation. Confirm the prompt by clicking **Next** and restart Eclipse.

You might encounter security warnings about unsigned contents. Ignore this warning since you know what you're doing and click **Install** anyway. You can find more information on this at *http://s-prs.co/v454431*. Finally, Eclipse asks you to restart the IDE. Confirm by clicking **Restart Now**. Upon restart, you're be prompted with several useful links and documentation installed locally on your laptop, as you can see in Figure 4.27.

Explore its contents to get a feeling for your fresh IDE. The **Release Notes** (see Figure 4.28) update you on the latest features.

Figure 4.27 Exploration of SAP BW Modeling Tools in SAP HANA Studio

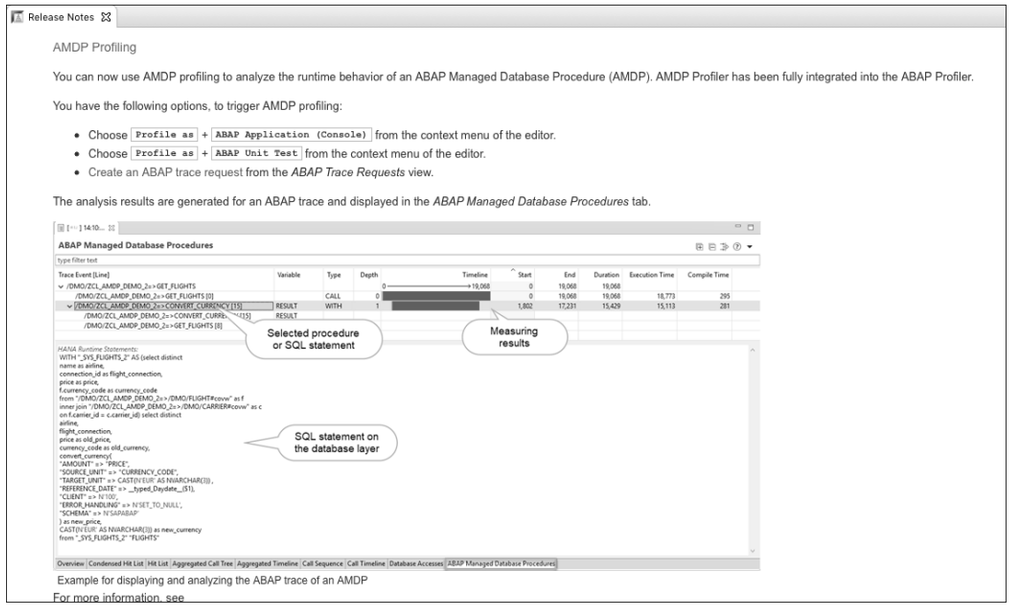


Figure 4.28 SAP HANA Studio News Page

SAP HANA Studio is now ready to connect to SAP BW/4HANA 2.0. To get started, select **ABAP Project** from the top menu bar (see Figure 4.29).

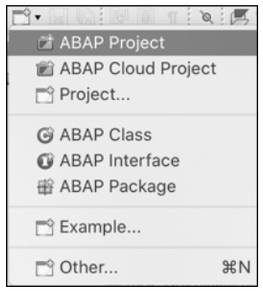


Figure 4.29 Starting the ABAP Project Wizard from the New Objects Menu

Next, choose the same system connection as for the SAP BW modeling tools (see Figure 4.30).

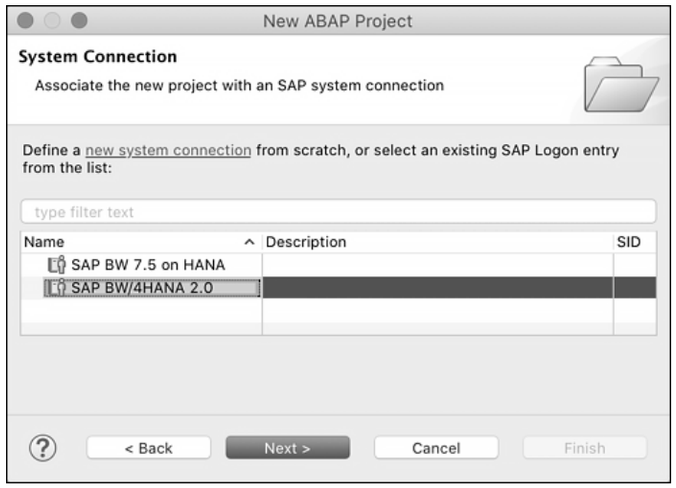


Figure 4.30 Choosing a System Connection within the ABAP Project Wizard

The SAP BW/4HANA server connection settings are the same as for the SAP BW modeling tools (see Figure 4.31).

Click **Next** to enter your SAP BW/4HANA development user and password and move on by clicking **Next** again. You are now asked to specify a **Project Name**. Remember, you cannot use the same name as for your SAP BW/4HANA Redalli project. Choose a name without blanks. Next, select your Redalli development package for your project (see Figure 4.32).

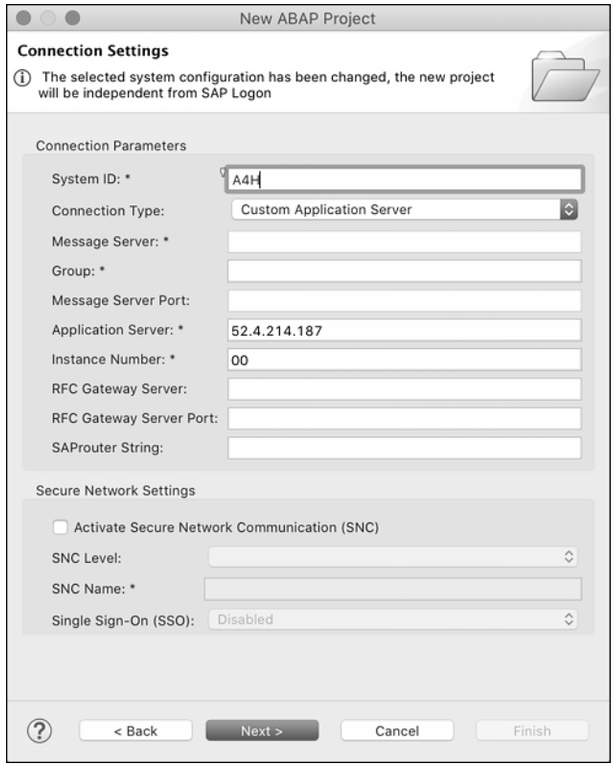


Figure 4.31 Specifying the Connection Settings for an ABAP Project

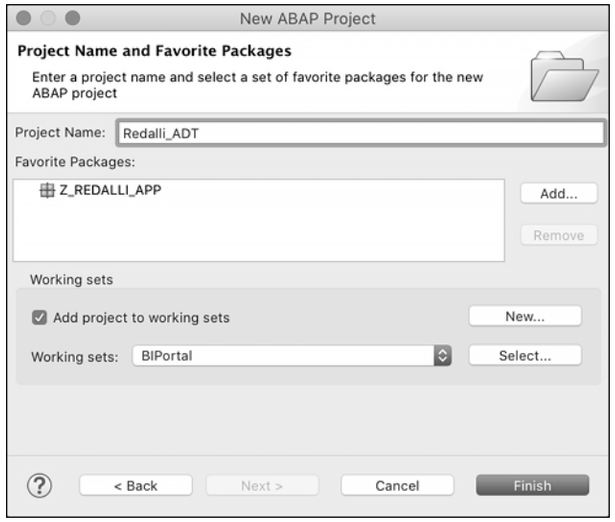


Figure 4.32 Selecting a Development Package

Add your project to your existing SAP BW/4HANA working set, as both contain objects for the same application development resources. Confirm by clicking **Finish**. Your new connection shows up below the **BWMT** (SAP BW modeling tools) connection (Redalli_ADT in our example, as shown in Figure 4.33).

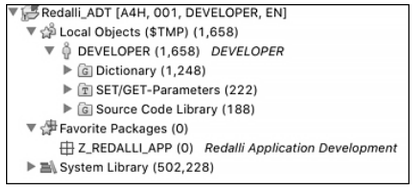


Figure 4.33 ABAP Development Tools Connection to the SAP BW/4HANA Server

You can now overwrite the default connection name that appears in your IDE. Since you'll have different projects for ABAP and SAP BW/4HANA, it makes sense to choose a corresponding prefix. Once that's done, repeat these steps for your SAP BW/4HANA connection.

4.1.7 Creating a Repository Workspace and Package

Now create a *repository workspace* and a *package* for your SAP HANA objects. The repository workspace on your SAP HANA system allows different Eclipse installations to access the SAP HANA repository so that different developers can share the same development repository. Figure 4.34 shows how to create a repository workspace.

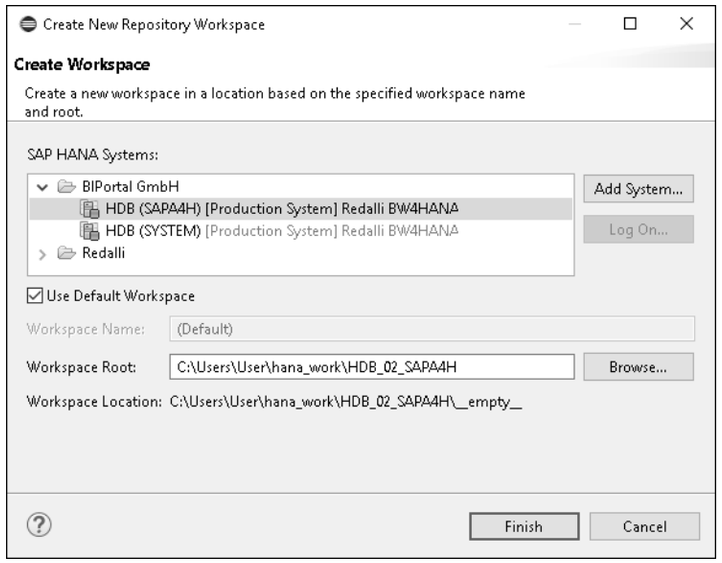


Figure 4.34 Creating a Repository Workspace

You can subdivide your developments into numerous *subprojects*, each of which relates to the *main project*. You can use the assignment shown in Figure 4.35 to create a reference to the main project. The best way to split a project is to use different packages that refer to the same repository. This allows developers to build their objects on those of other developers.

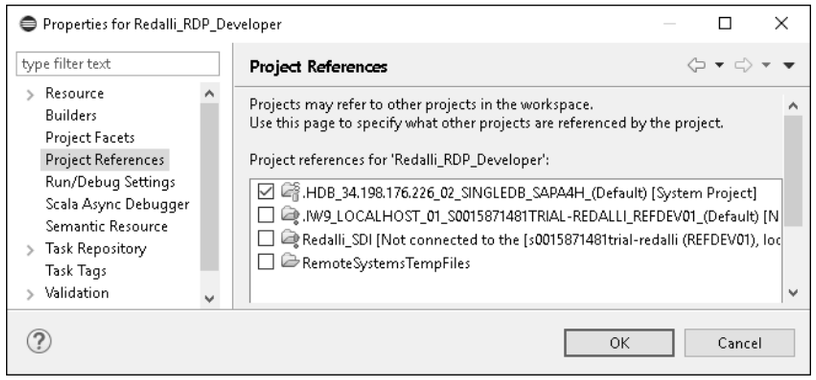


Figure 4.35 Assigning the Project "Redalli_RDP_Developer" to the Default Workspace

4.1.8 Creating a Developer Package

All objects created are assigned to a development package. This package is linked to modification rights for the objects it contains so that you can control which developer can work on these objects.

To create a development package for Redalli, switch to the **SAP HANA Modeler** perspective and open your repository's **Content** folder as user `ADMINISTRATOR`. In the context menu of the **Content** folder, choose **New • Package**. In the dialog that opens, enter a descriptive package name for the Redalli project, e.g., "Redalli_RDP" (RDP stands here for "Redalli development package"), as shown in Figure 4.36. Confirm by clicking **Finish**.

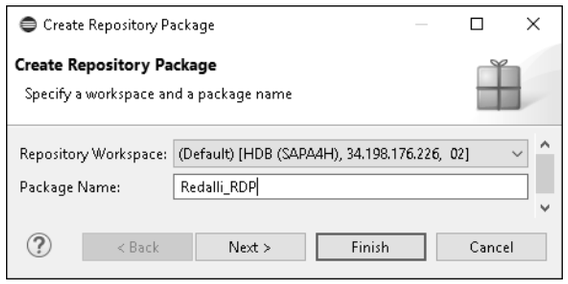


Figure 4.36 Creating a Package for the SAP HANA Database Objects

You have now created a development package and the corresponding repository workspace in which the developers of your project can work. This section completes the SAP HANA Studio setup.

4.1.9 Error Handling

Before we proceed, let's discuss some solutions for possible errors that may arise while you're installing and setting up SAP HANA Studio.

Exercise 1: Error Handling for the SAP Start Service

In this section, we'll briefly show you an example where direct access to the operating system level is necessary for problem analysis. There is another service, the *SAP start service*, which can be controlled via a separate connection. The operating system user `<SID>adm` (here, `hdbadm`) accesses the operating system using a secure HTTPS connection to start or stop the SAP HANA sysnftem. You can maintain this SAP start service from the context menu of the SAP HANA system:

1. Under **Properties,** choose the entry **SAP Start Service Logon**.
2. Then enter your *master password* for the administration user (see Figure 4.37).

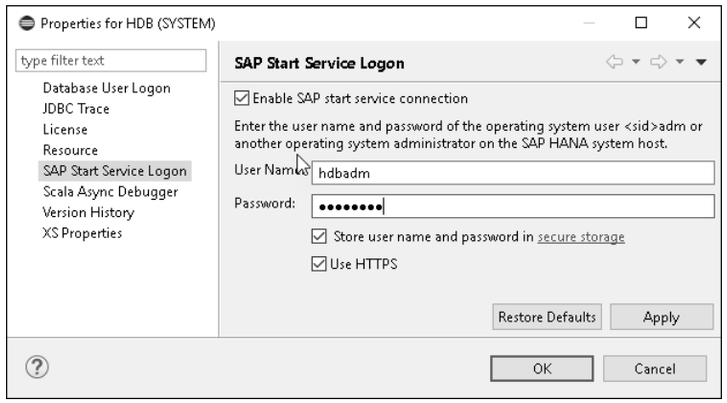


Figure 4.37 Maintaining the SAP Start Service Logon

However, since the HTTPS port is not currently open in SAP Cloud Appliance Library for the SAP BW/4HANA system, a timeout error occurs during the connection attempt. The relevant forums therefore often discuss whether the `sapstartsrv` service was not started because the options for starting and stopping the SAP HANA instance are not available in the context menu of the SAP HANA system, as shown in Figure 4.38.

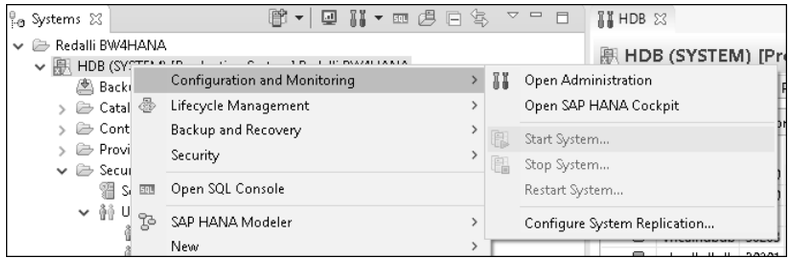


Figure 4.38 Start and Stop Not Available in HDB

The error message is displayed when you hover your mouse pointer over the system entry (see Figure 4.39).

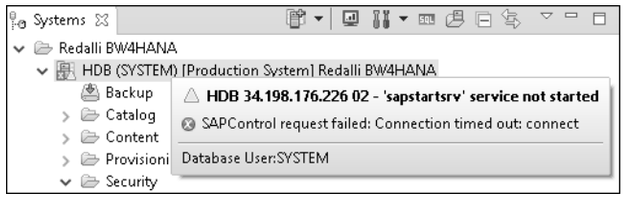


Figure 4.39 Error Message Due to Missing Access to the SAP Start Service

The **Error Log** shows you more detailed information about the connection problem (see Figure 4.40).

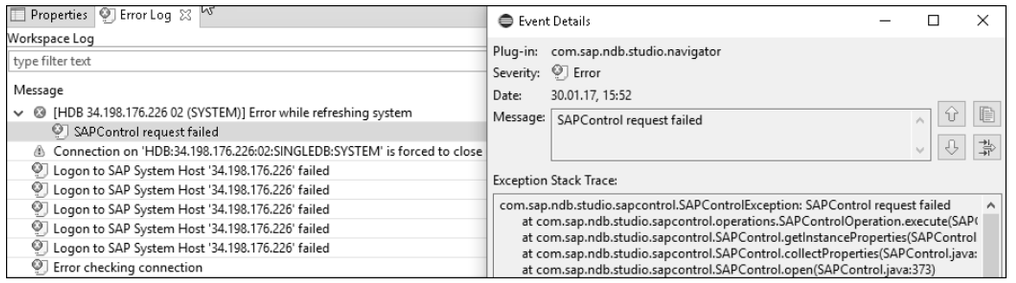


Figure 4.40 Error Log for Connection Problem in SAP Start Service

Since the **Use HTTPS** checkbox in Figure 4.37 is an option, it's always possible to establish a connection to the SAP start service without encryption. In the console, you can use the command `openssl s_client -connect < external IP address of the SAP HANA server>:50014` to check whether you receive the same connection error (of course, adjust the host address to suit your system). In most cases, however, the SAP start server is running, so the error message is a false message. You can confirm this by using the command `lsof -i -i -n -P | grep 5<HDB instance no>13` (here the instance number of the SAP HANA database is 02) to check whether *port 5<HDB instance no>13* is open.

The problem is therefore more likely to be with the network connection. Some services use the Domain Name System (DNS) name, which cannot always be resolved, depending on the network configuration. Thus the target machine remains inaccessible.

According to SAP Note 1760921, an entry in the *hosts file* under Windows (*hosts* without extension) can help here:

1. Enter the external IP address of your SAP HANA server, followed by the logical hostname:

   ```
   34.198.176.226          vhcalhdbdb
   ```

2. Save the *hosts* file afterward. If this is rejected, save the file to your desktop and then drag it to the old hosts' directory. In this case, confirm that you want to overwrite the old file.

3. You can also enter the fully qualified host name of the SAP HANA database. You get this name with the command `hostname -f`. In the case of Redalli, the hostname is `vhcala4hcs.dummy.nodomain`.

Querying DNS and Host Names

The Eclipse installation on the frontend server in the AWS can serve as a reference for the names to be entered. There, all used DNS names are assigned to the internal IP address in the *hosts file*, as you can see in Listing 4.1.

```
# localhost name resolution is handled within DNS itself.
#      127.0.0.1       localhost
#      ::1             localhost
# added by a service of the CAL Appliance Agent

10.0.41.45 frontend.dummy.nodomain frontend
10.0.43.115 vhcalhdbdb.dummy.nodomain vhcalhdbdb vhcala4hci.dummy.nodomain
vhcala4hci vhcala4hcs.dummy.nodomain vhcala4hcs
```

Listing 4.1 Assignment of the DNS Names to the Internal IP Addresses in the Hosts File of the SAP Cloud Appliance Library Frontend Server

If you're using your own PC, you have to assign the external IP address accordingly and then save the *hosts file*.

According to SAP Note 1639568, you should also change the value from **Native** to **Direct** in SAP HANA Studio under **Windows** (on a Mac under **Eclipse**) • **Preferences** (or **Settings**) • **General** • **Network** • **Active Provider** (see Figure 4.41).

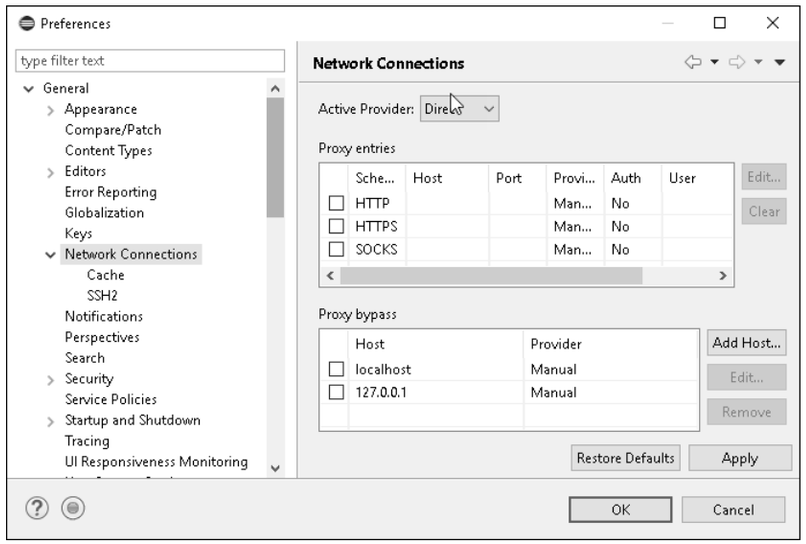


Figure 4.41 Switching the Eclipse Network Connection to an Active Provider

Basically, it's hardly possible to make a recommendation valid for all environments, since the network connection depends on routers, firewalls, etc. However, the SAP Notes mentioned can help you diagnose possible network problems.

Exercise 2: Error Handling when Accessing the Tenant DB

Once you've managed to connect to the system's database, you repeat the same steps with the first (and on SAP Cloud Appliance Library the only) tenant database managing SAP BW/4HANA's schema. You get all the connection details from the *Getting Started* guide attached to your instance on SAP Cloud Appliance Library. However, often, it is outdated or refers to a different solution on SAP Cloud Appliance Library. A quick and easy way to get all the connection details is by double-clicking the **Systems** connection shown in the left-hand pane and then navigating to the **Configuration** tab, as shown in Figure 4.42.

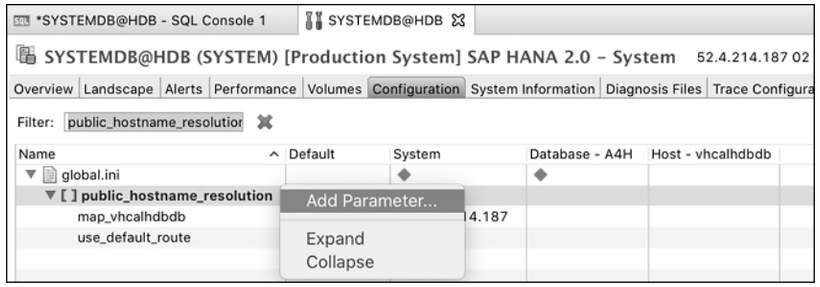


Figure 4.42 Public Host Name Resolution in SAP HANA Studio

Here, you can see the SAP HANA instance number (**02**) in the top-right and, right in front of it, the IP address of the host it runs on (**52.4.214.197**). Since SAP HANA 2.0 is by default container-based, it internally uses logical host names to pinpoint servers. From an external machine like your laptop, you therefore need to tell SAP HANA how to resolve host names into IP addresses. You see your host name in the far-right column (**vhcalhdb**). You can attach this mapping as a parameter to your system. Simply search for "public_hostname_resolution" in the **Filter** box. There should be exactly one entry for the parameter **use_default_route** with the default value **ip**. By right-clicking on the found parameter **public_hostname_resolution**, you can add your mapping. As a key, enter "*map_<host name>*", i.e., in our case "map_vhcalhdb". As the value, enter your SAP HANA host's IP address, i.e., in our case "52.4.214.197", and click **save**. You should now be able to connect to any tenant database. In the case of SAP BW/4HANA on SAP Cloud Appliance Library, the name of the tenant database is the same as the system ID of the SAP system: **A4H**.

Now that you're connected to SAP BW/4HANA's tenant database, you can add another connection from the context menu of your tenant connection. Right-click and choose **Add System with Different User**, as shown in Figure 4.43.

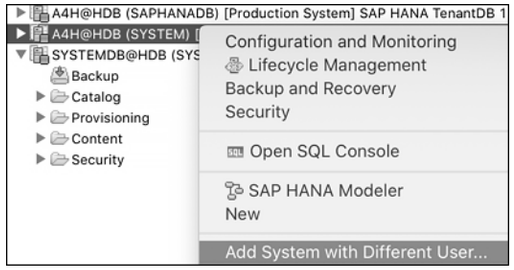


Figure 4.43 Add SAPHANADB User Connection to SAP HANA Tenant Database

In the following dialog, enter user "SAPHANADB" and your master password. It is the database user of your ABAP server with SAP BW/4HANA and hence has access to all tables and views within the SAP BW/4HANA system.

Exercise 3: A Quick Automatic Fix in the Error Log

In the main menu of SAP HANA Studio, you can display current error messages under **Tools • Show Error Log**. For example, when modeling *flow graphs* or other objects, you may receive an error message indicating that there is an incompatibility with the character set or another error has been detected. Here, you have the possibility of the *quick fix*. This means that in the context menu of the error message, you're offered the option to repair the inconsistency directly (see Figure 4.44).

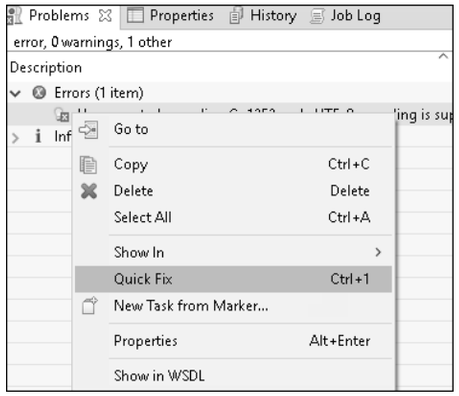


Figure 4.44 Selecting Quick Fix in the Context Menu of the Error Log

After you select the **Quick Fix** option, a window with a list of the repairs to be performed is displayed (see Figure 4.45).

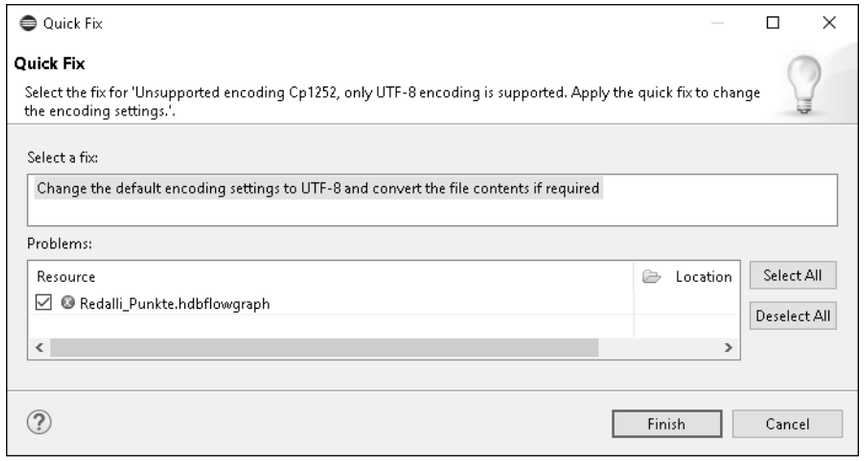


Figure 4.45 Example of a Quick Fix-Changing the UTF-8 Encoding

After confirmation with the **Finish** button, the flow graph model is correctly activated in this example. In the event of an error, the quick fix can be an option for quickly repairing objects in SAP HANA Studio.

4.2 SAP HANA Client Tool Installation

All third-party client software vendors use some of the external interfaces Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), SQL Database Connectivity (SQLDBC), or Multidimensional Expressions (MDX) via OLE DB for OLAP (ODBO). The drivers required for this are combined in the *SAP HANA client*. You can download the SAP HANA client from the SAP Software Download Center (see SAP Note 1825053). If you haven't yet installed the SAPCAR tool for unpacking .SAR files, then now is a good time. It's good practice to rename the SAPCAR file "SAPCAR.exe" so that future scripts can refer to it as the latest version. Next, run the command in Listing 4.2 in your console.

```
C:\Users\luedtke\Downloads>SAPCAR.exe -xvf IMDB_CLIENT20_005_101-80002083.SAR
SAPCAR: processing archive IMDB_CLIENT20_005_101-80002083.SAR (version 2.01)
x SAP_HANA_CLIENT …
SAPCAR: 114 file(s) extracted
```

Listing 4.2 Unpacking the SAP HANA Client Tools

You can now switch to the SAP HANA client directory and double-click the hdpsetup.exe installer. Check the default installation directory (on Windows) *C:\Program Files\SAP\hdbclient* for a large set of client tools for SAP HANA. Most third-party clients use the ODBC standard developed by Microsoft. On Macs you use the corresponding JDBC drivers. For JDBC you simply point your app to your JDBC directory. On Windows, however, you need to explicitly create an ODBC data source as shown in the next section.

4.2.1 Creating an ODBC Data Source for SAP HANA

After you have unpacked the SAP HANA client software and installed it on your PC, you can create a new data source or data source name under Windows:

1. To do this, select **Control Panel • ODBC Data Source Administrator (64-bit)** and switch to the **System DSN** tab.
2. Click **Add**. Then select the SAP HANA driver **HDBODBC** (see Figure 4.46). The HDBODBC driver is designed for 64-bit applications. For 32-bit applications, select the **HDBODBC32** driver. Confirm by clicking **Finish**.

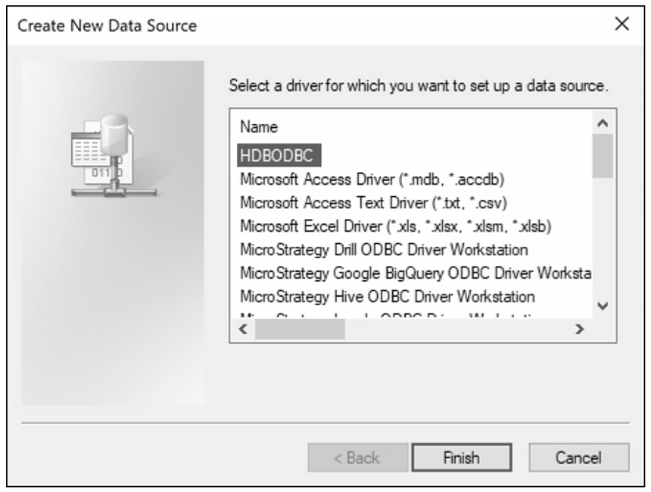


Figure 4.46 Creating a New SAP HANA Data Source in Windows 10

3. Enter the external IP address of your SAP BW/4HANA server on the next screen. Also, enter the port number `3<HDB instance>41` (here, port 30241 for the SAP HANA instance number 02 (see Figure 4.47).

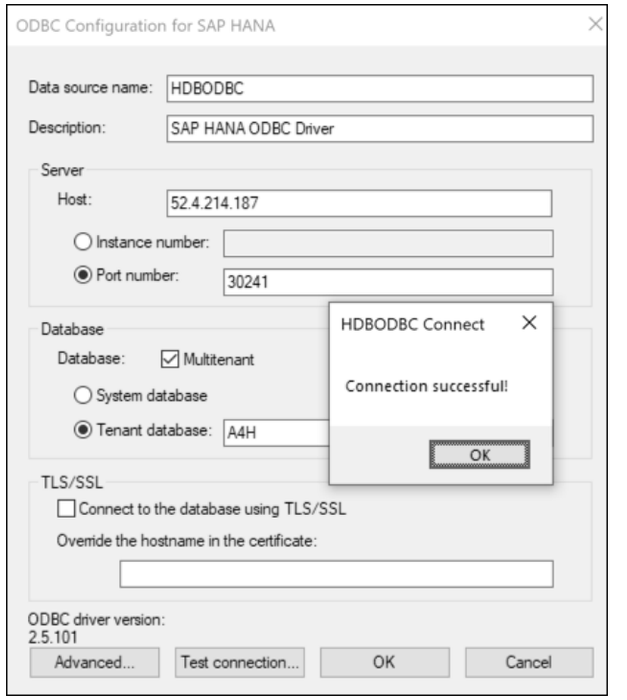


Figure 4.47 Specifying the SAP HANA Data Source for ODBC Client Tools

Tip: Check Port

The specific ports depend on the configuration of your SAP HANA system. Port 3<instance>15 is the default port for SQLDBC, MDX, ODBC, and JDBC communication with the *index server* of an SAP HANA database not running in multi-tenant mode. To be on the safe side, run the following SQL statement as a system user to check the port:

```
SELECT SERVICE_NAME, PORT, SQL_PORT, (PORT + 2) HTTP_PORT FROM SYS.M_SERVICES
WHERE ((SERVICE_NAME='indexserver' and COORDINATOR_TYPE= 'MASTER'))
```

In the **SQL_PORT** column, you see the port for external communication with the SAP HANA database. However, suppose you run your SAP HANA database in a multi-tenant environment as you do running SAP BW/4HANA 2.0. In that case, the SQL port number is assigned according to availability since not all tenants can use the same SQL port. Therefore, use the following SQL statement on the *system database to* check the SQL port of your specific tenant database:

```
SELECT DATABASE_NAME, SERVICE_NAME, PORT, SQL_PORT, (PORT + 2) HTTP_PORT FROM
SYS_DATABASES.M_SERVICES WHERE DATABASE_NAME='<DBNAME>' and ((SERVICE_NAME=
'indexserver' and COORDINATOR_TYPE= 'MASTER'))
```

Instead of the database name <DBNAME>, enter the name of your tenant database. Note that the user who executes this statement must exist as a user in the tenant database. To check the connection first, you can log on with the SYSTEM user, which exists in all SAP HANA systems.

Finally, click **Test connection** and enter a tenant user like "SAPHANADB" and your master password to check if the connection is working. You can now use BI frontend tools such as SAP Lumira, MicroStrategy, or Tableau on the SAP HANA database. For detailed information about the various BI options, see Chapter 8.

Note on Performance

Most third-party BI clients generate some sort of ANSI standard SQL. The disadvantage of this is that the server components are rarely optimized for the SAP HANA databases since they usually support numerous databases and therefore generate SQL that is applicable to the weakest database. This means that considerable performance advantages are lost and that specific functions of the SAP HANA database are rarely used for queries.

4.2.2 HDBSQL Command-Line Tool

Together with the SAP HANA client, an SQL client called HDBSQL is also installed. You can call this client from SAP HANA Studio:

1. To do this, select **Window • Show View • Other** and search for "Terminal" in the window that opens (see Figure 4.48). Confirm by clicking **OK**.

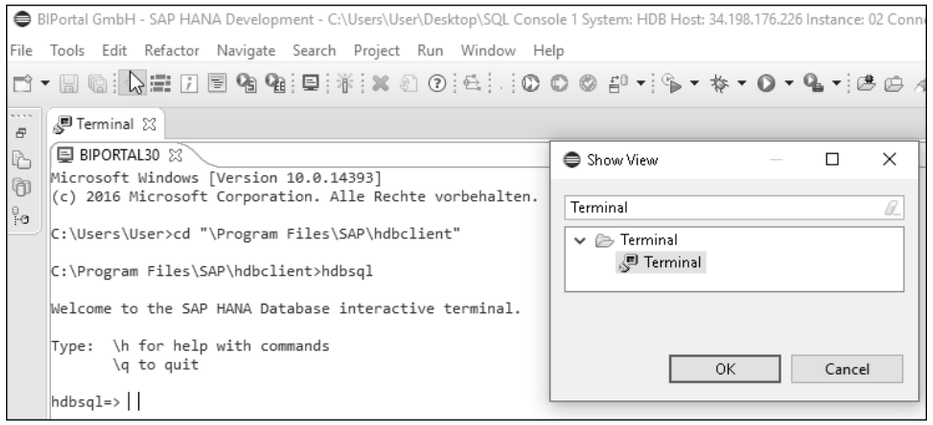


Figure 4.48 Opening the HDBSQL Client in SAP HANA Studio

2. **In the Terminal** window that opens, branch to the installation directory of your HDBSQL client (usually *\Program Files\SAP\hdbclient*) and start the HDBSQL client with the command `hdbsql`.

You can now execute SQL statements in the interactive client shell. The `\h` command gives you an overview of the possible entries. With `\q` you can leave the HDBSQL client. In the SAP documentation *SAP HANA Administration Guide* you can find a reference of all HDBSQL commands.

This section completes the installation of the SAP HANA client tools. Now you have physical access to your SAP HANA database.

Next, we'll create a connection to SAP Analytics Cloud.

4.3 Enabling SAP BW/4HANA for SAP Analytics Cloud

SAP Analytics Cloud is built natively on SAP Cloud Platform. It combines business intelligence, predictive, planning, and digital boardroom capabilities to analyze data from on-premise and cloud applications. SAP Analytics Cloud provides live connections (online) and data acquisition (batch) connectivity. In SAP Analytics Cloud, you can create models from your data sources, build stories based on those models, and perform online analysis with and without any data replication. This feature allows SAP Analytics Cloud to be used in scenarios where data cannot be moved into the cloud for security or privacy reasons or where your data already exists on a different cloud system. With replication, data is imported (copied) to SAP Analytics Cloud's SAP HANA database.

From a security perspective, SAP Analytics Cloud provides SAML 2 capabilities to enable single sign-on, simplifying authentication to SAP Analytics Cloud and connected data sources from your landscape. SAP Analytics Cloud provides the business logic and builds the queries required to see your browser's data. Your browser in turn sends those queries, through the reverse proxy, down or through a direct live connection to the on-premise database. The results of those queries are returned to the browser, where any charts are rendered. Throughout the process, the browser is interacting with the reverse proxy or through direct live connection (cross-origin resource sharing, or CORS), which in turn sends out the requests to SAP Analytics Cloud or the remote data source, depending on each request's path.

Older versions of SAP NetWeaver applications servers don't support native CORS connections. SAP Note 2596285 explains how to circumvent this issue by using a so-called ICM script. However, SAP BW/4HANA supports CORS. You can quickly check it using Transaction UCONCOCKPIT for unified connectivity settings, which is shown in Figure 4.49. Systems not supporting CORS usually don't support this transaction.

Unified Connectivity (UCON) is a tool to secure your system against intruders. For all kinds of communication scenarios, you can specify what is allowed and what is forbidden. The configuration of UCON is beyond the scope of this book. Therefore, we refer you to the documentation found at *http://s-prs.co/v454427*.

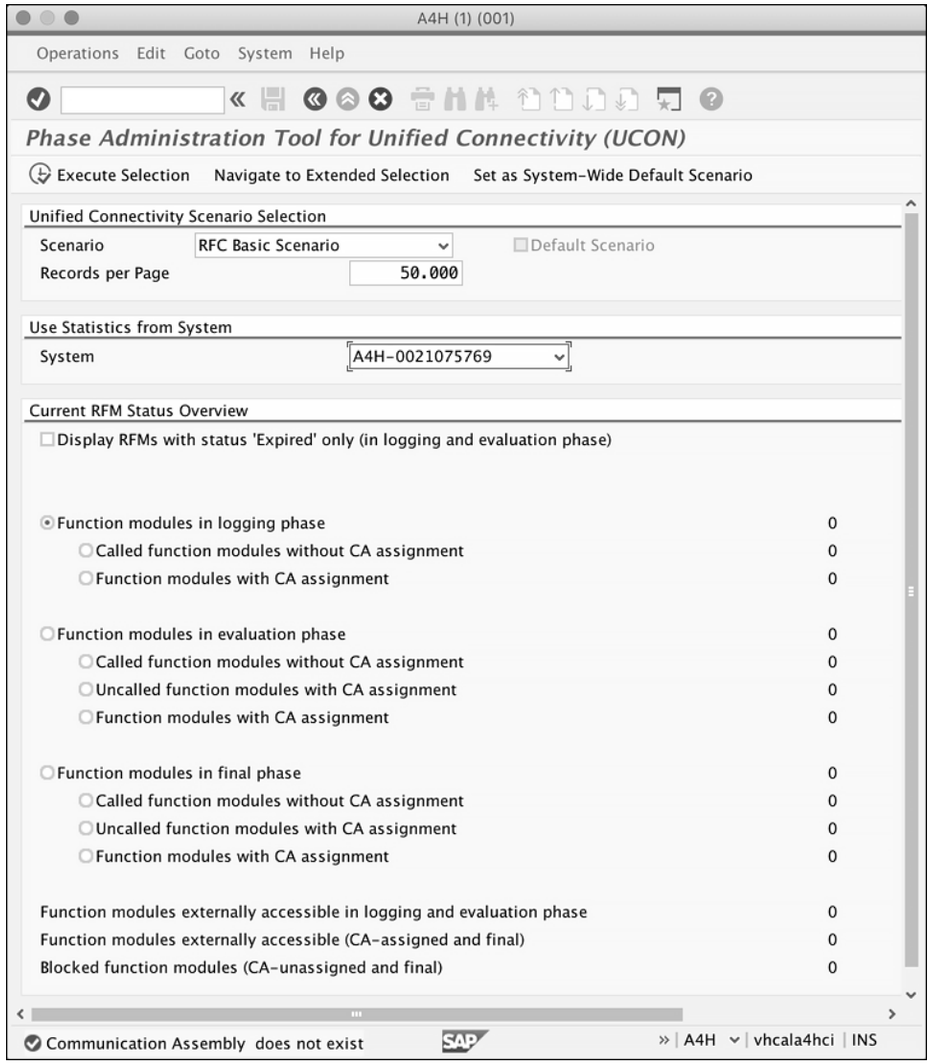


Figure 4.49 Unified Connectivity (UCON) Administration in SAP BW/4HANA

With SAML 2 and UCON in place, your system should be secure enough to proceed with CORS.

CORS is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. A

web page may freely embed cross-origin web pages, images, stylesheets, scripts, iframes, and videos. The same-origin policy in browsers is an essential concept in the web application security model. Under the policy, a web browser permits scripts in a first web page to access data in a second web page, but only if both web pages have the same origin. It's a critical security mechanism for isolating potentially malicious documents. In a live connection, a browser has to access both SAP Analytics Cloud for metadata and backend data sources (SAP HANA, SAP BW/4HANA, SAP S/4HANA, or universes).

If you use UCON to configure CORS, as you do with SAP BW/4HANA, you need to check that the parameter `icf/cors_enabled` is set to 1 in Transaction RZ11. As you can see in Figure 4.50, the **Current Value** is **1**.

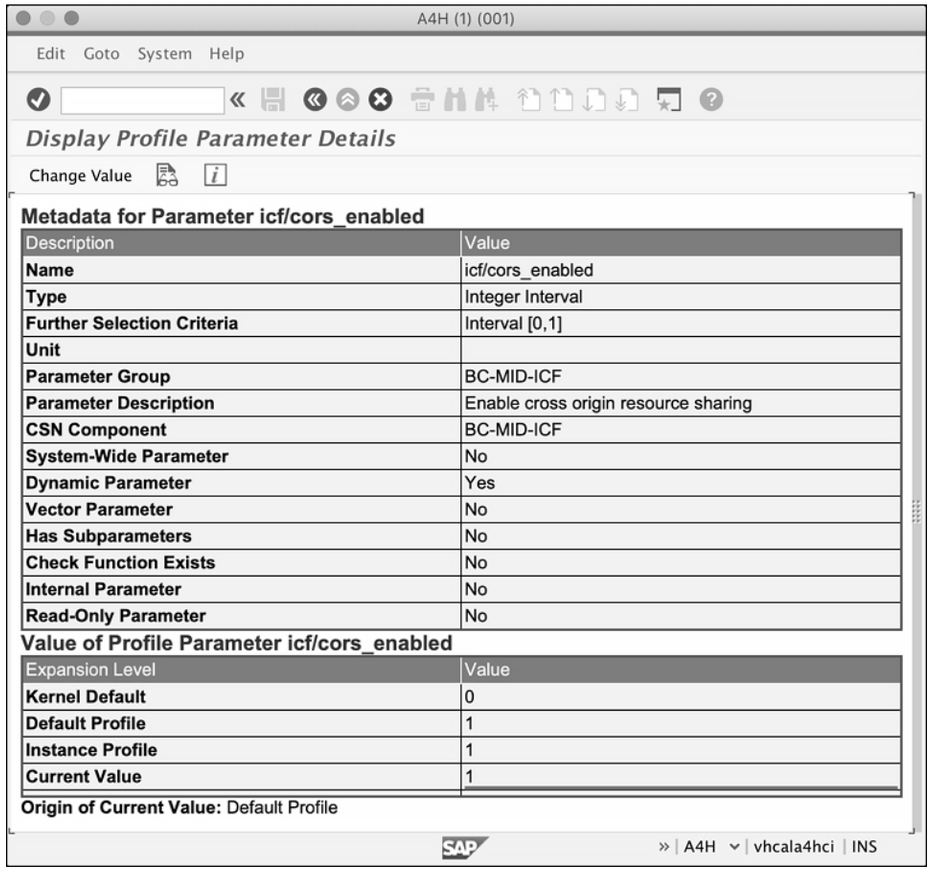


Figure 4.50 Checking CORS Enablement for SAP Analytics Cloud in SAP BW/4HANA

On SAP Cloud Appliance Library, `icf/cors_enabled` is disabled by default. To permanently enable CORS in your system parameters enter Transaction RZ10 and choose your default profile. Next, select **Extended maintenance** and click **Display**. You should see the screen in Figure 4.51.

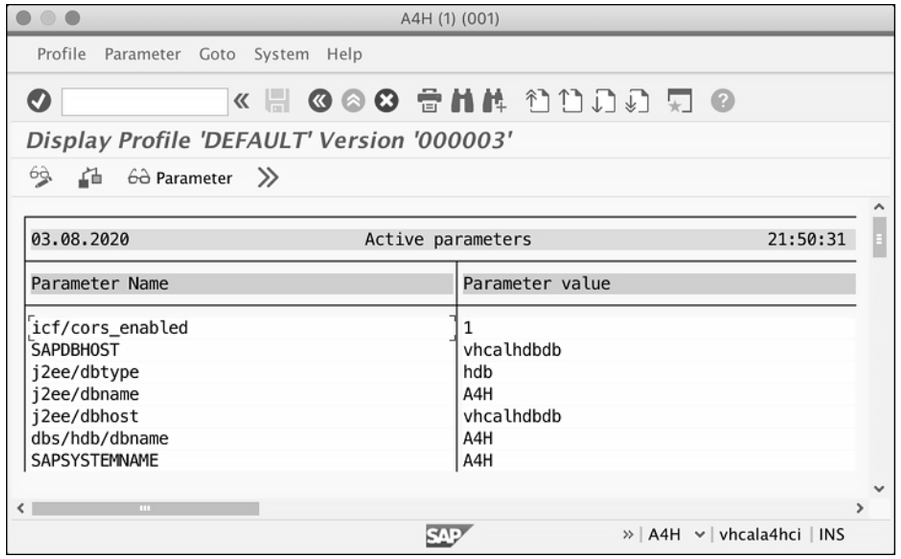


Figure 4.51 Enabling CORS for SAP Analytics Cloud in Transaction RZ10

If parameter `icf/cors_enabled` is set to 0, change it to 1 by clicking on the **Change** button (icon to the left showing a pencil). If the parameter does not exist yet, simply create it with key `icf/cors_enabled` and value "1". Click **Save** and return to activate your changed profile. If you get an error message telling you that the profiles selected don't correspond to the profiles being used, synchronize them by choosing **Import profiles • Of active servers** from the **Utilities** menu, as shown in Figure 4.52.

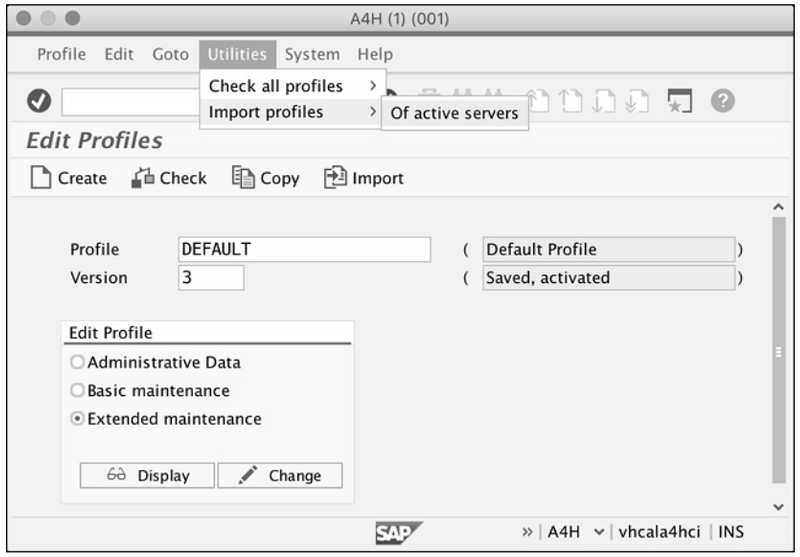


Figure 4.52 Importing Profiles from Active Servers in Transaction RZ10

You need to restart SAP BW/4HANA for the changes to take effect. The other parameters required should be set by running the task list SAP_BW4_AFTER_UPDATE (see Chapter 3 for more information about running task lists). This task list contains a step to activate the services for SAP Analytics Cloud access. However, you can also check it in Transaction SICF to ensure that the following services under */sap/bw/ina/* are active: `GetServerInfo`, `GetResponse`, `Loggoff`, and `BatchProcessing` (Figure 4.53).

Figure 4.53 Active Services for SAP Analytics Cloud access in Transaction SICF

If you still experience issues configuring your SAP Analytics Cloud access, refer to SAP Note 2659735.

Now you've made all the necessary configurations for live connections to SAP Analytics Cloud on the SAP BW/4HANA side. In Chapter 8 we'll show you what else needs to be done from an SAP Analytics Cloud perspective, so that a trusted connection can be established between SAP BW/4HANA and SAP Analytics Cloud.

4.4 Connecting to SAP Data Warehouse Cloud

You might ask yourself why SAP has chosen to build a cloud-native data warehousing solution—SAP Data Warehouse Cloud—from scratch rather than just place SAP BW/4HANA in the cloud like they have done with SAP S/4HANA cloud. The truth is that they actually evaluated this approach with Project Blueberry, which began in the summer of 2018. It turned out that SAP BW/4HANA was cloud-ready, just as we've presented in this book.

However, SAP carries around a legacy of almost 50 years with it. For SAP S/4HANA this is an asset. For analytics solutions it's a liability. Just think of the simple task of changing the background image. On Windows you might just right-click on the background and intuitively select a new image. Done. In SAP you might Google, "How to change background image in SAP start screen," and you'd get back lots of links, videos, tutorials, and other instructions telling you to execute a series of transactions and table maintenance operations before you can have your simple new image displayed. With more and more technology layers wrapped around this ancient core, integration and maintenance become major issues. Competitors noticed the opportunities of cloud-based data warehousing almost a decade ago and built solutions such as Amazon Redshift, Google BigQuery, and Snowflake. Snowflake went public in 2020 and is currently valued at US$60 billion, which is rather high compared to SAP's current market value of US$160 billion. Thus, SAP had to react and started the SAP Data Warehouse Cloud beta program in May 2019.

SAP Data Warehouse Cloud has SAP Analytics Cloud built in, in addition to its data warehouse and data integration capabilities. Built on SAP HANA Cloud Services within SAP Cloud Platform, it integrates data from all types of heterogeneous sources into one place.

SAP Data Warehouse Cloud has a new concept called spaces, which is a logical area that can be created for each line of business inside an organization. Here, the semantics are built with a natural language search index to provide specific key performance indicator (KPI) names based on the line of business spaces. This way, business users can create their reports. It's a reincarnation of the old SAP BW workspace concept that never fully materialized, partly due to the complexity of earlier SAP BW releases. On the cloud, however, you can have business power users work in physically separated systems without impacting the managed schemas and also without being restricted by central maintenance operations and policies. Today they often use Microsoft's PowerBI to break free from the IT department. With SAP Data Warehouse Cloud and its integrated SAP Analytics Cloud, they have a chance to come back to SAP and avoid all the data integration and replication issues that come with third-party applications. However, independent of the cloud data warehousing solution you deploy, you always need to source data from a structured backend such as SAP S/4HANA. And SAP BW/4HANA offers by far the best business content accessing SAP S/4HANA. Thus, it seemed logical

to have SAP BW/4HANA views replicated into SAP Data Warehouse Cloud. This integration would take anybody years to replicate.

You can model in SAP Data Warehouse Cloud using two methods, writing SQL code or using graphical user interfaces. Graphical modeling involves frontend business users creating their models, whereas the SQL modeling would need backend power users within the business. SAP Data Warehouse Cloud comes with pre-installed standard business content across different lines of business.

SAP Data Warehouse Cloud has even more value when connected to on-premise data sources or uses SAP BW/4HANA as a source for metadata. There are two options: connecting to the application server, allowing access only to data dictionary objects, or connecting to the SAP HANA database, allowing access to all tables and views. We think connecting to SAP HANA provides you with more options and does not force you to model everything at the application layer. Thus, we'll show you how to set up a connection to the middleware called the data provisioning agent. Please keep in mind that the SAP HANA modeler usually requires a full-use license when persistent objects (tables) are created outside the SAP BW/4HANA schema.

4.4.1 Installing the Data Provisioning Agent

To enable connections to on-premise sources, SAP Data Warehouse Cloud leverages SAP HANA smart data integration (SDI) and its data provisioning agent. The data provisioning agent functions as a gateway to SAP Data Warehouse Cloud.

You can download the latest data provisioning agent version from the SAP Software Center (*http://s-prs.co/v454424*). At the time of this book's publication, it is data provisioning agent HANA DP AGENT 2.0. Remember to select your operating system before downloading the agent. You can install the data provisioning agent as a standalone installation or alongside an existing SAP HANA or SAP BW/4HANA installation on a Windows or Linux machine. The benefit of a co-located installation lies in the shared resources and the ease of maintenance, as no network issues need to be dealt with when you install the data provisioning agent. The agent uses, by default, a nonsecure channel when communicating with the SAP HANA server. To enable secure communication, you must configure SSL with the data provisioning agent configuration tool after installation. Therefore, our preferred architecture is having the agent run on the same Linux machine as SAP HANA does. You usually have full control over this host such that, for example, logs can be viewed and restarts can be performed, if necessary. In this case, our host is a Linux box running as an Intel x86 processor instance on AWS.

One advantage of installing the data provisioning agent on your SAP HANA box is that you already have an OS-level user other than root for it: `hdbadm`. This gives you a head start for exploring the agent. However, you should use a separate user like `DPA_AGENT` with permissions set up following your company's security requirements in a produc-

tion environment. In any case, your data provisioning agent user needs to have read, write, and execute rights on the data provisioning agent folders, files, and programs. That's quite easy to do: Open your terminal window and switch into your download directory where your installation file is located. In our case, this is */Users/luedtke/Downloads/IMDB_DPAGENT200_04P_12-70002516.zip*. We want to transfer this to our SAP HANA box on AWS without unpacking it. The reason is that you want to preserve the directory structure when unpacking it. Cloud providers also charge you substantially more for data transfers than for computing or storage resources in the same time window. You thus want to minimize the number of bytes you send across the wire. We therefore first need to check that all the required structures are in place. For convenience reasons, open a second terminal and switch into the directory which holds your PEM file that is required to authenticate you against AWS. In our case, it's in */Users/luedtke/Documents/SAP+BW_4HANA+20.pem*.

You might have to tighten the permissions of your PEM file when you logged into your preconfigured SAP BW/4HANA 2.0 instance. Simply type into the command line:

```
chmod 400 SAP+BW_4HANA+2.0.pem
```

Now securely log into your SAP HANA box issuing the command:

```
ssh -i SAP+BW_4HANA+20.pem root@3.226.234.32
```

Here SSH uses your PEM file instead of a password to log in as your `root` user onto your SAP HANA box. The box's IP address will obviously vary depending on your instance. You'll see an answer like Listing 4.3.

```
Last login: Fri Oct 11 19:49:53 on ttys000
luedtke@BIPORTAL12 ~ % cd Documents
luedtke@BIPORTAL12 Documents % ssh -i SAP+BW_4HANA+20.pem root@3.226.234.32
Last login: Fri Oct 11 11:34:52 2019 from 79.196.248.207
SUSE Linux Enterprise Server 12 SP3 x86_64 (64-bit)
- Configured for SAP Cloud Appliance Library (SAPCAL)
As "root" (sudo or sudo -i) use the:
  - zypper command for package management
  - yast command for configuration management
Management and Config: https://www.suse.com/suse-in-the-cloud-basics
Documentation: http://www.suse.com/documentation/sles-12/
Forum: https://forums.suse.com/forumdisplay.php?93-SUSE-Public-Cloud
Have a lot of fun...
sid-a4h-hdb:~ #
```

Listing 4.3 Logon as Root onto AWS

Next, ensure that your user `hdbadm` has all the required access rights, as shown in Listing 4.4.


```
sid-a4h-hdb:~ # chmod -R 777 /usr/sap
sid-a4h-hdb:~ # sudo usermod -a -G root hdbadm
```

Listing 4.4 Enhancing Permissions of User hdbadm for the Data Provisioning Agent

The `chmod` command gives read, write, and execution rights (code 777) to all users on the */usr/sap* directory. The recursive option `-R` denotes that the same rights apply to all directories below.

The second command assigns `hdbadm` to the root group. To install the data provisioning agent on Windows, you must use the `administrator` user or a user in the administrator group. On Linux, the agent user that installs the data provisioning agent is also the installation owner. Ideally, you should be logged in as this user when starting the agent service. We used `hdbadm` mainly for convenience because this user already exists, but this should not be done in a production system. Thus, within your company, create or use an existing non-root agent user authorized to open a display and possessing full read and write access to the intended installation location. Now you can switch the user to `hdbadm` and create the default directory for your data provisioning agent installation, as shown in Listing 4.5.

```
sid-a4h-hdb:~ # su - hdbadm
hdbadm@sid-a4h-hdb:/usr/sap/HDB/HDB02> cd /usr/sap
hdbadm@sid-a4h-hdb:/usr/sap > mkdir dataprovagent
hdbadm@sid-a4h-hdb:/usr/sap > cd dataprovagent
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent> mkdir install
```

Listing 4.5 Creating Directories for the Data Provisioning Agent

The install directory holds the data provisioning agent ZIP file, which we can now transfer into it from the first terminal window, as shown in Listing 4.6.

```
luedtke@BIPORTAL12 Downloads% scp -i /Users/luedtke/Documents/SAP+BW_4HANA+
20.pem IMDB_DPAGENT200_04P_12-70002516.zip root@3.226.234.32:/usr/sap/
dataprovagent
```

Listing 4.6 Copying Local Files onto the AWS Server

The `scp` (secure copy) command transfers the data provisioning agent installation file into its default directory. You can also use the `scp` command to later download ini or log files to your local directory, as shown in Listing 4.7.

```
scp -i /Users/luedtke/Documents/SAP+BW_4HANA+20.pem root@3.226.234.32:/usr/sap/
dataprovagent/dpagentconfig.ini /Users/luedtke/Documents
```

Listing 4.7 Copying Files from the AWS Server into the Local Directory

From here, the unzip utility can expand it into the install subdirectory. Enter this command from a second terminal window:

```
unzip /usr/sap/dataprovagent/IMDB_DPAGENT200_04P_12-70002516.zip -d /usr/sap/
dataprovagent/install/
```

You can see where to add it in Listing 4.8.

```
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent> unzip /usr/sap/dataprovagent/IMDB_
DPAGENT200_04P_12-70002516.zip -d /usr/sap/dataprovagent/install/
Archive:  /usr/sap/dataprovagent/IMDB_DPAGENT200_04P_12-
70002516.zip  inflating: /usr/sap/dataprovagent/install/HANA_DP_AGENT_20_LIN_
X86_64/hdbinst  (..)
inflating: /usr/sap/dataprovagent/install/SIGNATURE.SMF
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent> cd install
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install>
```

Listing 4.8 Unzipping Data Provisioning Agent Files on the AWS Server

You receive several messages, as shown above. If this is your second attempt you're asked if the old files should be replaced. Confirm with "A" (all). The `-d` option lets you specify the target directory.

If you installed an earlier data provisioning agent version, uninstall it first as shown in Listing 4.9.

```
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install> ./hdbuninst
SAP HANA Lifecycle Management - Uninstallation 2.4.21
************************************************
Do you want to uninstall the following installation?
SAP HANA Data Provisioning Agent 2.4.1.2.0 /usr/sap/dataprovagent
Confirm  (y/n): y
Uninstalling SAP HANA Data Provisioning Agent...
Removing ds-lite configuration ...
Uninstalling package Installer...
Uninstalling package dataprovagent...
Uninstallation done.
Log file written to '/var/tmp/hdb_dataprovagent_uninstall_2019-10-11_18.32.16/
hdbuninst_dataprovagent.log' on host 'sid-a4h-hdb'.
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install>
```

Listing 4.9 Removing a Previous Data Provisioning Agent Installation on AWS

If you're unsure whether an earlier data provisioning agent version exists, don't worry: You'd receive the message, "SAP HANA Data Provisioning Agent installation kit detected," upon execution of the installation command.

There are a few packages that your installation depends on. If they're already present on your system, you can skip this part. However, the images delivered on SAP Cloud Appliance Library usually come plain and simple, containing just the necessary SUSE server installation. If you prefer either a graphical client directly connected to your device or a secure shell connection (SSH), you'd have to install the following packages first.

To install the data provisioning agent on Linux, GTK 2 must be installed so you can run the installation tool:

```
Zypper install gtkdevel2
```

This installs the development files for the GTK+ toolkit library (version 2), required by the GUI library for the X11 windows server. You receive a message like Listing 4.10.

```
Loading repository data...
Reading installed packages...
Resolving package dependencies...
```

Listing 4.10 Zypper Package Manager on AWS Server

A local X11 graphical environment and an X server must be available on the machine where you install it. For a full installation, i.e., the X11 server and X11 desktop environment, run a command such as:

```
zypper install -t pattern gnome-basic
```

It will install potentially many packages. However, compared to the overall installation footprint, this is negligible. Installing an X11 server is obviously beyond the scope of this book, and the process may change over time. However, we'll show you the steps we performed to give you the idea. You now need to configure your X11 server by editing the file */etc/ssh/sshd_config*. Set the options as shown in Listing 4.11.

```
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes
```

Listing 4.11 X11 Setup on the AWS Server

Run the `visudo` command. A vi text editor window opens with */etc/sudoers* on it. This is useful, as it also checks for syntax errors when you exit and allows you to re-edit it. Look for the line that reads `Defaults env_reset`, and change it to `Defaults !env_reset`. This tells `SUDO` never to zero out environment variables.

Then look for the line that reads `Defaults env_keep` and add the word `DISPLAY` to the end of the line (inside the quotation marks).

Next, create a new file, */etc/profile.d/copycookie.sh*, with the content shown in Listing 4.12.

```
if [ ! -z $ORIGINALUSER ] && [ $UID -eq 0 ]; then
HOMEDIR=`getent passwd $ORIGINALUSER | rev | cut -d\: -f2 | rev`
if [ ! -d ${HOMEDIR} ]; then
echo "Could not determine the homedir for user  $ORIGINALUSER"
else
echo "Copying Xorg cookie..."
cp ${HOMEDIR}/.Xauthority ~
fi
fi
```

Listing 4.12 X11 Copy Cookie Script on the AWS Server

From now on, all users that can use `SUDO` will have the proper environment variables and Xorg authorization cookies to run graphical programs as root.

Last, you might also have to configure your SUSE firewall settings (see *http://s-prs.co/v454425*). Edit */usr/share/doc/packages/SuSEfirewall2/SuSEfirewall2.sysconfig*. On our installation, we had to set the values shown in Listing 4.13.

```
FW_DEV_EXT="dsl0"
FW_DEV_INT="eth0"
FW_DEV_EXT="eth2"
FW_DEV_INT="eth0"
FW_DEV_DMZ="eth1"
FW_DEV_EXT="eth3 eth4"
FW_DEV_INT="eth0 eth1"
FW_DEV_DMZ="eth2"
```

Listing 4.13 X11 Display Options on the AWS Server

Ubuntu RDP should work out of the box. In macOS, however, it's usually not shipped with a compatible X11 client. In this case, you would have to install it first: Install XQuartz from *https://www.xquartz.org/* on your Mac, which is the official X server software for Mac. Then run **Applications • Utilities • XQuartz.app**. Right-click on the XQuartz icon in the dock and select **Applications • Terminal**. This should bring up a new xterm terminal window. In this xterm window, add ssh into your SUSE server using the -X or -Y argument (secure X11 forwarding). For example:

```
ssh -Y -i SAP+BW_4HANA+20.pem root@3.226.234.32
```

This lets you run desktop applications like the SAP download manager (see *http://s-prs.co/v454426*). This is a program file you should install if you want to download multiple files at once or if you want to schedule downloads to run at a later time. It lists software files ("objects") that you added to your download basket in the SAP ONE support launchpad. To run the SAP download manager, a recent version of the Java Runtime Environment (JRE) must be installed on your local computer. Check if JRE 1.7 or newer is installed with `java -version`. If you don't want to use the JRE provided by SAP for

licensing or technical reasons, you can install JRE 1.5 or higher. The data provisioning agent supports only 64-bit JREs.

The JRE is included in the JDK package shown Listing 4.14. You would obviously have to replace the JDK version with your standard or latest version.

```
sid-a4h-hdb:/usr/lib64/jvm/java # zypper install java-1_8_0-openjdk
Refreshing service 'SMT-http_smt-ec2_susecloud_net'.
Refreshing service 'cloud_update'.
Building repository 'SLE-Module-Adv-Systems-Management12-
Pool' cache ..........................................................[done]
```

Listing 4.14 Java JDK Installation on the AWS Server

Installation files are often compressed using SAP's compression program, called SAPCAR. You can then extract SAR files using the command:

```
sapcar -xvf *.sar
```

This command preserves the folder structure of the initially compressed archive. SAPCAR is obtained from the SAP Download Center.

Running the SAP download manager is required whenever your Internet connection's upload speed is too low for the rather large SAP installations. However, as the data provisioning agent is rather small, you can upload it to your SUSE server as shown for other files earlier. Before you start the installation of your data provisioning agent, check for the created subdirectory that contains installation files, as in Listing 4.15.

```
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install> ls -l
total 16
drwxr-xr-x 4 hdbadm sapsys  4096 Oct 11 18:35 HANA_DP_AGENT_20_LIN_X86_64
-rw-r----- 1 hdbadm sapsys 12242 Sep  5 23:54 SIGNATURE.SMF
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install> cd HANA_DP_AGENT_20_LIN_X86_64/
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install/HANA_DP_AGENT_20_LIN_X86_64> ls -l
total 72
-rwxr-xr-x 1 hdbadm sapsys 14600 Aug 30  2018 hdbclientreg
-rwxr-xr-x 1 hdbadm sapsys 14600 Aug 30  2018 hdbinst
-rwxr-xr-x 1 hdbadm sapsys 14600 Aug 30  2018 hdbsetup
-rwxr-xr-x 1 hdbadm sapsys 14600 Aug 30  2018 hdbuninst
drwxr-xr-x 7 hdbadm sapsys  4096 Oct 11 18:35 instruntime
drwxr-xr-x 3 hdbadm sapsys  4096 Oct 11 18:36 packages
```

Listing 4.15 Data Provisioning Agent Installation Files on the SAP HANA Linux Server

Here, you find the `hdbinst` program that installs your data provisioning agent using the command line. If you installed the GTK+ package properly, you can run `hdbsetup` to start the GUI installer. Otherwise, execute `hdbinst` as in Listing 4.16.

```
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install/HANA_DP_AGENT_20_LIN_X86_
64> ./hdbinst
```

Listing 4.16 Data Provisioning Agent Installation Directory on AWS

You now receive several messages describing the installation steps, as shown in Listing 4.17.

```
SAP HANA Lifecycle Management - dataprovagent Installation 2.4.1.2.0
************************************************************
Enter Installation path [/usr/sap/dataprovagent]:
Enter Agent Listener Port [5050]:
Enter Agent Administration Port [5051]:
Enter Custom JRE directory (to use bundled JRE, leave it blank):
Checking installation...
Preparing package 'dataprovagent'...
Preparing package 'Installer'...
Installing SAP HANA Data Provisioning Agent to /usr/sap/dataprovagent...
Installing package 'dataprovagent'...
Installing package 'Installer'...
chmod: changing permissions of '/usr/sap/dataprovagent': Operation not permitted
chmod: changing permissions of '/usr/sap/dataprovagent/
install': Operation not permitted
chmod: changing permissions of '/usr/sap/dataprovagent/IMDB_DPAGENT200_04P_12-
70002516.zip': Operation not permitted
############################################################
IMPORTANT NOTE:
############################################################
You have installed Data Provisioning Agent with default settings. The default
settings use insecure channel to communicate with HANA server. For secure
communication, SSL settings must be configured via the Data Provisioning Agent
Configuration tool after the installation is complete. Please see SAP HANA
Smart Data Integration Installation and Configuration Guide for details on
configuring SSL between the Agent and HANA server.
############################################################
Configuring ds-lite ...
Installation done
Log file written to '/var/tmp/hdb_dataprovagent_2019-10-11_18.44.26_9801/
hdbinst_dataprovagent.log' on host 'sid-a4h-hdb'.
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/install/HANA_DP_AGENT_20_LIN_X86_64>
```

Listing 4.17 Data Provisioning Agent Installation on the AWS Server

Since we're using the data provisioning agent default directory, simply confirm all messages by pressing the Enter key, including the agent's default ports. If you don't

want to use the SAP-provided JRE, you can specify your own. The *chmod* error messages shouldn't concern you, as you provided all subdirectories with access rights earlier.

After installing the data provisioning agent, we recommend that you review the installation log file for any errors and take any necessary corrective actions.

This command shows you all available logs:

```
ls /var/tmp/hdb_dataprovagent*
```

Look into the most recent one, as in Listing 4.18.

```
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent> ls /var/tmp/hdb_dataprovagent_
2019-10-11_18.44.26_9801/hdbinst_dataprovagent.log
```

Listing 4.18 Data Provisioning Agent Installation Log on the AWS Server

Here you'd replace the timestamp 2019-10-11_18.44.26_9801 with your latest log.

Congratulations! You've successfully installed the data provisioning agent for Linux. Next, we'll show you how to configure it to your environment.

4.4.2 Connecting and Configuring the Data Provisioning Agent

To register the data provisioning agent with SAP Data Warehouse Cloud, the user in SAP Data Warehouse Cloud needs to have the administrator role.

Log on to SAP Data Warehouse Cloud and from the main menu and open **Administration**.

In the **Administration** section, select **Create Agent Communication Credentials**. In the dialog box shown in Figure 4.54, enter a unique name like "Agent_1" for your new agent registration. Select **Create**.

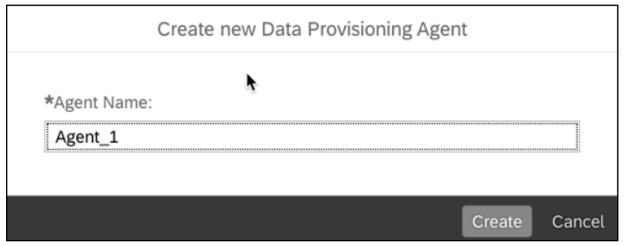


Figure 4.54 Creating a New SAP Data Provisioning Agent

Registration Name

Remember, the registration name cannot be changed later.

In the **Agent settings** dialog, you get SAP HANA server information from SAP Data Warehouse Cloud. As you can see from the **HANA User Name for Agent Messaging** field as shown in Figure 4.55, the agent name gets converted into uppercase letters and is prefixed with the string DP_MSG_, denoting that this SAP HANA user inside SAP Data Warehouse Cloud is used to exchange messages for the data provisioning agent.

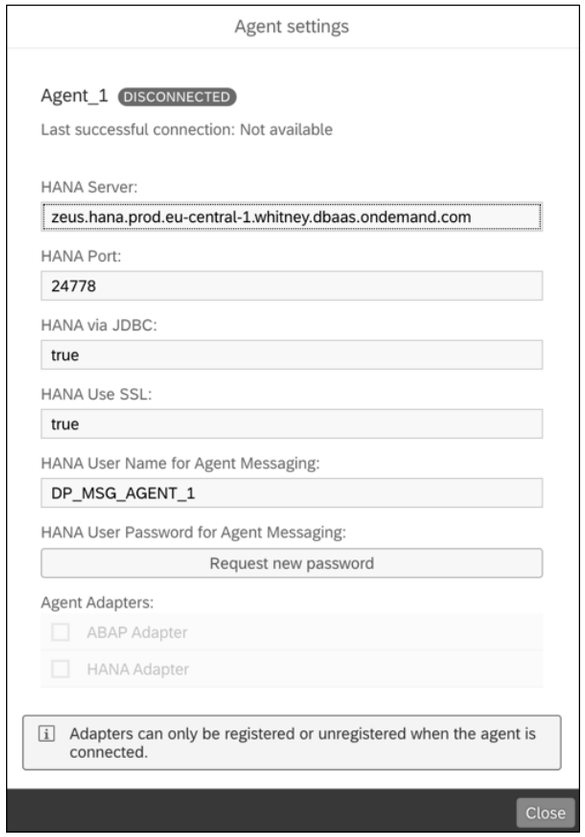


Figure 4.55 Lookup of SAP Data Provisioning Agent Details

Next, copy all the details into a text file, as you'll need them in the next step for configuring the data provisioning agent on your host.

Go back to your terminal window. If you installed the data provisioning agent on AWS, set the *cloud deployment parameter* to AWS by opening the file */usr/sap/dataprovagent/dpagentconfig.ini* in an Unix text editor like vi and setting the value to `AWS_`, as shown in Listing 4.19.

```
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent> vi dpagentconfig.ini
cloud.deployment=AWS_
```

Listing 4.19 Editing DPA Configuration Script dpagentconfig.ini

The text editor vi is a standard tool on all Linux and Unix platforms. It's a bit tricky to use, though. Enter "i" to switch to insert mode. Then use the arrow keys to navigate to the insertion position. Enter all the remaining parameters as you copied them earlier from the SAP Data Warehouse Cloud administration page. In this example, we entered the details as shown in Listing 4.20.

```
agent.name=DP_MSG_AGENT_1
hana.port=24778
hana.onCloud=false
hana.useSSL=true
hana.server=zeus.hana.prod.eu-central-1.whitney.dbaas.ondemand.com
jdbc.enabled=true
jdbc.host=zeus.hana.prod.eu-central-1.whitney.dbaas.ondemand.com
jdbc.port=24778
jdbc.encrypt=true
```

Listing 4.20 Data Provisioning Agent Configuration Settings in dpagentconfig.ini

Assign the server IP address or host name to **hana.server** and the HTTP port to **hana.port**. If you need a proxy to access the Internet due to a corporate firewall, you must also set the information shown in Listing 4.21.

```
cloud.useProxy=true
proxyHost=<your Proxy host>
proxyPort=<your Proxy port>
proxyType=http
jdbc.proxyHostname=<your Proxy host>
jdbc.proxyPort=<your Proxy Port>
jdbc.useProxy=true
jdbc.additionalParameters=proxyHttp=true
jdbc.proxytype=http
```

Listing 4.21 Optional Proxy Settings for the Data Provisioning Agent

Save the changes made to *dpagentconfig.ini* by typing "<ESC>:wq!". You'll find yourself back on the command line.

Adapters can only be registered or unregistered when the agent is connected. You therefore need to start your data provisioning agent first. On Linux, you simply need to run the commands in Listing 4.22.

```
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent> cd bin
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/bin> ./agentcli.sh --configAgent
```

Listing 4.22 Launching the Data Provisioning Agent Management Script agentcli.sh with the configAgent Option

On Windows the agent client is not a shell script but a .bat file: *agentcli.bat –configAgent*.

Once this is executed, the menu in Listing 4.23 appears.

```
************************************************************
                DPAgent Configuration Tool
************************************************************
1. Agent Status
2. Start or Stop Agent
3. Agent Preferences
4. Remote Source Credentials
5. SSL Keystores
6. SAP HANA Connection
7. Agent Registration
8. Adapter Registration
9. Custom Adapters
10. Agent & Adapter Versions
q. Quit
b. Back
************************************************************
Enter Option:1
Agent is not running. Start the agent and try again.
Press Enter to continue...
```

Listing 4.23 Data Provisioning Agent Status Check

To start your data provisioning agent press the Enter key and then enter "2" as the option. This will open another menu, as shown in Listing 4.24.

```
Enter Option:2
************************************************************
                   Start or Stop Agent
************************************************************
1. Start Agent
2. Stop Agent
3. Ping Agent
q. Quit
b. Back
************************************************************
Enter Option:1
Agent service location: /usr/sap/dataprovagent/bin/dpagent_service.sh
dpagent_service.sh location: /usr/sap/dataprovagent/bin/dpagent_service.sh
Starting agent service. Command: /usr/sap/dataprovagent/bin/dpagent_
service.sh start
Start service return code: 0
Press Enter to continue...
```

Listing 4.24 Data Provisioning Agent Startup on the AWS Server

After a few seconds, your data provisioning agent should be up and running. You can check this by pinging your agent with menu option "3". The result will be as shown in Listing 4.25.

```
************************************************************
Enter Option:3
Ping successful
Press Enter to continue...
```

Listing 4.25 Data Provisioning Agent Alive Check on the AWS Server

With the agent running, you can now verify its status via menu option "1". You should see the information in Listing 4.26.

```
************************************************************
Enter Option:1
************************************************************
Agent                    Status: Running
                         Listener port: 5050
                         Admin port: 5051
                         Agent Name:
HANA Connection          Server: zeus.hana.prod.eu-central-
1.whitney.dbaas.ondemand.com
                         Port: 24778
                         Protocol: JDBC
                         Agent Admin HANA user:
************************************************************
Press Enter to continue...
```

Listing 4.26 Data Provisioning Agent Installation Status

Your data provisioning agent is now active. But before it's ready to be connected to your SAP Data Warehouse Cloud instance, you need to configure the connection details. So, first, you need to shut down your agent with menu option "2" followed by option "q" to quit from the menu, as shown in Listing 4.27.

```
************************************************************
Enter Option:2
Stopping dpagent_service. Command: /usr/sap/dataprovagent/bin/dpagent_
service.sh stop
Stop Service return code: 0
Agent was shutdown gracefully
Press Enter to continue...
************************************************************
Enter Option:q
hdbadm@sid-a4h-hdb:/usr/sap/dataprovagent/bin>
```

Listing 4.27 Data Provisioning Agent Shut Down on the AWS Server

This pattern of stopping and then starting the agent is required whenever you make a configuration change to the agent.

The command-line agent configuration tool requires the `DPA_INSTANCE` environment variable to be set to the installation root location (*usr/sap/dataprovagent*):

```
export DPA_INSTANCE=/usr/sap/dataprovagent
```

Multiple instances of the data provisioning agent may be installed on a single Linux host. Be sure that you set `DPA_INSTANCE` to the instance you want to modify before starting the configuration tool. If you do not set the environment variable correctly, you may unintentionally modify a different agent instance's configuration.

SAP Data Warehouse Cloud is built on top of SAP HANA XS, the extended application services framework. In this instance, the data provisioning agent master user is Agent 1, as defined earlier. You can set the credentials for this SAP HANA XS user by running the shell script

```
usr/sap/dataprovagent/bin/agentcli.sh --setSecureProperty
```

Enter the username presented in the SAP Data Warehouse Cloud **Agent settings** dialog (Figure 4.55): `DP_MSG_AGENT_1`.

Next, select option "2" and enter the agent's SAP HANA XS password presented in the **Agent settings** dialog from step 1. In our case, it's `QylFYOxqawh8Ss6d`.

You can request a new password at any time from the **Agent setting** dialog shown above by clicking the **Request new password** button. The new password is then saved to SAP Data Warehouse Cloud and is ready to be used by the data provisioning agent. Type "q" to quit the script.

Now, stop and restart the data provisioning agent by again running the script:

```
usr/sap/dataprovagent/bin/agentcli.sh -configAgent
```

To stop the agent, select option "2", **Start or Stop Agent**, and then option "2", **Stop Agent**. Next, select option "1", **Start Agent**, to restart your Agent 1. Then, select option "1", **Agent Status**, to check the connection status. If the connection succeeded, you should see the message **Agent 1 connected to HANA: Yes**. Select "q" to quit the script.

If you have kept the **Agent settings** dialog in SAP Data Warehouse Cloud open, you can close it now. Your data provisioning agent is now connected. If the list of the registered data provisioning agents doesn't display the updated connection status, click on the **Refresh** button.

4.4.3 Registering the Data Provisioning Adapters

In SAP Data Warehouse Cloud, you need to register the data provisioning adapters that are needed to connect to SAP HANA and ABAP on-premise sources.

In the **Administration** screen, select **Edit** on your newly connected data provisioning agent. In the **Agent settings** dialog, under **Agent Adapters**, select both **HANA Adapter** and **ABAP Adapter** (Figure 4.56).

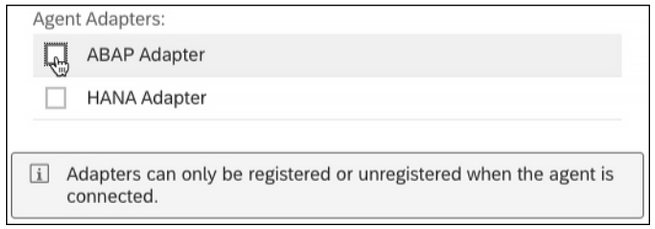


Figure 4.56 Selecting Both Adapters for ABAP and SAP HANA Systems in SAP Data Warehouse Cloud

Close the dialog. The selected adapters are now registered with SAP Data Warehouse Cloud. The data provisioning agent is connected, and the adapters are registered and available for creating connections to SAP HANA and SAP ABAP on-premise sources like your SAP BW/4HANA system.

4.5 SAP Data Intelligence

SAP Data Intelligence is a data sharing, pipelining, and orchestration solution that helps companies accelerate and expand the flow of data across their data landscapes. Metadata in SAP Data Intelligence is managed via the *metadata explorer*. A data scientist searching for specific data is presented with all the *catalogs* with a potential for such data.

You might ask yourself why there is a need for yet another data management platform. The answer is simple: SAP BW/4HANA expects structured data and a thought-through plan of how the data is going to be used. As you've seen above, all application data resides in separated schemas. If there is broader data to be processed with new features and attributes, the schema needs to be changed. Now, imagine you've built a LSA++ (see Chapter 2) based on your base schema with hundreds of data consumers and potentially thousands of queries. If you're lucky, the upper layers continue to function as before, but still without the new data fields and attributes. The lower layers certainly have to change. Since they potentially carry terabytes of data, that's not an easy task, although a column-oriented in-memory database like SAP HANA greatly simplifies this task, as fields are represented as columns that can simply be added. We'd call this a birth defect of the data warehousing concept, and many attempts have been made to fix it, e.g., the data vault approach. However, dropping or changing otherwise referenced columns remains the main issue.

Now, imagine there is no schema—the data is not separated at all. Without a schema there is also no planning and entity-relationship modeling required in advance. This sounds very similar to the spreadsheet approach: Just store any kind of CSV file in a folder, open it, and decide what to do with it. This concept is called schema-on-read: The user decides at runtime how to interpret the data. Schema-less data management solutions, however, use a more sophisticated concept that's similar to JSON, one that does not force you to use a specific delimiter or adhere to a fixed number of columns. It works more like business cards: Some people still have a fax number, some don't. In a relational model you'd have to carry a mostly empty fax field for all your contacts. And in case a new technology pops up like Whatsapp or WeChat, you'd have to change your schema with all the implications described above. With a schema-less data management solution, however, there are just records with a fax field and some records without it. Now, when you try to access those files and look for the fax number, it pops up if it exists and displays a NULL value if not. The system won't crash just because records have a different structure. Now, try to upload a CSV file with records having different structures into a relational database: It crashes at the first deviation from the expected type and field position. Data warehousing, however, is required to harmonize data. Often, stringent rules have to be applied: An empty or nonexistent fax field might indicate that the record is corrupt and should not be processed at all. Hence, a hybrid solution is needed that allows companies to upload all external data first and then force harmonization and security rules upon it later. Solutions like Snowflake, as discussed in Section 4.4 already provide this kind of hybrid. SAP is on its way to integrating these two worlds, with SAP HANA primarily focused on structured data and SAP Data Intelligence primarily focused on semi-structured data.

SAP Data Intelligence provides visibility and access to a broad range of data systems and assets. It meets the governance and security needs of enterprises, ensuring that appropriate policy measures are in place to meet regulatory and corporate requirements. SAP Data Intelligence is also a data orchestration and management solution running on Kubernetes. All SAP Data Intelligence operations are executed as Kubernetes pods.

The first iteration of SAP Data Intelligence was released in 2017, under the product name SAP Data Hub, to deal with big data and sophisticated data orchestration. In 2019 SAP Data Hub was released as a managed service on SAP Cloud Platform, and it was this version that was first given the name SAP Data Intelligence. The name SAP Data Intelligence highlights the evolution of SAP Data Hub to operationalize data science and machine learning with the inclusion of the SAP Leonardo machine learning library. SAP Data Intelligence provides all the integration, orchestration, metadata management, connectivity, and services of SAP Data Hub with the services of the SAP Leonardo machine learning library in the cloud.

In Q1 2020 SAP Data Intelligence 3.0 was released, which extends the cloud version of SAP Data Intelligence to on-premise and BYOL (bring your own license) deployments:

- SAP Data Intelligence, cloud edition, is available on SAP Cloud Platform through a subscription or SAP Cloud Platform enterprise agreement.
- BYOL is a licensing model where SAP Data Intelligence can be deployed on-premise in your own data center, on any hyperscale public cloud (AWS, Google Cloud Platform, Microsoft Azure) or private cloud.
- Existing SAP Data Hub customers receive all cloud-based machine learning tooling as part of support packages. When release 3.0 is applied, the product will be updated to SAP Data Intelligence 3.0.

4.6 SAP S/4HANA Embedded Analytics

Data warehouses supply external BI clients with requested information. Often, large dashboards arebuilt, focusing mostly on strategic solutions like SAP Digital Boardroom. At the operational level, however, the vast majority of users have to make decisions within seconds simply based on their experience and sometimes without access to a BI client at all. Now, imagine having your BI solution built into transactions. Based on experience and current circumstances, it can recommend a decision and justify it with the predicted outcome. For example, it could tell you that increasing the grace period for a specific customer by 10% might also increase the default rate by 1% but at the same time increase customer retention by 20% and thus be justified: The analytics is embedded into transactions. The analytics engine is mostly based on SAP Analytics Cloud technology, as described earlier in this chapter. For the embedded SAP Analytics Cloud backend there are different ways to provision the data, as we'll discuss in the next two sections.

4.6.1 Embedded SAP BW in SAP S/4HANA

According to current roadmaps as of this book's publication, the SAP NetWeaver-based SAP BW 7.5 system within SAP S/4HANA will not be upgraded to SAP BW/4HANA in the foreseeable future. Hence, we recommend not relying too heavily on a soon-to-be obsolete SAP BW solution for your embedded analytics. Instead, it should be based on the underlying core data services.

4.6.2 Core Data Services for ABAP

Core data services (CDS) come in two flavors: ABAP CDS views and SAP HANA CDS views. Although they share the same name and follow a similar concept, the same CDS code typically cannot be run in both scenarios. The idea behind CDS is to push as much logic down to the database as possible. In both scenarios the CDS description is eventu-

ally compiled into SAP HANA-executable code. However, only ABAP CDS can be consumed in ABAP like data dictionary objects. SAP HANA CDS views are part of the SAP HANA XSA database application layer. Since SAP S/4HANA is mainly a mixture of ABAP and ABAP CDS, implementing embedded analytics with ABAP CDS sounds reasonable. Since ABAP CDS is beyond the scope of this book, we refer you to the excellent book *Core Data Services for ABAP* by Renzo Colle, Ralf Dentzer, and Jan Hrastnik (SAP PRESS 2019).

4.7 Summary

After reading this chapter, you should now be able to set up your client tools for accessing SAP BW/4HANA. We explained how to install SAP HANA Studio and the SAP HANA client. You also learned to enable SAP BW/4HANA for SAP Analytics Cloud, the first step towards a working SAP Analytics Cloud connection. In Chapter 8, we'll discuss the options you have for using SAP Analytics Cloud in more detail.

SAP Data Warehouse Cloud is useless without data—we showed you how to set up the data provisioning agent, to establish a trusted connection between the new cloud data warehouse and SAP BW/4HANA. And, finally, we gave you a brief look at SAP Data Intelligence and SAP S/4HANA embedded analytics, two important tools that support SAP BW/4HANA.

In the next chapter, we'll take a look at security and operations tasks for SAP BW/4HANA 2.0.

Contents

Preface 15

1 SAP BW/4HANA and Its Ecosystem of Tools and Services 21

1.1 What Is Business Intelligence and How Is It Related to Digitization? 22

1.2 Merging Data Lakes and Data Warehouses 24

1.3 SAP Business Warehouse Releases 27

1.3.1 SAP Business Warehouse 7.3 on Any Database 27

1.3.2 SAP Business Warehouse 7.4 Powered by SAP HANA 28

1.3.3 SAP Business Warehouse 7.5 and SAP Business Warehouse Edition for SAP HANA 30

1.4 Navigating SAP BW/4HANA 31

1.4.1 Technical Users 31

1.4.2 Business Users 33

1.5 Deployment Scenarios 36

1.5.1 On-Premise 36

1.5.2 Public Cloud 37

1.5.3 Managed Cloud 44

1.6 Related Cloud Technologies 45

1.6.1 SAP Analytics Cloud 45

1.6.2 SAP Data Warehouse Cloud 46

1.6.3 SAP Data Intelligence 47

1.6.4 SAP Cloud Platform 49

1.7 Use Case of this Book 51

1.8 Summary 54

2 Architecture 57

2.1 The Evolution of the Layered Scalable Architecture (LSA++) 58

2.1.1 From LSA to LSA++ 58

2.1.2 Model View Controller and LSA++ 62

2.2 Business Intelligence Clients ... 67
2.3 Business Content ... 69
2.4 Data Protection Workbench ... 70
2.5 Multi-Temperature Data Management ... 72
2.5.1 Data Lifecycle Manager ... 72
2.5.2 Dynamic Tiering ... 75
2.6 Summary ... 77

3 Installation 79

3.1 Administering Instances in a Public Cloud Environment ... 79
3.1.1 The Architecture of Amazon Web Services ... 79
3.1.2 Creating an Account on SAP Cloud Appliance Library and Aligning It with Your Cloud Provider ... 83
3.1.3 Amazon Web Services Management Access from Windows ... 99
3.2 Setting Up the SAP BW/4HANA Client ... 100
3.2.1 Setting Up the SAP GUI Connection ... 101
3.2.2 Creating an Administrative User ... 104
3.2.3 Running Task Lists ... 109
3.2.4 Installing an SAP BW/4HANA License ... 111
3.2.5 Installing an SAP HANA License ... 113
3.2.6 Installing Support Packages ... 114
3.2.7 Setting Up the Generation of External SAP HANA Views ... 125
3.3 Installing SAP Business Planning and Consolidation, version for SAP BW/4HANA ... 127
3.3.1 Download ... 128
3.3.2 Add-On Installation ... 128
3.3.3 Configuration ... 134
3.4 Summary ... 138

4 Tools 139

4.1 SAP HANA Studio Installation and Configuration ... 139
4.1.1 Connecting the SAP BW/4HANA Server and SAP BW Modeling Tools via a Remote Desktop ... 140
4.1.2 Installing Eclipse and the SAP BW Modeling Tools ... 148
4.1.3 Connecting to SAP HANA ... 152

4.1.4 Creating an SAP HANA Archive Link ... 153
4.1.5 Configuring the SAP HANA Usage Type ... 155
4.1.6 Installing the ABAP Development Tools for Eclipse ... 155
4.1.7 Creating a Repository Workspace and Package ... 160
4.1.8 Creating a Developer Package ... 161
4.1.9 Error Handling ... 162
4.2 SAP HANA Client Tool Installation ... 167
4.2.1 Creating an ODBC Data Source for SAP HANA ... 168
4.2.2 HDBSQL Command-Line Tool ... 170
4.3 Enabling SAP BW/4HANA for SAP Analytics Cloud ... 171
4.4 Connecting to SAP Data Warehouse Cloud ... 176
4.4.1 Installing the Data Provisioning Agent ... 177
4.4.2 Connecting and Configuring the Data Provisioning Agent ... 185
4.4.3 Registering the Data Provisioning Adapters ... 190
4.5 SAP Data Intelligence ... 191
4.6 SAP S/4HANA Embedded Analytics ... 193
4.6.1 Embedded SAP BW in SAP S/4HANA ... 193
4.6.2 Core Data Services for ABAP ... 193
4.7 Summary ... 194

5 Security and Operations 195

5.1 SAP BW/4HANA Authorizations ... 196
5.1.1 Protected Actions ... 196
5.1.2 Protected Data Access ... 202
5.1.3 Granted Actions and Data Access ... 205
5.2 Read Access Logging ... 212
5.2.1 Configuring Read Access Logging ... 212
5.2.2 Enabling Read Access Logging on a DataStore Object (Advanced) ... 218
5.3 SAP HANA Authorizations ... 223
5.3.1 Creating SAP HANA Users Manually ... 223
5.3.2 Authorization Control via Decentralized User Administration ... 226
5.3.3 Authorization Control within the SAP HANA Database ... 227
5.3.4 Setting Up the Authorization Concept for the SAP HANA Database ... 229
5.3.5 Representation of Authorization Trees ... 247
5.3.6 Detecting Authorization Errors ... 250
5.3.7 Setting Up Privileges for ABAP Development Tools ... 253
5.3.8 Modeling Analytical Privileges ... 254

5.4 SAP BW/4HANA Transports ... 258
5.4.1 Change and Transport Management ... 259
5.4.2 Transport Management in SAP HANA ... 261
5.4.3 Mixed Transports ... 265
5.5 SAP BW/4HANA Operations ... 266
5.5.1 Data Warehousing Workbench ... 267
5.5.2 Process Chains App ... 268
5.5.3 Process Types ... 270
5.5.4 Monitoring Views for Core Data Services ... 272
5.6 Summary ... 273

6 Basic Data Modeling 275

6.1 Modeling Objects ... 275
6.1.1 Creating InfoAreas ... 276
6.1.2 Creating Source Systems ... 276
6.1.3 Creating Application Components ... 286
6.1.4 Creating Data Sources ... 287
6.1.5 Creating Data Flow Objects ... 303
6.1.6 Creating Open ODS Views ... 308
6.1.7 Creating InfoSources ... 320
6.1.8 Creating InfoObjects ... 323
6.1.9 Creating Transformations ... 329
6.1.10 Creating DataStore Objects (Advanced) ... 331
6.1.11 Creating Data Transfer Processes ... 359
6.1.12 Creating Open Hub Destinations ... 366
6.1.13 Creating CompositeProviders ... 373
6.1.14 Creating Queries ... 375
6.1.15 Creating Variables ... 377
6.1.16 Creating Semantic Groups ... 384
6.2 Modeling Process Chains in the SAP BW/4HANA Cockpit ... 392
6.2.1 Generating Process Chains from Data Flows ... 392
6.2.2 Meta Process Chains ... 399
6.3 Master Data Modeling ... 401
6.3.1 Surrogate ID Tables and Views ... 402
6.3.2 Text Tables ... 403
6.3.3 Tables and Views for Attributes ... 404
6.3.4 Tables and Views for Hierarchies ... 405
6.3.5 Tables and Views for Enhanced Master Data Updates ... 406

6.4 Hybrid Data Modeling with SAP HANA ... 412
6.4.1 Native DataStore Objects ... 416
6.4.2 Data Distribution Optimizer ... 418
6.5 Modeling for SAP Business Planning and Consolidation, version for SAP BW/4HANA (Embedded Configuration) ... 419
6.5.1 Creating a Planning-Enabled InfoProvider ... 419
6.5.2 Creating Data Slices on the Basis InfoProvider ... 422
6.5.3 Creating a Characteristic Relationship on the Basis InfoProvider ... 424
6.5.4 Creating an Aggregation Level ... 426
6.5.5 Creating a Planning-Enabled Query ... 428
6.5.6 Running a Planning-Enabled Query in SAP Analysis for Microsoft Office ... 430
6.6 Summary ... 433

7 Advanced Data Modeling 435

7.1 Data Distribution ... 436
7.1.1 Data Distribution in Semantically Partitioned Groups ... 436
7.1.2 Data Distribution through Partitioning ... 437
7.1.3 Data Distribution with Dynamic Tiering ... 443
7.2 Virtual Data Modeling ... 447
7.2.1 Virtual Characteristics with SAP HANA Information Views ... 447
7.2.2 Open ODS Views on Calculation Views ... 456
7.2.3 CompositeProvider on Calculation Views ... 461
7.3 Modeling CompositeProviders ... 466
7.3.1 Inner Joins in CompositeProviders ... 466
7.3.2 Outer Joins in CompositeProviders ... 467
7.3.3 Row-Level Security with CompositeProviders ... 469
7.4 Native SAP HANA Data Modeling ... 469
7.4.1 Generating External SAP HANA Views for SAP BW/4HANA Queries ... 471
7.4.2 Generating External SAP HANA Views for Characteristics ... 473
7.4.3 Generating External SAP HANA Views for DataStore Objects (Advanced) ... 478
7.4.4 Generating External SAP HANA Views for CompositeProviders ... 480
7.4.5 Naming Convention for Self-Developed Information Models ... 483
7.4.6 Modeling Calculation Views ... 485
7.4.7 Modeling Classic Analytical Privileges ... 488
7.5 Data Modeling in SAP Data Warehouse Cloud ... 491
7.5.1 Configuring SAP Data Warehouse Cloud ... 492

7.5.2 Data Modeling in Data Builder ... 500
7.5.3 Creating Entity-Relationship Models ... 506
7.5.4 SAP BW/4HANA Model Transfer ... 508

7.6 Remodeling Code with the ABAP Development Tools ... 508
7.6.1 Customer Exits in SAP BW/4HANA ... 508
7.6.2 Developing Transformations with ABAP Managed Database Procedures ... 508

7.7 Summary ... 515

8 Analytics Clients for Business Intelligence 517

8.1 SAP Analytics Cloud ... 517
8.1.1 Creating an Account ... 518
8.1.2 Live Connectivity to SAP BW/4HANA ... 520
8.1.3 Import Connectivity to SAP BW/4HANA and SAP HANA ... 526
8.1.4 Live Connectivity to SAP HANA ... 540
8.1.5 SAP Analytics Cloud, Add-In for Microsoft Office ... 552
8.1.6 Tenant Configuration ... 555
8.1.7 Cloud Model Building ... 556

8.2 SAP Lumira ... 572
8.2.1 Connecting with SAP Lumira, Designer Edition ... 573
8.2.2 Connecting with SAP Lumira, Discovery Edition ... 580

8.3 SAP Analysis for Microsoft Office ... 584

8.4 Native BI Portal Applications with SAP HANA XSA and SAP Fiori ... 586

8.5 Third-Party Clients for the Mac ... 587
8.5.1 MicroStrategy Desktop ... 587
8.5.2 Tableau Desktop ... 593

8.6 Microsoft Excel via MDX ... 596

8.7 Summary ... 599

9 Deployment Options 601

9.1 Conversion versus Transformation ... 601

9.2 Planning Your Migration to SAP BW/4HANA ... 602
9.2.1 Technical Requirements ... 602
9.2.2 Business Content for SAP BW/4HANA ... 604

9.3 SAP BW/4HANA in a Virtual Private Cloud ... 609
9.3.1 Installation Procedure in the Virtual Private Cloud ... 611
9.3.2 Capacity Planning with Amazon Web Services ... 612
9.3.3 Preconfiguration with AWS CloudFormation ... 612
9.3.4 Amazon Machine Images ... 614
9.3.5 Authorization Concept of Amazon Web Services ... 615
9.3.6 Scale-Out Architectures with AWS CloudFormation ... 618
9.3.7 SAP Landscape Management ... 619

9.4 Greenfield Installation ... 623

9.5 In-Place Conversion ... 626
9.5.1 Changes at the Reporting Level ... 628
9.5.2 Changes to Data Flows ... 630
9.5.3 Migration Cockpit for SAP BW/4HANA ... 630
9.5.4 Transfer Tool of the Starter Add-on for SAP BW/4HANA ... 637
9.5.5 Remote Conversion ... 643
9.5.6 Shell Conversion ... 643

9.6 Summary ... 644

10 Roadmap 645

10.1 Data Distribution and Management ... 645

10.2 Data Protection Workbench ... 647

10.3 Process Chains ... 647

10.4 Performance Improvements ... 648

10.5 SAP Analytics Cloud Integration ... 648

10.6 SAP Data Warehouse Cloud Integration ... 648

10.7 Summary ... 649

The Authors ... 651
Index ... 653

Index

A

ABAP CDS 281
ABAP CDS ODP
  data sources 289
ABAP Data Dictionary data type 465
ABAP Development Tools 155
  installation 155
  privileges 253
ABAP Development Tools for Eclipse 155
ABAP managed database procedures
  (AMDP) 155
  methods 265
  scripts 265
ABAP routine 509
ABAP Routine Analyzer 633
Access control lists 227
Acquisition layer 65
Administrator
  privileges 229
aDSO 305
  activation errors 421
  aggregation level 427
  characteristic relationship 424
  create 331
  data slices 422
  external SAP HANA view 478
  index 357
  landscape redistribution 437
  options 335
  partitioning 436
  patterns 335
  persistence objects 335
  planning activation errors 421
  planning objects 335
  planning-enabled 346
  querying records 342
  read access logging (RAL) 218
  remodeling 389
  semantic partitions 436
  staging (write-optimized) 349
  templates 66
  write-optimized 349
ADT
  installation 155
  privileges 253
Aggregation level 426–428
Agile 61
Agile project methodology 601
Amazon AMI 81, 609, 614
Amazon CloudWatch 81
Amazon EBS 82
Amazon EC2 ..... 82, 613
Amazon Glacier 82
Amazon S3 82
  object 93
Amazon Virtual Private Cloud 81
Amazon Web Services 611, 613
  architecture 79
  authorization 615
  billing alert 93
  CloudTrail 616
  identity and access management 615
Analyst
  role 246
Analytic view
  naming 483
Analytical privileges 228, 485
  classic 254
  model 254
  naming convention 485
  SQL-based 255
Analytics users 223, 227
Apache Drill 33
Apache Hadoop 25, 33, 72, 415
Apache Spark 33, 47
Apache Tomcat 526
Appliance 609
Application lifecycle management 263
Application privileges 229
Architecture 58
  scalable 60
Archived data 72
Array 610
Associations 313
Attribute view 448
  create 451
  naming 483
Auditing 226
Authorization
  analytical 227
  controls 226
  dependency viewer 248
  errors 250
  replication 241
  trace 250, 252

AWS
automatic recovery ... 82
Boto Library ... 93
Boto SDK ... 93
CLI user guide ... 96
fees ... 83
Python ... 93
SDK ... 93
AWS CloudFormation ... 82, 612
AWS command line interface ... 91
installation ... 94
AWS IAM ... 81
AWS management console ... 91, 616
AWS Snowball ... 82
AWS Storage Gateway ... 82
AWS Toolkit for Eclipse ... 96

B

Backup and restore procedures ... 614
BEx query ... 628
BI clients ... 67–68
Mac OS ... 587
BI Consumer Services (BICS) ... 34, 69, 586
interface ... 573
BI content ... 69
BI portal ... 586
Big data analytics ... 414
Business content
upgrade ... 122
Business developer ... 414
Business intelligence ... 517
Business process
integration ... 413
Business process reengineering ... 413

C

Calculation view ... 27, 308, 485
create ... 461
graphical ... 485
naming ... 483
SQLScript-based ... 485
Capacity planning ... 612
Catalog procedure ... 484
CDS ... 272
monitoring ... 272
Certificate authorities ... 548
Change and transport management ... 259
Change and transport system (CTS) ... 258
Change list ... 266
Characteristics ... 70, 323
naming conventions ... 323
read access type ... 453
Checklist ... 632
Clash strategy ... 73
Classic DataStore objects ... 335
Clone ... 623
Closed-loop reporting ... 70
Cloud
computing ... 36
private ... 609
Cloud connector
installation ... 526
Code pushdown ... 27
Cold data ... 72
cold store flag ... 337
Column store ... 443
Column store table
partitioned ... 77
Column views ... 125
Commodity computer cluster ... 33
Compile server ... 98
CompositeProvider ... 28, 384
calculation view ... 461
create ... 373
join types ... 466
Open ODS view ... 459
Computer network ... 80
Computer network system ... 33
Condition store ... 255
Content vendor ... 261
Controller ... 62
Core data services (CDS) ... 272
for ABAP ... 281
Core update ... 119
Corporate memory ... 65, 352
CPU
credit ... 613
virtual ... 611
Cross-origin resource sharing (CORS) ... 520
SAP HANA ... 548
Customer exit ... 508

D

Darwinism
digital ... 22
Dashboard ... 593
Data
cold ... 73
hot ... 73
unstructured ... 25
warm ... 75

Data aging ... 73
Data analysis ... 58
agile ... 61
Data builder ... 500
entity-relationship models ... 506
modeling attributes ... 502
modeling measures ... 502
SQL views ... 503
Data discovery ... 580
Data distribution ... 436
nearline storage ... 442
SAP HANA dynamic tiering ... 443
Data distribution optimizer ... 418–419
Data flow
automatic creation ... 372
create ... 303
history ... 306
migration ... 630
objects ... 304
process chains ... 393
transient ... 389
Data flow objects
template ... 303
Data flow transfer ... 627
Data integration ... 33
Data lake ... 25
Data lifecycle manager ... 72
Data management ... 58, 72
Data mart ... 65
layer ... 65
Data mining workbench ... 32
Data model ... 62
Data modeling ... 275
mixed models ... 223
Data protection workbench ... 71
Data provisioning agent ... 177, 185
Data provisioning server ... 98
Data replication ... 308
Data retention ... 33
Data science ... 61
Data silo ... 25
Data source
create ... 307
SAP ILM ... 71
Data source replication ... 287, 289
Data sources ... 287
ABAP CDS ODP ... 289
file ... 292
SAP BW/4HANA ODP ... 287
SAP HANA ODP ... 290
Data transfer process ... 359
create ... 359
monitor ... 361
Data transformations ... 60
Data type ... 464
Data volume ... 413
Data warehousing ... 26, 60
operations ... 24
with SAP HANA ... 412
Data warehousing workbench ... 106, 127, 259, 267
BI content ... 604
Database management system users ... 238
Database migration option (DMO) ... 628
Database object
permissions ... 470
Database procedure ... 484
naming ... 484
Database system
relational ... 413
DataSource
migration ... 630
DataStore object ... 27, 29
aDSO ... 29, 304, 331
for direct writing ... 350
native ... 416
standard ... 27
write-optimized ... 349
DataStore object (advanced) ... 29, 335
DB Connect ... 630
DBMS users ... 238
Default security group ... 81
Default store ... 77
Delivery unit ... 258, 261
privileges ... 231
Delta merge ... 270
Delta store ... 270
Deployment
dynamic ... 36
Descriptive language ... 257
Design patterns ... 62
Design-time object ... 234, 470
Destination ports ... 245
Developer
role ... 235
user ... 237
Development package ... 260
Digital boardroom ... 67
Digital darwinism ... 22
Digital Readiness Index (DRI) ... 23
Digitization
guidelines ... 23
Dimension ... 27
Dimension table ... 27
Direct attached storage (DAS) ... 610

Direct update DSO 350
Domain Name System (DNS) 81
Drill 33
DSO
direct update 350
DSO to aDSO mapping 335
Dynamic tiering 30, 75

E

E table 27, 344
Eclipse 36, 140
ABAP Development Tools 150
connecting to SAP BW/4HANA 152
IDE for Java Developers 148
IDE installation 148
perspectives 140
Enterprise data hub 25
Enterprise data warehousing 26
Enterprise information management 630
Error log 166
Ethernet 610
Extended storage 75
External interfaces 167
External SAP HANA view 471
External system 413

F

F table 27, 344
Fact table (F table) 27
FCP 610
Feature 313
Fibre Channel Protocol (FCP) 610
Field properties 302
File
data sources 292
source systems 284
FileZilla 100
Forecasting and replenishment (F&R) 25
Frontend tools 517
Full update 342
Full-vector scan 357

G

G view 74
GDPR 44, 71
Google Cloud Platform 49
Greenfield installation 602, 623
Group
semantic 384, 436

H

Hadoop Distributed File System (HDFS) 33
Hash algorithms 437
HDBODBC driver 68, 575
HDBSQL 170
HDD 33
High cardinality 325
High-performance computing 414
Horizontal partitioning 392
Host 81
virtual 620
Hot data 72
Hub-and-spoke architecture 25
Hyper-converged system 609
Hypervisor 609

I

IAM 615
policy 615
Identity and access management (IAM) 615
Index 357
inverted 357
Index server 98
Industry 4.0 21
InfoArea 276
create 276
InfoCube 344
InfoCube to aDSO mapping 344
SAP HANA-optimized 27, 344
InfoObject
characteristics 323
create 323
key figures 323
InfoProvider
planning-enabled 419
Information model 470
mixed models 223
naming convention 483
Information view 447
changes 127
naming convention 483
InfoSource
create 320
Infrastructure-as-a-service (IaaS) 37
Inner joins 466
In-place conversion 602
Input parameters
naming conventions 483
Instance copy 623
Instance storage 613

Instance type 613
Integration layer 60
Interface 68
IP address 80
external 80
internal 41
masking 80
packet 80
private 80
public 80

J

Java connector 526
installation 531
Java Database Connectivity (JDBC) 34
JSON 96

K

Key
semantic 366
technical 366
Key figures 70, 323
naming conventions 323
propagation layer 323

L

LAN 611
Landscape transformation 32
Layer
integration 60
semantic 33, 60
supply 60
Layered scalable architecture (LSA) 28, 58
License 111
Lightweight Directory Access Protocol
(LDAP) 227
Linux server
connection 99
Loading time 32
Local area network (LAN) 611
Local network
virtual 611
Logical unit number (LUN) 610
Low code application platform (LCAP) 57
LSA 191
MVC pattern 62
LSA++ 28, 64

M

Main memory 33
Main store 270
Maintenance planner 117
Managed persistence object 74
Management information systems (MIS) 62
Master data modeling 401
Master data read class 326
MDX 34, 68
Meta chains 399
Metadata explorer 191
Microservice 615
Microsoft Azure Cloud Services 44
Microsoft Excel 381
MicroStrategy Desktop 587
Migration options 623
Migration tool 627
Modeler
user 237
Modeling
master data 401
Modeling objects 275
Modeling role 235
Model-view-controller (MVC) 62
Monitor for periodic process chains 268
Multidimensional Expressions
(MDX) 34, 596
Multistore table 77
Multitemperature data management 72
MYHANA source system 284

N

Named user 227
Nameserver 81, 98
Naming convention 483
NAS 610
NAT 80
gateway 81
Native DataStore object (nDSO) 416
Navigation attribute 327
nDSO
tables 417
Nearline storage 30, 73, 442
Network 80
private 80
Network Address Translation (NAT) 80
Network attached storage (NAS) 610
Network interface 80
virtual 611
Network interface (NIC) 611

Node utilization forecast 73
Nonfunctional properties 63
NoSQL 415
data management systems 413

O

Object ownership 470
Object privilege 228
OData protocol 587
OData service 586
ODP source system
ABAP CDS 281
create 278
SAP BW/4HANA 278
OLAP 63
processor 62
OLE DB for OLAP 34, 68, 596
On-demand installations 36
Online analytical processing (OLAP) 32, 59
Online transaction processing (OLTP) 59
On-premise installations 36
Open Database Connectivity (ODBC) 34
Open hub destination
create 366
manage 373
Open ODS view 29, 65, 308, 456
CompositeProvider 309
create 309
data source 309
existing source system 319
prerequisites 309
SAP HANA table 315
Open operational DataStore view
(Open ODS view) 29
Operational data provisioning
(ODP) 29, 277, 630
Operational DataStore (ODS) 29, 308
Operational delta queue (ODQ) 278
Optimized row columnar (ORC) 33
Outer joins 467

P

P view 74
Package 160
imported 229
native 229
privileges 229
Parquet 33
Partitioning 418, 437
horizontal 392
Partitioning (Cont.)
level 1 437
level 2 439
physical 392
range 439
semantic 385, 436
vertical 392
PartProvider 272
Password
rules 225
Patterns 24
PEM 99
Persistence object 29
Persistent staging area (PSA) 335
Persisting characteristics 447
Perspective 140
Physical partitioning 392
Planning applications 419
characteristic relationship 424
data slices 422
debugging 432
SAP Analysis for Microsoft Office 430
Planning functions 585
Prepare mode 638
Preprocessor server 98
Primary index 357
Primary key 357
violation 73
Privacy-Enhanced Mail (PEM) 99
Private cloud 44, 609
Privileges
administrator 229
analytical 228, 254, 485
effective 250
Procedural language 256
Process chain 392
app 268
editor 393
maintenance 270
meta 399
process type 270
scheduling 396
wizard 393
Product backlog 52
Product increment 52
Product owner 52
Profile 585
Project management
agile 601
Projection 475
Propagation layer 65
Provider
virtual 29

Pruning node view (P view) 74
PuTTY key generator 99
PuTTYgen 99
Python 93
Python Package Index 93

Q

Query 375
create 375
external SAP HANA view 472
migration 628

R

RAM 33
Range partitioning 439
Raw data 61
Read access logging (RAL) 71, 212
Read access type 447
Read permission 317
Reference tables 485
Relational database 413
Release strategy 52
Remodeling job 388
Remote conversion 602
Remote desktop protocol (RDP) 41, 140
Reporting layer 66
Reporting tools 517
Repository 470
Repository procedure 484
Repository workspace 160
Representational state transfer (REST) 587
Requirements analysis 601
Resizing 614
REST 587
authorization 233
Roadmap 645
Role
create 491
naming convention 485
Root package 232
Row-level security concept 255
Runtime object 234

S

SAN 610
Sandbox 623
SAP Analysis for Microsoft Office 34, 584, 628
planning functions 585
SAP Analysis for Microsoft Office (Cont.)
planning-enabled query 430
profile 585
SAP Analytics Cloud 45, 67, 518, 569
account 518
add-in for Microsoft Office 520, 552
analytics models 552
CORS 521
free trial 518
host rule 523
HTTP whitelist 521
import connection 539
live connection 520
reporting 552
SAP BW/4HANA connection 524
SAP HANA connection 540
tenant configuration 555
UCON 521
SAP Analytics Cloud agent 526
SDK 526
set up 534
SAP BEx Analyzer 34, 584
SAP BEx Query Designer 584
SAP BEx Web Application Designer 35
SAP BPC, version for SAP BW/4HANA 127
SAP Business ByDesign 277
SAP Business Explorer (BEx) 34
SAP Business Planning and Consolidation
(SAP BPC) 30, 127
SAP Business Suite
semantic connection 69
SAP Business Technology Platform 26
SAP Business Warehouse
Edition for SAP HANA 30
powered by SAP HANA 28
SAP BusinessObjects BI 68
SAP BusinessObjects Live Office 584
SAP BW
embedded mode 30
on SAP HANA 627
release 7.3 27
release 7.4 28
release 7.5 30
SAP BW migration cockpit 630
SAP BW modeling tools 31, 59
SAP BW query designer 34, 472
SAP BW, edition for SAP HANA 30
SAP BW/4HANA
analysis authorizations 241
architecture 58
BI clients 67
business content 69

SAP BW/4HANA (Cont.)
client ... 100
cloud ... 611
configuration ... 623
create users ... 237
developer edition ... 40
developer role ... 240
installation ... 623
license ... 111
mode ... 643
ODP ... 278
on-premise ... 602
project names ... 148
query ... 375
roadmap ... 645
sizing ... 635
technical requirements ... 602
Unicode ... 603
variable ... 377
SAP BW/4HANA instance
clone ... 623
create ... 41
SAP BW/4HANA modeling tools
installation ... 150
SAP BW/4HANA ODP data sources
replication ... 287
SAP BW/4HANA server
connection ... 146
SAP BW/4HANA starter ... 30
SAP BW/4HANA versus SAP HANA ... 416
SAP BW-managed schema ... 61
SAP Cloud Appliance Library ... 37
documentation ... 43
registration ... 38
SAP Cloud Platform Integration ... 50
SAP Data Hub ... 192
SAP Data Intelligence ... 47, 191, 193
SAP Data Provisioning Agent ... 50
SAP Data Warehouse Cloud ... 46, 176–177, 491–495, 497–503
auditing ... 499
connections ... 495
data builder ... 500
data modeling ... 491
data provisioning agent ... 185
schema ... 496
spaces ... 493
time data ... 498
SAP Development Tools for Eclipse ... 150
SAP Digital Boardroom ... 193
SAP Fiori ... 587
SAP GUI ... 101
SAP GUI for Java ... 140
SAP HANA
access control lists ... 227
administrator privileges ... 229
analysis users ... 247
analyst role ... 246
analytical authorizations ... 227
analytical privileges ... 228
application privileges ... 229
authorization concept ... 229
authorization controls ... 226
authorization trees ... 247
client ... 167, 596
data lifecycle manager ... 72
data modeling ... 469
data type ... 465
data warehousing ... 412
design-time objects ... 234
developer role ... 235
development objects ... 234
dynamic tiering ... 73, 75, 444
hardware key ... 113
information model ... 65, 483
information view ... 447
MDX provider ... 596
modeling role ... 235
object privileges ... 228
ODBC data source ... 168
package privileges ... 229
real users ... 227
repository ... 232, 470
repository privileges ... 232
roles ... 235
security console ... 225
system privileges ... 228
system replication ... 614
transport management ... 261–262
user password rules ... 225
user schema ... 228
users ... 237
SAP HANA 2.0 ... 632
SAP HANA administration console ... 141
SAP HANA authorizations
errors ... 250
SAP HANA Cloud ... 176
SAP HANA data warehousing foundation ... 392, 416
SAP HANA database user
create ... 224
SAP HANA Enterprise Cloud ... 44–45
SAP HANA extended application services, advanced model (SAP HANA XSA) ... 49

SAP HANA external views
generation ... 125
SAP HANA Marketplace ... 49
SAP HANA objects
packages ... 160
repository workspace ... 160
SAP HANA ODP
data sources ... 290
SAP HANA smart data access ... 29
SAP HANA start service
error handling ... 162
SAP HANA Studio ... 139
archive link ... 153
error handling ... 162
error log ... 166
quick fix ... 166
SAP BW modeling ... 141
source systems ... 276
transformation wizard ... 330
SAP HANA transport for ABAP ... 265
SAP HANA view ... 28, 125
external ... 125, 471, 473
SAP HANA web-based development workbench
privileges ... 231
SAP HANA XS engine ... 586
SAP HANA XS, classic model
server ... 98
SAP ID Service ... 586
SAP Identity Service ... 586
SAP IQ ... 73
SAP Landscape Management ... 619
SAP Landscape Transformation ... 277
SAP Landscape Virtualization Management (SAP Landscape Management) ... 619
SAP Leonardo ... 192
SAP Lumira ... 572
SAP Lumira, designer edition ... 36, 572–573
SAP Lumira, discovery edition ... 36, 572
SAP NetWeaver AS ABAP ... 602
SAP Note Analyzer ... 636
SAP Quick Sizer ... 612
SAP S/4HANA ... 412
embedded analytics ... 45, 193
semantic connection ... 69
users ... 414
SAP Solution Manager ... 603
SAP start service ... 98
SAP Store ... 49
SAP Vora ... 33, 47, 415
SAP Web Dispatcher ... 98
SAP Web IDE for SAP HANA ... 139
SAPUI5 ... 587
Scalability ... 58
Scalable architecture ... 61
Scale-down procedure ... 611
Scale-out architecture ... 33, 415, 611
tools ... 416
Scale-out scenario ... 611
Scale-up requirement ... 611
Scaling
horizontal ... 611
vertical ... 611
Schema ... 228
SAP BW-managed ... 61
Schema-on-read ... 33, 192
Script server ... 98
Scrum ... 51
Secondary index ... 357
Security console ... 225
Security group ... 615
Semantic group ... 384
create ... 384
Semantic key ... 366
Semantic layer ... 33, 60
Semantic partitioning ... 385, 436
multidimensional ... 387
Semantically partitioned object (SPO) ... 436
Service-level agreement (SLA) ... 63
Shell conversion ... 602
SID ... 402
Single sign-on (SSO) ... 227
Single source of truth ... 28, 65
Sizing ... 635
Snapshot ... 613
Snowball appliance ... 82
Snowflake schema ... 592
Software provisioning manager (SWPM) ... 623
Software update manager (SUM) ... 627
Software-as-a-service (SaaS) ... 44
Source systems ... 276
CDS ... 281
create ... 276
file ... 284
POS ... 297
SAP HANA ... 282
type ... 278
Sprint backlog ... 52
SQL ... 34, 170
SQL Database Connectivity (SQLDBC) ... 34
SQL-based analytical privileges ... 485
SQLDBC ... 34
SQLDBC driver ... 69
SQLScript ... 60, 256
best practices ... 258
programming ... 258

SSH key 99
SSL encryption 153
SSO 227
Standard DataStore object 27
Standard DSOs 335
Star schema 592
Starter add-on
installation 638
Storage appliance 82
Storage area network (SAN) 610
Storage manager 621
Stored procedure 484
Strict mode 643
Subnet 80
mask 80
Subscription 49
SUM 627
Supply layer 60
Support package manager 115
Support packages
installation 114
Surrogate ID (SID) 27, 402
SUSE firewall settings 182
SUSE Studio 81
Switch 611
virtual 611
System
hyper-converged 609
System copy 623
System privileges 228
System replication 614

T

Table partitioning 77
Tableau 593
TCP/IP protocol 50
Technical user 224, 366
Tenant database
error handling 165
Text variables
naming convention 606
Time dimension 329
Transactions
ODQMON 278
RS2HANA_CHECK 244
RS2HANA_VIEW 127
RSA1 ..... 106, 127, 259, 267
RSB4HTRF 627
RSDMWB 32
RSPC 270
RSPCM 268
Transactions (Cont.)
RZ10 102
SAINT 638
SCTS_HTA 265
SE01 260
SE03 259
SICK 253
SLICENSE 111
SPRO 126
SRALMANAGER 212
STMS 116
SU01 ..... 104, 238
UCONCOCKPIT 171
Transfer tool 637
Transformation
create 329
Transformation finder 633
TransientProvider 28
Transport 258
mixed 265
Transport management system 116
Transport organizer 260
tools 259
Transport route 264

U

Unicode 603, 627
Unified connectivity (UCON) 171, 521
UNION 387
view (G view) 74
Unstructured data 25
Upgrade path 634
Upgrades
maintenance planner 117
User
_SYS_REPO 470
analytical user 223
analytics 227
SAP HANA database 224
SAPHANADB 317
technical 224
User administration 226
User schema 228

V

Value ID 357
Variable 377
create 377
Variables
historic 606
naming convention 605
Variant 382
vCPU 611
Vendor lock-in 610
Vertical partitioning 392
Virtual machine 41
Virtual private cloud (VPC) 81, 611
Virtual tape library 82
Virtualization 610
manager 619
solution 609
VirtualProvider 28
vLAN 611
vNIC 611
vSwitch 611

W

WAN 611
Warm data 72, 75
Whitelist 638
Wide area network (WAN) 611
Workbook
create 380
migration 628
saving 383
Write-optimized DSO 349
to aDSO mapping 349

X

X1 instance 43
XML for analysis (XMLA) 596
XML-based analytical privileges 485

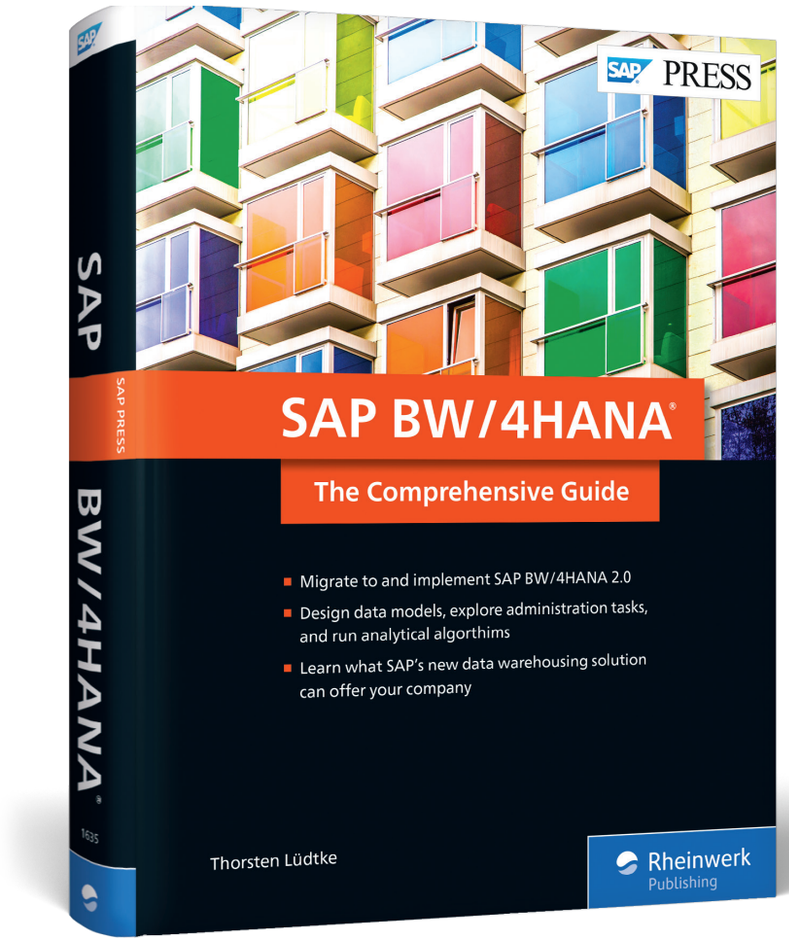
Z

Zero records 27

SAP PRESS

First-hand knowledge.

Rheinwerk Publishing



Thorsten Lüdtke, Marina Lüdtke

SAP BW/4HANA 2.0: The Comprehensive Guide

665 Pages, 2021, $89.95
ISBN 978-1-4932-1635-2

www.sap-press.com/4544

Thorsten Lüdtke is an independent SAP BI architect and founder of BIPortal GmbH (https://bi-portal.com). He specializes in data warehouse architectures based on SAP BW/4HANA, SAP HANA, and SAP Data Intelligence. Thorsten entered the analytics field in 1997 as a MicroStrategy employee, developing applications for business intelligence.

Marina Lüdtke is the managing director of BIPortal GmbH in Hamburg, Germany. Previously she worked as a sales representative for various chemical companies in Germany, East Africa, and South America. Using advanced analytics, she generated sales of over US$10 million annually. At BIPortal, she helps clients achieve the same success with BIPortal's analytical tools and SAP Analytics Cloud on SAP BW/4HANA.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.