Installing and Upgrading

Generated on: 2020-12-08 14:31:19 GMT+0000

SAP Commerce | 1905

PUBLIC

Original content: https://help.sap.com/viewer/a74589c3a81a4a95bf51d87258c0ab15/1905/en-US

Warning

Installing and Upgrading SAP Commerce

Learn everything you need to successfully install, configure and run SAP Commerce.

Get Started

Understand the basic requirements for installing or deploying SAP Commerce.

System Requirements

Licenses

SAP Commerce Directory Structure

About Patch Releases

Test and Develop

Set up and configure a local SAP Commerce instance for test and development purposes.

Download

Before You Start

Installing SAP Commerce Manually

Installing SAP Commerce Using Installer Recipes

Setting Up an Eclipse Project

Go Live

Prepare to take SAP Commerce into production.

Setting Up a Production System

Upgrading SAP Commerce

Installation Reference

Familiarize yourself with some fundamentals before starting your SAP Commerce installation. Learn about the prerequisites for installing, find out where and how to download the software, and get to grips with licenses and the directory structure.

If you're installing a patch release, it may affect your custom extensions. Find all relevant information and procedures in the topics that follow.

System Requirements

SAP Commerce is a Java application that can run on many system combinations. Understand the basic software and hardware requirements for installing SAP Commerce.

Licenses

SAP Commerce uses standard SAP license management. A valid license allows you to start SAP Commerce, and unlocks features.

Download

Download the 1905 version of SAP Commerce. Use the most recent release builds for your projects to benefit from the latest features and fixes.

SAP Commerce Directory Structure

SAP Commerce has a directory structure that makes it easy for you to work with, and separates custom files and configuration from the base features.

About Patch Releases

SAP Commerce is regularly updated and improved with bug fixes and enhancements. These updates are delivered as a complete SAP Commerce package.

System Requirements

SAP Commerce is a Java application that can run on many system combinations. Understand the basic software and hardware requirements for installing SAP Commerce.

In addition to the following listed requirements, you need a modern, standard web browser to access administration features and front ends. For any third party requirements not listed here, see Third-Party Compatibility.

Test, Demonstration, and Development Requirements

Application Server

The following requirements apply to the SAP Commerce software only, excluding third- party applications or databases. They assume a basic scenario including the SAP Commerce product package, Eclipse, and the embedded Tomcat web server.

The **Minimum** requirements should be sufficient for demonstrating the software in sales and presale presentations. For development, fulfill the system requirements as listed in the **Recommended** column:

Demonstration/Development	Minimum	Recommended
CPU	Dual Core, for example i5	Quad Core, for example i7
RAM [2]	8 GB	16 GB
Hard Disk [3]	7200 rpm IDE	SSD [2]
Hard Disk Space	10 GB	20 GB

Database

You do not need to install a separate database if you just want to try out and demonstrate SAP Commerce. It comes bundled and preconfigured with the lightweight HSQLDB database, which typically is sufficient for primary tests. This database is installed and set up by default without the need for further configuration.

Production Requirements

Application Server

For production systems, SAP recommends a multi-node system with at least one dedicated backoffice application server, several application servers, and a solr search server. The exact configuration depends on the load you expect on your system. For detailed recommendations, please contact your SAP representative. The following are minimum requirements for a single application node.

Production	Small Servers	Large Servers [2]
CPU	Dual Core, for example i5	multiple Quad Core CPUs
RAM	4 GB	16 GB or more
Hard Disk [1]	RAID 7200 rpm SAS/SCSI [1]	RAID 1/5 10000 rpm+ SAS/SCSI [1]
Hard Disk Space [3]	40 GB	>40 GB
Network [4]	100 M Bit Ethernet [4]	One or two 1 GB Ethernet adapters [4]

[1] SAP highly recommends a proven backup system. You typically do not need high performance hard disks because all IO bound data is either placed on the web server or in the database.

[2] The recommended setup for production systems depends on the load you expect on the system. SAP Commerce implementations range from one single core to a scaled cluster with 10 or more nodes and modern quad core machines. Consult your SAP representative for more information on correct sizing.

[3] You should have enough hard disk capacity to hold all log files. SAP does not recommend storing media, image, or article data on the application server tier.

[4] If using a clustered system, consider having separate Ethernet adapters between the application servers to route cache invalidation UDP multicast packets.

See also Clustered Environment or Caching.

Database

SAP recommends that you use an enterprise database system such as SAP Hana, Oracle, MySQL, or Microsoft SQL Server. For more information, see Third-Party Compatibility.

Related Information

Infrastructure Considerations for On-Prem SAP Commerce

Third-Party Compatibility

SAP Commerce is compatible with third-party software and platforms. Review the supported third-party software versions before installing SAP Commerce in your environment.

Java

Software / Platform	Supported Versions
JRE / JDK	SapMachine 11.0 (required)

Application Servers

Software / Platform	Supported Versions	Notes
Apache Tomcat	8.5.57, bundled	For Tomcat compatibility with Data Hub, see Install the Basic Prerequisites.

Operating Systems

Software / Platform	Supported Versions
Microsoft Windows	Any version that supports SapMachine 11.
Linux	Any version that supports SapMachine 11.
mac OS	⚠ For development purposes only.

Databases

Software / Platform	Supported Versions	Notes

Software / Platform	Supported Versions	Notes
MySQL (Single Node, Active/Passive)	5.6, 5.7, 8.0	The driver (MySQL Connector): • For the 8.x databases, use the latest 8.x driver, for example 8.0.15. • For the 5.x databases, the 5.x drivers (at least 5.1.x) are recommended due to some known problems occuring if you use the 8.x driver with the 5.x databases.
Percona XtraDB Cluster	5.6, 5.7	
SAP HANA	HANA 1.00 SPS12, HANA 2.0	HANA 1.00 for SAP Commerce Cloud on SAP Infrastructure only.
Oracle (Single Node, Active/Passive, Active/Active)	11.2, 12.1, 12c Release 2 (12.2), 19c (patch release 1905.11 and later)	Oracle RAC is the only supported Active/Active setup for Oracle DB. The driver: • For the 11.2, 12.1, and 12c Release 2 (12.2) databases, use the ojdbc6-11.2.0.3 driver. • For the 19c database, use the ojdbc8-19.3.0.0 driver.
Microsoft SQL Server (Single Node, Active/Passive)	2014, 2016, 2017	
Microsoft Azure SQL	12	
HSQLDB (Single Node)	2.3.4	For development purposes only.
Amazon Aurora on AWS	Certification by Amazon AWS pending.	For more information, see https://aws.amazon.com/rds/aurora/ .

Search Engines

Category	Software / Platform	Supported Versions
Search Engines	Solr	7.7

Web Browsers

The following browsers are supported by the backend and admin user interfaces.

Software / Platform	Supported Versions	Notes
Microsoft Internet Explorer	10, 11	Backoffice and Cockpits only
Microsoft Edge	Evergreen	
Mozilla Firefox	Evergreen	
Google Chrome	Evergreen	
Apple Safari	Evergreen	

CPU Architectures

SAP Commerce doesn't support the following CPU architectures:

- ppc64 (including ppc64le)
- x86_32

Others

Software / Platform	Supported Versions
ImageMagick	6.7.3-5 and higher

Licenses

SAP Commerce uses standard SAP license management. A valid license allows you to start SAP Commerce, and unlocks features.

You must have a valid license to use SAP Commerce, either locally or in production. When you first install any SAP license for SAP Commerce, the system creates a new file in `<HYBRIS_HOME>/config/license`, named `installedSaplicenses.properties`. This file contains all the required information related to your active licenses.

> ⚠ **Caution**
> Do not edit `installedSaplicenses.properties`. Editing this file may corrupt license information and lead to a non-working system.

SAP provides the following types of licenses:

- A temporary license for test and development purposes
- A regular license for use in a production environment

Temporary License

The temporary license is available for test and demonstration purposes. An initial 30-day temporary license is applied during the installation process.

If you need to extend the temporary license period, you can use the command-line license tool to generate a further 90-day license once the initial 30-day license expires. This gives you a maximum of 120 days (30 + 90), after which you must obtain a regular license. For more information, see Command Line License Tool.

Regular License

When installing an on-premise instance of SAP Commerce, your license key is provided as a file. Install this license using the command-line tool. For more information on the tool and how to use it, see the related links. Use the SAP support portal to obtain a permanent license key.

Related Information

Command Line License Tool
Internal License Key Generation Tool
License Key Help
SAP Note 2137024

Command Line License Tool

SAP Commerce includes a simple shell script to allow you to manage your licenses from the command line.

The command line license tool allows you to install either regular or temporary licenses, delete licenses, or display information about installed licenses. The tool is provided in the form of two scripts:

- **license.sh** for Unix-based systems
- **license.bat** for Windows systems

Both `license.sh` and `license.bat` scripts are located in *<HYBRIS_HOME>*`/bin/platform/`, where *<HYBRIS_HOME>* is a root directory of SAP Commerce. Run the scripts directly from this location.

The tool provides several options you can use to manage your licenses. The following examples are based on `license.sh` for Unix. You can use the same options with `license.bat` for Windows.

Display Tool Help

To display all available options, run the script either without any options, or with the `-help` option.

```
$ ./license.sh -help
                                SAP license key administration toolkit for Java environment
                                Usage: license.sh <Option(s)>
                                Options:
                                -help  . . . . . . . . . . . . . . . . . . . . Display this help
                                -version . . . . . . . . . . . . . . . . . . . Display the tool version
                                -get . . . . . . . . . . . . . . . . . . . . . Display infos about the s
                                -number  . . . . . . . . . . . . . . . . . . . Same as -get
                                -install <filename>  . . . . . . . . . . . . . Install license keys from
                                -show  . . . . . . . . . . . . . . . . . . . . Display license keys
                                -temp <Product>  . . . . . . . . . . . . . . . Create a temporary licens
                                -temppossible <Product>  . . . . . . . . . . . Check whether a temporary
                                -delete <System Id> <Hardware Key> <Product> Delete license key(s)

                                For each of the parameters after "-delete" you can use "*" as a wildca
                                Remember to use quotes to mask the "*"!
```

You can also use `-h` as a shortcut.

Install a Temporary License

To install a new temporary license for HSQLDB, use the `-temp` option.

```
$ ./license.sh -temp CPS_SQL
                                First temporary license key installed.
```

For the correct database code to use for other supported databases, see SAP Commerce License Attributes.

⚠ Caution

You can install only one temporary license per product.

Install a Permanent License From a File

To install a license from a file, use the `-install` option, and specify the file path.

```
$ ./license.sh -install /path/to/fileContainingLicense.txt
                                1 SAP license key(s) successfully installed.
```

You can also use `-i` as a shortcut.

Display Information About Installed Licenses

To display information about installed licenses, use the `-show` option.

```
$ ./license.sh -show
                                System. . . . . : CPS
                                Hardware Key. . : Y4989890650
                                SW product. . . : CPS_SQL
                                SW product Limit: Unlimited
                                Begin date. . . : 10/May/2016
                                Expiration date : 08/Aug/2016
                                License key type: Temporary
                                Installation No.: 0000000000
                                System No.. . . : 0000000000000000000
                                Validity. . . . : valid
```

You can also use `-s` as a shortcut.

SAP Commerce License Attributes

A valid SAP license for SAP Commerce contains some predefined attributes. Choose the correct product name code for your database when generating a license using the command line tool.

The following attributes are generated with any SAP Commerce license.

key	value	comment
HARDWARE-KEY	Y4989890650	Unlike other SAP products, an SAP Commerce license is not issued for a specific hardware setup. Instead, the hardware key component is filled with this **preset** value.
SWPRODUCTNAME	CPS_*<database code>*	SAP Commerce has the product code CPS, which is combined with the **database** that the license has been issued for. See the following table for the list of available values.

Provide an appropriate three-letter database code to complete the SWPRODUCTNAME attribute when generating a temporary license.

Database	Value
SAP Hana	CPS_**HDB**
Microsoft SQL Server (Single Node, Active/Passive)	CPS_**MSS**
Oracle (Single Node, Active/Passive, Active/Active)	CPS_**ORA**
MySQL (Single Node, Active/Passive)	CPS_**MYS**
Percona XtraDB Cluster	CPS_**MYS**
HSQLDB (Single Node)	CPS_**SQL**

When you create your license, choose a three-letter system ID (SAPSYSTEM), which you then **must** add to the platform configuration. You can define this in your`local.properties` file as follows:

```
# System ID
        license.sap.sapsystem=XYZ
```

If omitted, the platform uses CPS as the default value.

Download

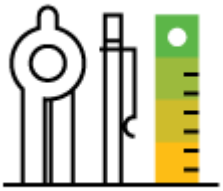
Download the 1905 version of SAP Commerce. Use the most recent release builds for your projects to benefit from the latest features and fixes.

SAP Commerce is protected by a license that allows you to start it up and unlock features; you must install a license to use it. The software uses standard SAP license management. There are two types of licenses: an initial, temporary license valid for 30 days, which you can extend for a further 90 days, and a permanent commercial license. For further instructions on installing license keys, see Licenses.

Customers

Download SAP Commerce software releases and patches from the **SAP Software Download Center** or **SAP ONE Support Launchpad**. Search for “SAP Commerce” in Downloads using the search field provided. In case of problems, get the information and support you need using the SAP Support Portal - **My Support**.

Partners



For production purposes, use your customer's S-user ID to download SAP Commerce from the **SAP Software Download Center**. If you need a customer S-user ID, get in touch with your customer. For demonstration and development purposes, use your own S-user ID.

Resources

SAP Commerce Downloads

Patch Releases

My Support

Users & Authorizations

The following ZIP files are available for download with version 1905. To find them, go to the SAP ONE Support Launchpad and search for package name + version in **Downloads**. For example, “SAP COMMERCE 1905” (without quotes).

Product	Version	Download Package Name	Description

Product	Version	Download Package Name	Description
SAP Commerce	1905	SAP COMMERCE	Download this to set up a local development or on-premise production installation of SAP Commerce. Includes the following ZIP files, which can be downloaded separately: • The main ZIP file, CX COMMERCE, containing the SAP Commerce core functionality. • The Data Hub ZIP file, CX DATAHUB. To add this product to SAP Commerce, download the ZIP, and copy it into your expanded HYBRISCOMM directory. From there, expand the new ZIP file using the unzip -u command. The -u option updates the existing SAP Commerce file hierarchy, adding the Data Hub directories at the correct location. **Note** The SAP COMMERCE package is for on-premise SAP Commerce only.
SAP Commerce, localization for China	3.0	SAP CHINA COMMERCE LOC	SAP Commerce, localization for China enhances the omnichannel capabilities of SAP Commerce, accelerator for China with marketplace integration and social media integration.
SAP Commerce Billing Integration	2.8	SAP HYBRIS BILLING INTEG	SAP Commerce Billing with SAP Commerce enables you to sell subscription and usage-based services through SAP Commerce as an omni-channel solution.
SAP Commerce CRM Integration	2.8	HYBRIS CRM INTEG	The integration of SAP CRM back-end with SAP Commerce allows customers to leverage the power of SAP CRM to enhance the capabilities of the industry-leading commerce solution from SAP Commerce, and thus reap the benefits of both worlds.

Product	Version	Download Package Name	Description
SAP Commerce Integration Package for SAP for Retail	2.7	SAP INDUSTRY PACK. RETAIL	Retail-specific features are provided by the SAP Commerce, integration package for SAP for Retail.
SAP Commerce Telco & Utilities Accelerator	1907	SAP INDUSTRY PACK. TELCOMEDIA	The Accelerator is a ready-to-use web framework enabling Service Providers to sell more online and offline in an intuitive and cost-effective way.
SAP Commerce Telco & Utilities Accelerator	1911	SAP INDUSTRY PACK. TELCOMEDIA	The Accelerator is a ready-to-use web framework enabling Service Providers to sell more online and offline in an intuitive and cost-effective way.
SAP Commerce Telco & Utilities Accelerator	2003	SAP INDUSTRY PACK. TELCOMEDIA	The Accelerator is a ready-to-use web framework enabling Service Providers to sell more online and offline in an intuitive and cost-effective way.
SAP Commerce Financial Services Accelerator	1907	SAP INDUSTRY PACK. FINANCIAL	Financial Services Accelerator is a tailor-made, omnichannel solution for direct insurance companies, banks, and other financial service providers that are looking to develop a more streamlined, seamless, and personalized customer experience, while meeting the unique needs of their industry. Financial Services Accelerator comes with preconfigured reference integrations and storefront templates for insurance and banking industries.
SAP Commerce Financial Services Accelerator	2002	SAP INDUSTRY PACK. FINANCIAL	Financial Services Accelerator is a tailor-made, omnichannel solution for direct insurance companies, banks, and other financial service providers that are looking to develop a more streamlined, seamless, and personalized customer experience, while meeting the unique needs of their industry. Financial Services Accelerator comes with preconfigured reference integrations and storefront templates for insurance and banking industries.
SAP Commerce Travel Accelerator	5.0	SAP INDUSTRY PACK. TRAVEL	Deliver Omni-channel Digital Traveller Engagement with SAP Commerce.

SAP Commerce Directory Structure

SAP Commerce has a directory structure that makes it easy for you to work with, and separates custom files and configuration from the base features.

The `<HYBRIS_HOME_DIR>/hybris` folder contains the following directories:

- `bin`: The platform and features, within the following subdirectories:
  - `platform`: The core SAP Commerce platform extensions that are loaded on startup. This directory also has the build framework, and extension templates.
  - `modules`: The extensions, organized by module. Each module provides a related set of business functionality.
  - `custom`: Any custom extensions or overrides that you may develop. This directory is created when you first set up a custom extension using the `extgen` template.
- `config`: Configuration files and license key, including properties files and `localextensions.xml`, which defines the included extensions.
- `data`: Data files such as media files and LucerneSearch indexes. The directory also contains HSQLDB data files, if you are using the default database in a test or development environment.
- `log`: Log files, including Tomcat and JDBC logs.
- `temp`: Any temporary files.

Technical Aspects

- The `bin` directories do not contain any customizable configuration data. Separating the binaries from those files you may wish to retain from one version to the next provides an easy upgrade path. The upgrade process does not replace any custom configuration files.

  > ⚠ **Caution**
  >
  > Never change anything within the `bin` directory. Do not keep any custom data configuration in this directory or any of the subdirectories. The upgrade process may replace the `bin` directory and subdirectories with a newer version of SAP Commerce and remove your changes.

  For a list of the modules and extensions in the `bin` directory, see Modules and Extensions and AddOns.
- Custom configuration data such as the license, `local.properties`, and `localextensions.xml` files must reside in the `/config` directory.
- If no `config` directory is available when building SAP Commerce, the system prompts you to choose a configuration template: develop or production. For further details, see Configuration Templates.
- The `config` directory for development is an Eclipse project. Add this as a separate project. For more information, see Setting Up an Eclipse Project.

About Patch Releases

SAP Commerce is regularly updated and improved with bug fixes and enhancements. These updates are delivered as a complete SAP Commerce package.

A patch release is a **complete SAP Commerce** solution and should not be confused with a hot fix or a separate patch for a specific module. It is delivered as a whole in order to guarantee that it is fully tested. This avoids having unlimited combinations of patched modules. The latest patch release always includes all the corrections of previous patches. For this reason, we only provide the latest version as we want to be sure that you benefit from all available corrections.

Installing a patch release is an easy update that does not compare to an upgrade to a new release. Unlike an upgrade to a new release, there are no upgrade steps that need to be done.

Note

SAP Commerce Accelerator is released as source and are **implementation templates**. Due to the template character of Accelerators and their need for intense customization, you cannot apply patches to customized Accelerator storefront templates. Therefore, we do not offer backports of bug fixes for Accelerator storefront templates for prior releases of Accelerator. You must maintain your Accelerator storefront template implementations on your own once you customize the template.

Maintenance Guarantee

An update means that SAP guarantees the following:

- API Compatibility
- Data Model Integrity
- Database Integrity
- No new Functionality

To ensure compliance with these guarantees, each patch release undergoes the following tests:

- Initializations with basic testing for different recipes
- Approximately 150,000 integration and compatibility tests running for every recipe
- Final comparison and smoke testing
- All testing based on the same procedure as for major or feature releases

Related Information

Upgrading SAP Commerce
Updating SmartEdit When Installing an SAP Commerce Patch

Updating a Local Installation

Install an SAP Commerce patch release into a local development environment.

Procedure

1. Create the folder where you want to extract the SAP Commerce ZIP file.
2. Download the latest patch release version of SAP Commerce 1905.

   For more information on available patch releases, see Patch Releases.
3. Extract the ZIP file into the newly created folder.
4. Back up your custom extensions, in case they are located in `<HYBRIS_HOME>/hybris/bin`.
5. Delete the `<HYBRIS_HOME>/hybris/bin` directory of your existing installation.
6. Copy the `<HYBRIS_HOME>/hybris/bin` directory from the patch release to your existing `<HYBRIS_HOME>/hybris` directory.
7. Copy your custom extensions back to `<HYBRIS_HOME>/hybris /bin`.
8. Build SAP Commerce.

Related Information

Download

Installing a Local Instance

Create a local installation of SAP Commerce for testing, development, or demonstration. Configure the system manually according to your specific needs, or use one of the provided installation recipes to use a pre-configured setup. If necessary, integrate your environment into your Eclipse IDE.

Before you begin any installation, ensure that you are familiar with the prerequisites and licensing.

Before You Start

To install and run SAP Commerce in a local environment, you require a compatible Java SDK. Ensure that the correct SDK for your SAP Commerce version is installed and configured correctly, then download and unpack the SAP Commerce ZIP before moving on to the installation steps.

Installing SAP Commerce Manually

Follow the basic steps for setting up an SAP Commerce test and development environment.

Installing SAP Commerce Using Installer Recipes

To automate your installation of SAP Commerce for development and demonstration purposes, you can use an installer recipe. Recipes are gradle scripts that take care of creating directories, moving files, updating configuration and properties files, and initializing the system.

Setting Up an Eclipse Project

Preconfigured Eclipse `.project` and `.classpath` files allow you to import SAP Commerce and its packages into Eclipse, ready for custom development.

Related Information

System Requirements
Licenses

Before You Start

To install and run SAP Commerce in a local environment, you require a compatible Java SDK. Ensure that the correct SDK for your SAP Commerce version is installed and configured correctly, then download and unpack the SAP Commerce ZIP before moving on to the installation steps.

Install the Java SDK

SAP Commerce requires a compatible Java SDK to install and run the application.

Context

SAP Commerce 1905 requires JDK 11 or later. It is fully compatible with SAP Machine 11.

Procedure

Follow the instructions on the SDK source web site to download and install the correct version for your operating system.

Related Information

SAPMachine

Download and Unpack SAP Commerce

SAP Commerce releases are available as ZIP archive files. Download and unpack the required files before you begin installation.

Prerequisites

Ensure you are familiar with the system requirements. For more information, see System Requirements.

Context

ℹ Note

There are multiple installation ZIP files:

- `CXCOMM190500P_X-XXXXXXXX.zip`: Contains all the files to install and run SAP Commerce.
- `CXDATAHUB190500P_X-XXXXXXXX.zip`: Contains the files to install SAP Commerce Data Hub.

Data Hub is distributed separately. To add this product to SAP Commerce, download the ZIP, and copy it into your expanded `CXCOMM190500P_X-XXXXXXXX` directory. From there, expand the new ZIP file using the `unzip -u` command. The `-u` option updates the existing SAP Commerce file hierarchy, adding theData Hub extension directories in the correct location.

Select only the ZIP archives you want to install. You don't need to download all the ZIP archives listed. To download and unpack the SAP Commerce ZIPs, perform the following steps.

Procedure

1. Download the packages you want to install.

   For more details, see Download.

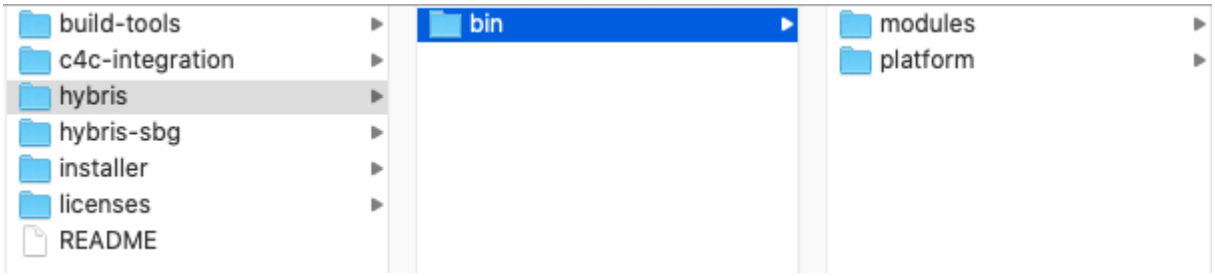
2. Create the directory where you want to extract the ZIP archives.

   Follow these best practice guidelines when creating the installation directory:

   - Ensure that the installation directory is close to the system root directory, for example `C:\hybris`. You may download the files directly to the root directory, because the ZIP files already contain a `hybris` directory. Microsoft Windows requires directory paths to be shorter than 255 characters. Any part of a directory path exceeding this limitation is truncated. The SAP Commerce modules installation creates several sub-directories, so use short directory paths.
   - Avoid directory paths containing spaces such as `C:\hybris Platform`. Building SAP Commerce fails if the directory path contains spaces.

3. Extract the ZIP archives into the directory you just created.

   Upon completion, the chosen directory should contain the structure similar to the following example:



⚠ Caution

When extracting a ZIP file on Windows, you may encounter an error due to path length. You may receive an error message similar to "Cannot create <pathname> total path and file name length must not exceed 260 characters". If so,

you can:

- Use an alternative extraction tool, one that doesn't limit the path or name size.
- `jar -xvf <zipfilename>`.

Do **not** use the JAR utility to unzip the ZIP file on Unix-based systems, as this will change the timestamps of the contained files and break the build process.

Installing SAP Commerce Manually

Follow the basic steps for setting up an SAP Commerce test and development environment.

These basic steps describe how to install and initialize SAP Commerce with the provided SAP Commerce Server. The server is a preconfigured application server based on Apache Tomcat. Both HSQLDB and Tomcat are included in the SAP Commerce package, and are designed to be used for testing, development, and demonstration purposes. This guide also provides optional steps that describe how to use different databases also supported by SAP Commerce.

Install and configure only the packages that are needed in your project. Any initial data provided with SAP Commerce is intended for demonstration purposes only.

Installer recipes included with SAP Commerce allow you to install a preconfigured environment quickly using the default HSQLDB on your local machine. The following procedures cover installing SAP Commerce using manual configuration. For information on installing using recipes, see Installing SAP Commerce Using Installer Recipes.

Set Up Your Build

After you have successfully downloaded and unpacked the required files, you must then create your SAP Commerce project directories before proceeding.

Configure the Functionality

Define what functionality you require by specifying the required extensions, and performing other configuration tasks.

Initialize SAP Commerce

Initialize your SAP Commerce instance to generate the database schema and the type system, and import any essential system data. Before initialization, the system is not fully functional.

Start SAP Commerce

SAP Commerce is shipped with an embedded Apache Tomcat server instance. This embedded application server makes it easy to deploy SAP Commerce in a development and test environment.

Install Data Hub Locally

Set up a local instance of Data Hub alongside your SAP Commerce installation to test integration scenarios that require SAP Commerce Data Hub.

Set Up Your Build

After you have successfully downloaded and unpacked the required files, you must then create your SAP Commerce project directories before proceeding.

To set up SAP Commerce, carry out the following steps.

1. Set up Apache Ant
2. Start a clean build to create your project directories

Related Information

Building SAP Commerce
Installation Based on Specified Extensions

Setting up Apache Ant

SAP Commerce comes pre-bundled with Apache Ant, a tool for automating the software build processes. Configure Apache Ant before you build SAP Commerce.

Context

SAP Commerce provides three script files that you can use to set your Ant environment for the current command console session:

- **`setantenv.bat`**: a batch file for Microsoft Windows systems
- **`setantenv.ps1`**: a shell script file for Windows Powershell
- **`setantenv.sh`**: a shell script file for Unix-based systems such as Linux or Mac OS X

Procedure

1. Open a command prompt in Microsoft Windows, or a shell in the Unix family
2. Navigate to `hybris/bin/platform` in your installation.

   For example:

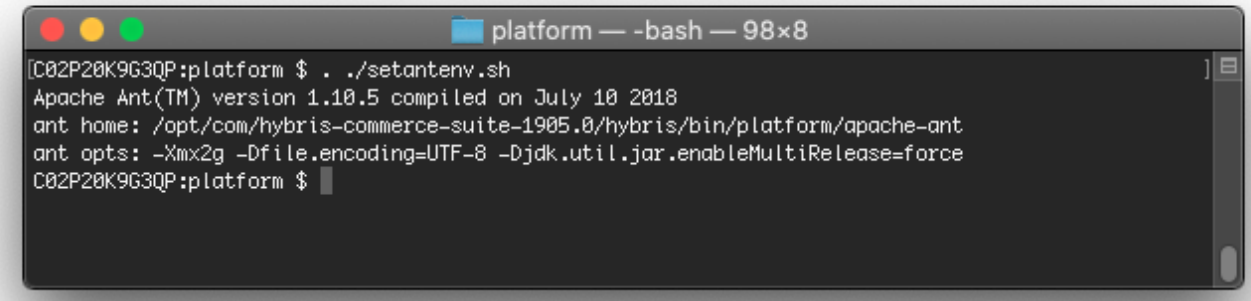
   ```
   C:\hybris\bin\platform
   ```

3. Run the provided script for your platform.
   - For Windows systems, enter `setantenv.bat`
   - For Windows Powershell, enter `.\setantenv.ps1`
   - For Unix systems, including Mac OSX, enter `. ./setantenv.sh`

   Do not close the command window, as you need it for the next steps of the installation procedure. Doing so may require you to set the Ant environment once again.

Results

You should see something similar to the following:



Creating the Project Directories

Once you have set up your Ant environment, use Ant to create your SAP Commerce project framework.

Context

Running `ant clean all` on a new instance of SAP Commerce creates any configuration files and other essential directories. This is a necessary step in the installation process, and must be done once before you proceed with the configuration.

If a build already exists, it deletes and recreates it. Repeat this command any time you wish to create a new build in place of an old one.

Procedure

1. At the command prompt, navigate to `/hybris/bin/platform` and enter the following command.

   ```
   ant clean all
   ```

   The system prompts you to choose a configuration template for your environment. You can choose from either development, or production.

   ```
   Buildfile: C:\hybris\bin\platform\build.xml
        [echo]
       [mkdir] Created dir: C:\hybris\log
       [mkdir] Created dir: C:\hybris\data
       [mkdir] Created dir: C:\hybris\temp\hybris
       [input]
       [input]  **** NO CONFIG FOLDER FOUND **** use the jar utility to
                       unzip the zip file on unix based systems, as this will change the timestamps
                       the contained files and break the build process.Buildfile: C:\hybris\bin\pla
        [echo]
       [mkdir] Created dir: C:\hybris\log
       [mkdir] Created dir: C:\hybris\data
       [mkdir] Created dir: C:\hybris\temp\hybris
       [input]
       [input]
       [input]  No config folder was found at C:\hybris-4.5\hybris\config.
       [input]  A "fresh" folder containing basic configuration files and the hybris
       [input]  demo licence will be created for your convenience.
       [input]  Please adjust and review the configuration files (and license) and
       [input]  call 'ant' again. This directory will never be overridden or
       [input]  touched again. Always use this configuration folder for configuration
       [input]  of platform, do not change anything within the platform folder.
       [input]
       [input]  Please choose the configuration template.
       [input]  Press [Enter]
       [input to use the default value ([develop], production)
   ```

2. Select the appropriate configuration template.
   - To choose the default develop template, press Enter.
   - To choose the production template, type `production`, then press Enter.

   Always choose the **develop** template for test, development, and demonstration purposes.

   When the build process is complete, you should see output similar to the following:

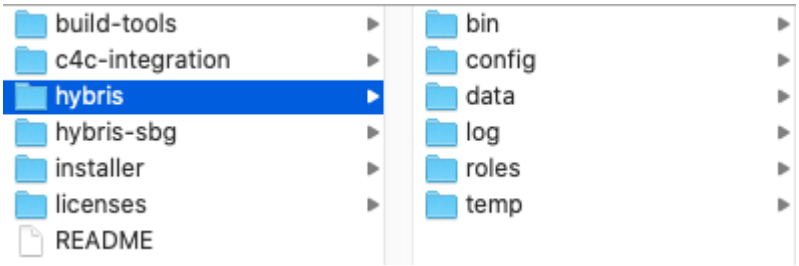
   ```
   server:
        [echo]
        [echo] Configuring server at C:\hybris\bin\platform/tomcat-6
        [echo] Using config set at C:\hybris\config/tomcat
        [echo]
       [mkdir] Created dir: C:\hybris\log\tomcat
       [mkdir] Created dir: C:\hybris\data\media
        [copy] Copying 8 files to C:\hybris\bin\platform\tomcat-6
        [copy] Copying 6 files to C:\hybris\bin\platform\tomcat-6
        [copy] Copying 1 file to C:\hybris\bin\platform\tomcat-6\lib
        [echo]
        [echo] Embedded server does not seem to be running (no PID found). No restart necessary.
        [echo]
   ```

```
     all:
          [echo] Build finished on 26-March-2013 11:23:33.
          [echo]

     BUILD SUCCESSFUL
     Total time: 3 minutes 15 seconds
     C:\hybris\bin\platform>
```

Results

Your `hybris` directory now contains the following:



Configure the Functionality

Define what functionality you require by specifying the required extensions, and performing other configuration tasks.

SAP Commerce is composed of component extensions. Each extension offers specific additional functionality, and may be dependent on other extensions to provide a module of features such as the Customer Service Module, or SmartEdit. You can also include your own custom extensions.

Setting the Administrator Password Manually

For security reasons, SAP Commerce comes with no default password for the Administrator user. Set this password in the `local.properties` file before you build or initialize the system for the first time.

Context

Carry out these steps only if you are installing SAP Commerce manually. For the correct procedure when you are installing using a recipe, see Setting the Administrator Password with a Recipe.

Procedure

1. Navigate to `hybris/config` and open the `local.properties` file for editing
2. Add the Administrator password with the `initialpassword.admin` property. or the `y_initialpassword_admin` environment variable.

   ```
   initialpassword.admin=<yourSecurePass>
   ```

3. Save the file.

Next Steps

When you initialize the system for the first time, the password is loaded into the database. You can change it later using the Backoffice Administration Cockpit. Delete the password from `local.properties` after you initialize the system to protect it against unauthorized access.

Adding Required Extensions

Define which extensions you require by adding them to the `localextensions.xml` file. You can add either existing extensions that ship with SAP Commerce, or your own custom extensions.

Context

Only the core Platform extensions load by default. Any other functionality you require must be configured explicitly by including the desired extensions in `localextensions.xml`. You must always rebuild the installation whenever you change this file.

Procedure

1. Go to `/hybris/config` and open the `localextensions.xml` file for editing.
2. Add the list of extensions you wish to include within the `<extensions>` tag, similar to the following, then save the file.

```
<hybrisconfig xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xsi:noNamespaceSchemaLocatio
                                                <extensions>
                                                <path dir='${HYBRIS_BIN_DIR}' autoload='false' /
                                                <extension name='adaptivesearchsolr' />
                                                <extension name='adaptivesearchbackoffice' />
                                                <extension name='adaptivesearchsamplesaddon' />
                                                <extension name='adaptivesearchwebservices' />
                                                <extension name='commerceservicesbackoffice' />
                                                <extension name='solrfacetsearchbackoffice' />
                                                <extension name='solrserver' />
                                                <extension name='yacceleratorbackoffice' />
                                                <extension name='yacceleratorinitialdata' />
                                                <extension name='yacceleratorfulfilmentprocess'
                                                <extension name='yacceleratorstorefront' />
                                                <extension name='ycommercewebservices' />
                                                <extension name='ycommercewebservicestest' />
                                                <extension name='electronicsstore' />
                                                <extension name='apparelstore' />
                                                ...
                                                </extensions>
                                                </hybrisconfig>
```

3. If you wish to configure the behaviour of the extensions, add configuration parameters to the `local.properties` file, which is also in the `config` directory.
4. Build SAP Commerce now, or if you have additional configurations to perform, proceed with those before performing the build.

Providing a Custom Database Configuration

SAP Commerce is pre-bundled with HSQLDB, and is configured to use it out of the box. To use any other supported database, or to change the HSQLDB performance, further configuration is necessary.

HSQLDB is a light-weight SQL database that runs within a Java Virtual Machine. It is useful for development environments due to its speed. SAP Commerce comes pre-bundled with a compatible version of HSQLDB. For most development or test scenarios, the default setup should be sufficient, and no further configuration is necessary. If you wish to change the default database setup, see one of the following:

- To change the default setup of HSQLDB, see HSQLDB.
- To set up another supported database, see Third-Party Databases.

Initialize SAP Commerce

Initialize your SAP Commerce instance to generate the database schema and the type system, and import any essential system data. Before initialization, the system is not fully functional.

You can initialize the system from the command line using Apache Ant. Do this before starting SAP Commerce. Alternatively, you can start SAP Commerce first, and then perform the initialization from the Administration Console.

Initializing From the Command Line

The Apache Ant target `ant initialize` performs a complete initialization and loads all essential data.

Context

Initializing SAP Commerce before starting the system ensures that you have a fully functional installation as soon as it is running. It also allows you to diagnose and correct any errors during the initialization process using the command line output.

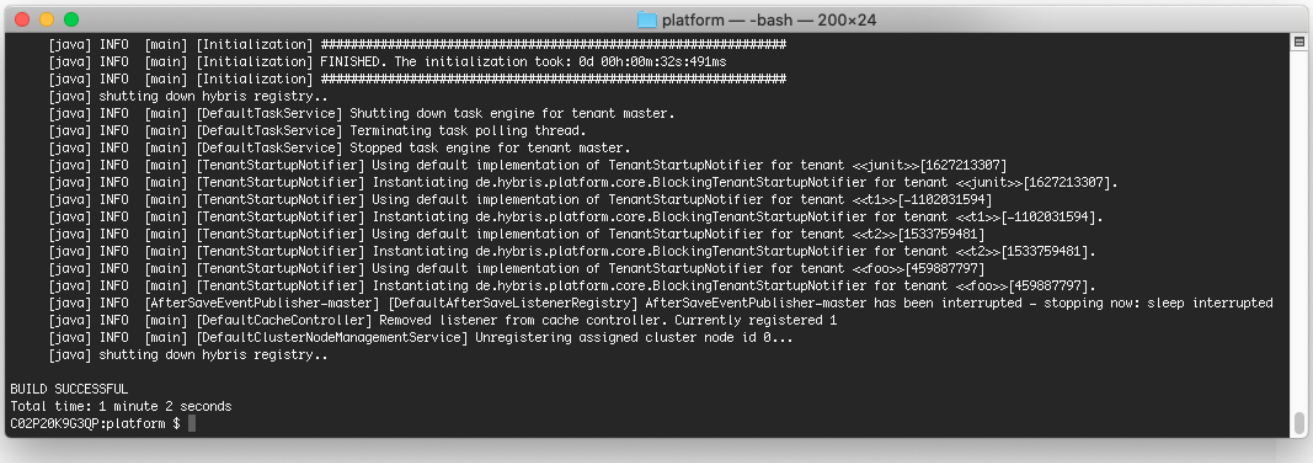
Procedure

1. Go to `hybris/bin/platform` in your SAP Commerce instance.
2. Enter the following command:

```
ant initialize
```

Results

The initialization process may take some time, depending on which extensions, and how many extensions, you have included. When completed successfully, you should see output similar to the following:



Initializing From the Administration Console

You can initialize your installation at any time from the Administration Console after starting SAP Commerce.

Prerequisites

SAP Commerce must be started and running to access the Administration Console. First complete all configuration steps, including setting an administrator password. To start SAP Commerce, see Start SAP Commerce.

Procedure

1. Log in to SAP Commerce Administration Console.

   The default address is `https://localhost:9002`.

2. Go to **Platform** **Initialization** .

   The **Initialization** page opens with preconfigured settings for initialization, and all essential data and all project data is selected. If you haven't previously initialized the system, then the **Initialization** page is the only page available.

3. Click **Initialize** to begin the initialization process.

   > ⚠ **Caution**
   > The first action of initialization is to remove all tables from the SAP Commerce database. All data is deleted in the process. For details on initialization and system update, see Initializing and Updating SAP Commerce.

   When initialization is complete, you should see a display like the following:

   ```
   Creating project data for cscockpit ...
   Creating project data for importcockpit ...
   Creating project data for cmscockpit ...
   Creating project data for btgcockpit ...
   Creating project data for productcockpit ...
   Creating project data for yacceleratorcockpits ...
   Creating project data for mcc ...
   Creating project data for backoffice ...
   Creating project data for commercesearchbackoffice ...
   Creating project data for admincockpit ...
   Creating project data for b2badmincockpit ...


   FINISHED. The initialization took: 0d 00h:57m:19s:598ms

   Continue...
   ```

   If your browser fails to display information about the end of the initialization process, look at the console output.

4. Click **Continue** and log in again to access all Administration Console features.

Related Information

Initializing and Updating SAP Commerce

Start SAP Commerce

SAP Commerce is shipped with an embedded Apache Tomcat server instance. This embedded application server makes it easy to deploy SAP Commerce in a development and test environment.

Context

This procedure covers how to set up SAP Commerce with the default SAP Commerce Server. To deploy to a different web application server, see your server's documentation.

The SAP Commerce Server has different modes of operation. You can choose to start up in any of the modes, depending on your needs.

Procedure

1. In your command line, go to `hybris/bin/platform`.
2. Run the appropriate startup script for your platform.

- To start in normal mode in Windows, use `hybrisserver.bat`
- To start in normal mode in Unix, use `./hybrisserver.sh`

Normal mode is the default mode. To start in other modes, add the mode name to the start command. The following modes are available:

Mode	Description
debug	Starts in debug mode. Use this mode together with the develop configuration template.
minimal	Starts with a minimal wrapper configuration.
jprofiler	Integrates with jprofiler. First add the path to your jprofiler installation directory by adding the `jprofiler.path` definition to your `local.properties` file.
version	Doesn't start the server, but displays information about the embedded Tomcat server environment.

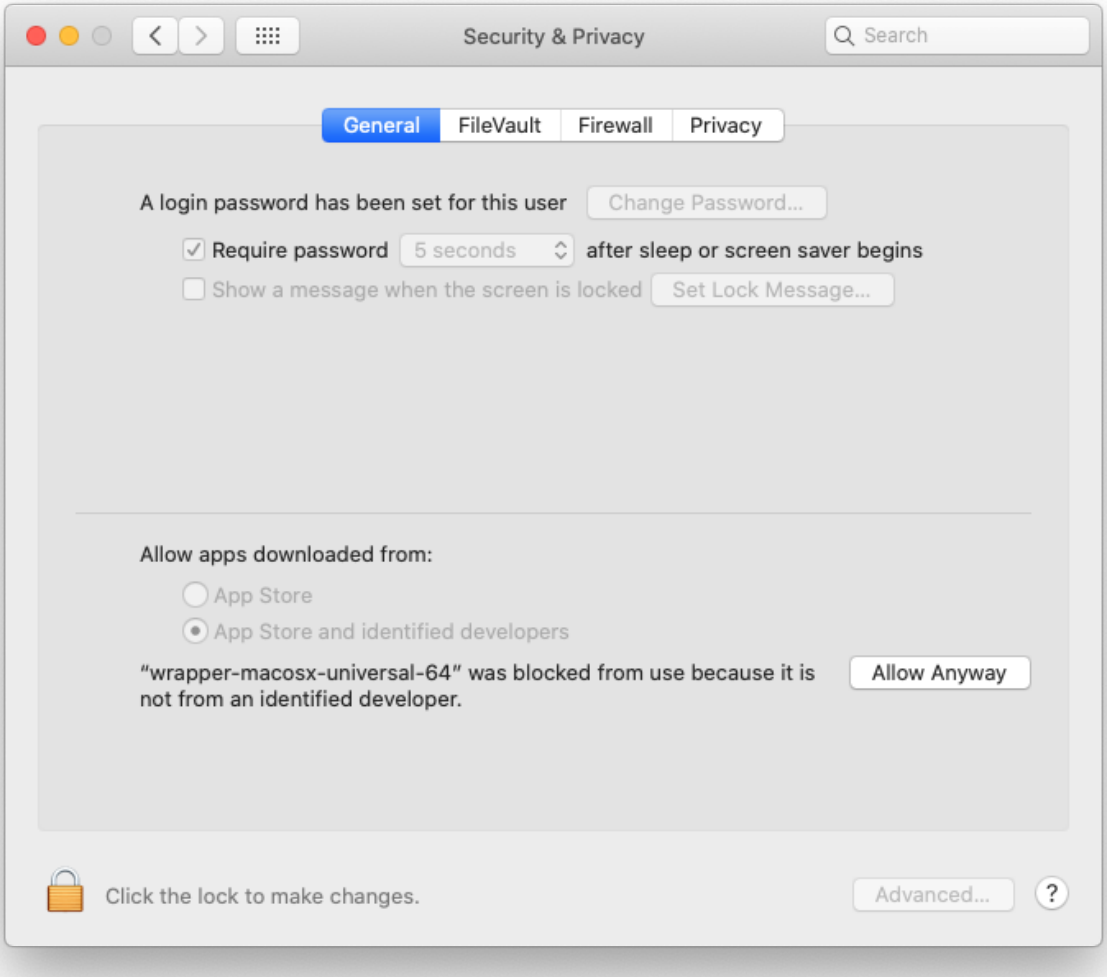
3. If necessary, allow the Apache Tomcat libraries to run in Mac OS Catalina.

   Starting with Catalina, Apple introduced stricter controls over what software could be run in the Mac OS. If you are starting SAP Commerce for the first time in Catalina or a later Mac OS, you may see a warning like the following:

   To allow the software to run, do the following:

   a. Click **Cancel**

   b. Open System Preferences and select **Security & Privacy** ▸ **General**



c. Click **Allow Anyway**, then return to the command line and run the start script again.

d. When the warning appears the next time, click the **Open** button that is now provided.

You may have to repeat this process for more than one Tomcat library. Once all affected libraries are allowed, SAP Commerce starts as normal, and you should not see these warnings again.

Results

Startup is complete when the log displays a line similar to the following:

```
INFO: Server startup in 26438 ms
```

Install Data Hub Locally

Set up a local instance of Data Hub alongside your SAP Commerce installation to test integration scenarios that require SAP Commerce Data Hub.

Install the Basic Prerequisites

Get set up to install Data Hub.

Context

Data Hub is a Java web application that uses a relational database. The following steps are required for minimal installation and, unless otherwise stated, are **not** valid for third-party software versions.

Procedure

1. Download the latest Data Hub ZIP file containing the DH web application. See the Download section of this guide for further information.
2. Install SapMachine JDK 11 from https://sap.github.io/SapMachine/ . Data Hub supports 64-bit Java.
3. Install Apache Tomcat 9.0.x from https://tomcat.apache.org/download-90.cgi .

   Follow the instructions in the Apache Tomcat documentation to install.
4. Configure Tomcat to add SSL authentication. Follow the instructions in the Tomcat SSL/TLS Configuration How-To guide.

   → Tip

   The folders you create in the next steps are required for the Hello World tutorials and for using Data Hub.

5. Create an `/opt/datahub` folder.
6. In the `datahub` folder, create the following new folders:
   a. `config`, used for the files `local.properties` and `datahub.encryption.key.txt`
   b. `extensions`
7. Create an XML file.

   → Remember

   All of the Data Hub documentation is based on the `datahub-webapp.xml` file you create in the following steps.

8. Open a new file in a text editor and copy the following into the file:

```
<?xml version='1.0' encoding='utf-8'?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application -->
<Context antiResourceLocking="true"
                docBase="{full path to your Data Hub installation}/hybris/bin/ext-integration/da
                reloadable="true">

    <Resources className="org.apache.catalina.webresources.StandardRoot" >
      <PostResources className="org.apache.catalina.webresources.DirResourceSet"
                    base="/opt/datahub/config" internalPath="/" webAppMount="/WEB-INF/classes" /
      <PostResources className="org.apache.catalina.webresources.DirResourceSet"
                    base="/opt/datahub/extensions" webAppMount="/WEB-INF/lib" />
    </Resources>

</Context>
```

9. Edit the *<docBase>* parameter to reflect the full path to your SAP Commerce installation.
10. Save the file as `datahub-webapp.xml` into the *<TOMCAT_HOME>*`/conf/Catalina/localhost` folder. **Note:** the *<TOMCAT_HOME>*`/conf/Catalina/localhost` does not exist until Tomcat is run.
11. Install the cURL command-line tool.

    From https://curl.haxx.se/ , download and install the cURL software that is appropriate for your operating system.

Related Information

Install SAP Commerce Data Hub

Install SAP Commerce Data Hub

The following steps help you install Data Hub for use with the Hello World tutorial.

Prerequisites

Before proceeding with the following installation steps, complete the steps described in Install the Basic Prerequisites.

Context

Follow this procedure to install the SAP Commerce Data Hub web application.

Procedure

1. If you do not already have the latest SAP Commerce, download the ZIP file.

   For more information, see Download.

2. Expand the ZIP file on your file system.

3. Download the separate Data Hub ZIP file from the same location, and expand it into your SAP Commerce directory.

    a. Access and download the Data Hub software from the Download Center of SAP Service Marketplace. You may have to contact your S-User (SAP Super User). https://support.sap.com/en/my-support/users.html

    b. Copy the downloaded Data Hub ZIP file into your expanded SAP Commerce directory, at the same level as the `hybris` directory.

    c. From the command line, extract the Data Hub ZIP archive with the command `unzip -u <filename.zip>`, replacing `filename.zip` with the actual name of the downloaded file.

    The `-u` option **updates** the existing `hybris` directory, and places all necessary Data Hub files in the `hybris/bin/ext-integration/datahub` subdirectory.

4. Find the Data Hub version number in the name of the WAR file located in `<HYBRIS_HOME>/hybris/bin/modules/datahub/web-app`. Update the `datahub-webapp.xml` file created in Install the Basic Prerequisites.

5. Data Hub implements HTTPS for all communications to its endpoints and with Tomcat. The Data Hub side of the HTTPS communications is on by default. However, you must add HTTPS authentication to Tomcat by reconfiguring. See https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html .

6. Go to a command line and change directory to *<TOMCAT_HOME>*`/bin`.

7. Launch Tomcat by running *<TOMCAT_HOME>*`/bin/startup.sh` from the command line. When Tomcat launches, the Data Hub WAR file is expanded.

8. Stop Tomcat by typing CTRL-C and then running *<TOMCAT_HOME>*`/bin/shutdown.sh` from the command line. With Tomcat stopped, you can further configure Data Hub.

9. From the command line, change directory to `/opt/datahub/config`.

10. With your favorite text editor, create a file named `local.properties` and save it to `/opt/datahub/config`.

11. Open the `local.properties` file. By default, the following properties are set to false.

```
datahub.cleanup.rawitem.enabled=true
datahub.cleanup.canonicalitem.enabled=true
datahub.cleanup.targetitem.enabled=true
datahub.cleanup.publisheditems.enabled=true
```

i Note

Data Hub includes a set of default local properties. For an example, see datahub-sample.gradle. For a list of available configuration properties for Data Hub, see Configuration Properties.

→ Remember

The cleanup extension is absolutely critical to maintaining a Data Hub that performs well. For additional information, see Activating Data Hub Database Cleanup.

Configure Data Hub for a Test Environment

Set up a Data Hub test environment that you can use for the Hello World tutorials.

Context

By default, Data Hub enables basic authentication on its REST API. Configure your installation with access credentials for the two basic security roles, then test your installation. After you complete these steps, you can use this environment for the Tutorial: Setting Up and Running Hello World.

Procedure

1. Create a text file called `local.properties` in the `/opt/datahub/config/` directory and add the following content:

```
datahub.security.basic.admin.user=admin
datahub.security.basic.admin.password=<password>
datahub.security.basic.read_only.user=rouser
datahub.security.basic.read_only.password=<password>
```

ℹ Note

The `datahub.security.basic.admin.user` name and the `datahub.security.basic.read_only.user` name must be different.

2. HSQLDB is the default database of SAP Commerce and, unless otherwise specified, is the default database used by Data Hub. No configuration is necessary, and you do not need to start it.
3. Start Tomcat by running `<TOMCAT_HOME>/bin/startup.sh` from the command line.
4. In a new command line window, run the following command to start Data Hub.

```
curl -G https://localhost:8443/datahub-webapp/v1/version
```

The system returns the version number when the configuration is successful.

ℹ Note

It may take a minute or two for Data Hub to start.

Installing SAP Commerce Using Installer Recipes

To automate your installation of SAP Commerce for development and demonstration purposes, you can use an installer recipe. Recipes are gradle scripts that take care of creating directories, moving files, updating configuration and properties files, and initializing the system.

ℹ Note

The installer is currently only intended to install SAP Commerce in development environments or for demonstration purposes. Do not use the installer to install SAP Commerce in a **production environment**. For instructions on installing in a production environment, see Installing SAP Commerce Manually.

With the installer, you can easily install the SAP Commerce setup you need with the desired applications, such as B2C Accelerator. You do this using a pre-defined installer recipe. Each installer recipe includes SAP Commerce Platform and all the required modules for the specific application. The installer replaces the manual procedure for configuring and setting up SAP Commerce.

⚠ Caution

Don't use the installer to run one recipe after another on the same SAP Commerce. The installer won't uninstall previous configurations, and doesn't restore your SAP Commerce file system to its original settings. To install a different recipe, use a clean environment.

Prerequisites

Prior to installing SAP Commerce, make sure that your system meets the pre-installation requirements. For more information, see System Requirements.

To review the available recipes and understand which components are included in each one, see Installer Recipes.

Installer Script Options

The following table lists the available options when running the installer script:

Option	Description	Example
`-d`	Sets the Gradle log level to debug.	`./install.sh -d <recipe>`
`-h`	Prints help to screen.	`./install.sh -h <recipe>`
`-i`	Runs the specified task for the specified recipe type.	`./install.sh -i <recipe>`
`-l`	Lists all available recipes.	`./install.sh -l <recipe>`
`-P`	Sets the platform home to the specified directory.	`./install.sh -r <recipe> -P /users/carsten/commercesuite/hybris/bin/platform`
`-r`	Runs the default setup task for the specified recipe.	`./install.sh -r <recipe>`
`-r <recipe> initialize`	Initializes SAP Commerce for the specified recipe.	`./install.sh -r <recipe> initialize`
`-r <recipe> start`	Starts the SAP Commerce Server.	`./install.sh -r <recipe> start`
`-s`	Prints out the stack trace for all exceptions.	`./install.sh -s <recipe>`
`-t`	Lists all the tasks available for the specified recipe.	`./install.sh -t <recipe>`

[Setting the Administrator Password with a Recipe]

For security reasons, SAP Commerce comes with no default password for the Administrator user. Add this password to the properties section of your installer recipe before you build or initialize the system for the first time.

[Using the Installer Recipes]

Easily install and set up an SAP Commerce instance that is tailored for your needs with a dedicated installer recipe.

[Installer Recipes]

Learn about the modules included in the available installer recipes for SAP Commerce.

[Creating Installer Recipes]

You can create your own Gradle recipe to easily install the custom SAP Commerce configuration of your choice.

Related Information

Download

Licenses

Using the Installer Recipes

Setting the Administrator Password with a Recipe

For security reasons, SAP Commerce comes with no default password for the Administrator user. Add this password to the properties section of your installer recipe before you build or initialize the system for the first time.

Context

All installer recipes are located in the `installer/recipes/` directory of your SAP Commerce instance. You can edit the `build.gradle` file of any recipe to add additional configuration properties.

→ Tip

You can also set the initial administrator password on the command line during build or initialization of the recipe, without having to edit the build.gradle file, as follows:

```
./install -r recipename -A initAdminPassword=yourpass

./install -r recipename initialize -A initAdminPassword=yourpass
```

Procedure

1. Locate the `build.gradle` file for your chosen recipe.

   For example:

   ```
   hybris-commerce-suite-1905.0/installer/recipes/b2c_acc_plus/build.gradle
   ```

2. Open the file for editing and add the following as the first item in the properties section at the head of the file:

   ```
   def pl = platform {
       localProperties {
           ...
           property 'initialpassword.admin', 'yourPass'
           ...
   ```

3. Save the file and proceed with the installation.

Next Steps

Change the password in Backoffice after the system initializes to protect it against unauthorized users.

Using the Installer Recipes

Easily install and set up an SAP Commerce instance that is tailored for your needs with a dedicated installer recipe.

Prerequisites

Before you begin, do the following:

- Install the correct Java SDK, and then download and unzip SAP Commerceas described in Before You Start.
- Set a password for the administrator user as described in Setting the Administrator Password with a Recipe.

Procedure

1. Open a command line and go to the `installer` directory of your SAP Commerce instance.
2. Build the SAP Commerce setup you need using the appropriate recipe by entering the following command:
   - On Windows: `install.bat -r <recipe_name>`
   - On Linux or Mac: `./install.sh -r <recipe_name>`

   For example, to build SAP Commerce with B2C Accelerator, enter the following command:
   - On Windows:

     ```
     install.bat -r b2c_acc_plus
     ```

   - On Linux or Mac:

```
./install.sh -r b2c_acc_plus
```

For a complete list of available installer recipes, see Installer Recipes.

> **Note**
>
> Refer to the `readme.txt` file provided in the recipe folder for specific instructions on how to install the particular SAP Commerce setup that your require. Some recipes include additional instructions to install or initialize SAP Commerce.

3. Initialize SAP Commerce by entering the following command:
   - On Windows: `install.bat -r <recipe_name> initialize`
   - On Linux or Mac: `./install.sh -r <recipe_name> initialize`

   For example, to initialize SAP Commerce with B2C Accelerator, enter the following:
   - On Windows:

     ```
     install.bat -r b2c_acc_plus initialize
     ```

   - On Linux or Mac:

     ```
     ./install.sh -r b2c_acc_plus initialize
     ```

4. Start SAP Commerce by entering the following command:
   - On Windows: `install.bat -r <recipe_name> start`
   - On Linux or Mac: `./install.sh -r <recipe_name> start`

   For example, to start SAP Commerce with B2C Accelerator, enter the following:
   - On Windows:

     ```
     install.bat -r b2c_acc_plus start
     ```

   - On Linux or Mac:

     ```
     ./install.sh -r b2c_acc_plus start
     ```

   After successful startup, your SAP Commerce instance is installed and ready for you to use.

Next Steps

If you are using the `b2b_acc_plus` recipe, or if your installation includes the B2B AddOns for PunchOut, Secure Portal, or Account Summary, then update the system after starting SAP Commerce. To do so, log on to the SAP Commerce Administration Console, select **Platform** **Update** , select **Toggle all** under the **Project data settings** section of the **Update** page, and then click **Update**.

Related Information

Installer Recipes

Installer Recipes

Learn about the modules included in the available installer recipes for SAP Commerce.

Installer recipes are intended for demo and development purposes only, and should not be used for production systems. They provide a quick way to install the various extensions and AddOns for SAP Commerce. All recipes are located in the `installer/recipes` folder. Depending on your requirements, you use one of these recipes for your SAP Commerce installation.

For instructions on installing SAP Commerce using recipes, refer to Installing SAP Commerce Using Installer Recipes

This table lists all the available installation recipes found in the `recipes` folder and the modules included in each one. For more details about the extensions and AddOns included in each module, click the link in the Modules column.

Recipe Folder	Storefront	Modules
not applicable - applies to all recipes	not applicable	These modules are included in all installation recipes: • Adaptive Search Module • API Registry Module • Assisted Service Module • Backoffice Apps Module • Backoffice Framework Module • Base Accelerator Module • Base Commerce Module • Cockpit Applications • Cockpit Core • Commerce Services Module • Core Accelerator Module • Customer Service Module • Integration APIs • SAP Cloud Platform Extensions Integration Module • Search and Navigation Module • Text Field Configurator Template Module • WCMS Module

Recipe Folder	Storefront	Modules
b2b_acc_plus	powertools	Commerce B2B Accelerator Includes: • B2B Accelerator Module • B2B Accelerator AddOns Module • B2B Commerce Module • B2B Accelerator AddOns Module • Backoffice Addons Module • Coupon Module • Entitlements Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SmartEdit
b2b_c4c	powertools	SAP Cloud for Customer integration Includes: • B2B Accelerator Module • B2B Commerce Module • Backoffice Addons Module • SAP Cloud for Customer Integration Module • Coupon Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SmartEdit

Recipe Folder	Storefront	Modules
`b2b_china`	powertools	Commerce B2B Accelerator for China Includes: • B2B Accelerator Module • B2B Commerce Module • Backoffice Addons Module • China Accelerator AddOns Module • China Accelerator Address Module • Consignment Tracking Module • Coupon Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SmartEdit
`b2c_acc_cis`	• apparel • electronics	Commerce Infrastructure Services Includes: • Backoffice Addons Module • B2C Accelerator Module • Commerce Infrastructure Services (CIS) Module • Coupon Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SmartEdit

Recipe Folder	Storefront	Modules
`b2c_acc_plus`	• apparel • electronics	Commerce B2C Accelerator Includes: • B2C Accelerator Module • Backoffice Addons Module • Assisted Services Integration Module • B2C Accelerator AddOns Module • Coupon Module • Context-Driven Services Foundation Integration • eventtrackingws AddOn • Personalization Integration Module • Entitlements Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SmartEdit • SAP Hybris Merchandising

Recipe Folder	Storefront	Modules
`b2c_acc_ymkt`	• apparel • electronics	Includes: • B2C Accelerator Module • Backoffice Addons Module • B2C Accelerator AddOns Module • Backoffice Apps Module • Coupon Module • eventtrackingws AddOn • Platform Module • hybrisAnalytics AddOn • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • y2ysync • SAP Marketing Cloud • SmartEdit

Recipe Folder	Storefront	Modules
`b2c_b2b_acc_cpq`	• apparel • electronics • powertools	B2B Accelerator and B2C Accelerator with CPQ integration Includes: • B2B Accelerator Module • B2B Commerce Module • Backoffice Addons Module • B2C Accelerator Module • Coupon Module • SAP Framework Core Module • Product Configuration with SAP Variant Configuration and Pricing • SAP Product Configuration (On-Premise Edition) • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SAP Model Module • SAP Customer • SAP CPQ Integration Module • SCPI Integration Module • SmartEdit **Note** You can deploy and use one configuration engine only. Use either Cloud Engine (CPS), On-Premise Engine (SSC), or Mock Engine (demo only). **Note** Before you can use On-Premise Engine (SSC), you must install SAP CPQ Configuration Runtime Engine (also referred to as SSC) as described in the installation guide under Installing the Configuration Runtime Engine (FBS_SOLCONF_ENGINE 3.0). For more information on SAP Configure, Price, and Quote for product configuration in general, see SAP Product Configuration (On-Premise Edition).

Recipe Folder	Storefront	Modules
`b2c_b2b_acc_dp`	• apparel • electronics • powertools	B2B Accelerator and B2C Accelerator with digital payments integration. Includes: • B2B Accelerator Module • B2B Commerce Module • Backoffice Addons Module • B2C Accelerator Module • Coupon Module • SAP Digital Payments Add-On Integration Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SmartEdit
`b2c_b2b_acc_oms`	• apparel • electronics • powertools	B2B Accelerator and B2C Accelerator with Order Management Includes: • B2B Accelerator Module • B2B Commerce Module • Backoffice Addons Module • B2C Accelerator Module • Coupon Module • Order Management Services Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • yForms Module • Entitlements Module • SmartEdit

Recipe Folder	Storefront	Modules
b2c_c4c	• apparel • electronics	B2C Accelerator with SAP Cloud for Customer integration Includes: • B2C Accelerator Module • Backoffice Addons Module • SAP Cloud for Customer Integration Module **Note** Out-of-the-box, the recipe contains mocks. For information on how to set up SAP Commerce with SAP Cloud for Customer, see SAP Commerce - SAP Cloud for Customer Integration Guide.

Recipe Folder	Storefront	Modules
`b2c_china`	electronics	Commerce B2C Accelerator for China Includes: • B2C Accelerator Module • Backoffice Addons Module • China Accelerator AddOns Module • China Accelerator Address Module • China Accelerator Alipay PSP Module • China Accelerator Logistics Module • China Accelerator Payment Module • China Accelerator People Profile Module • China Accelerator Tax Invoice Module • China Accelerator WeChat PSP Module • Consignment Tracking Module • Coupon Module • Customer Coupon Module • Customer Interests Module • Message Center Module • Notification Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • Selective Cart Module • Stock Notification Module • Timed Access Promotion Engine Module • SmartEdit

Recipe Folder	Storefront	Modules
`marketplace_acc`	marketplacestore	Marketplace Accelerator Includes: • Coupon Module • Marketplace Accelerator • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SmartEdit

Recipe Folder	Storefront	Modules
`marketplace_acc_china`	chinesestoreaddon	Marketplace Accelerator with China-specific features. Includes: • Marketplace Accelerator • China Accelerator AddOns Module • China Accelerator Address Module • China Accelerator Alipay PSP Module • China Accelerator Logistics Module • China Accelerator Payment Module • China Accelerator People Profile Module • China Accelerator Tax Invoice Module • China Accelerator WeChat PSP Module • Coupon Module • Consignment Tracking Module • Customer Coupon Module • Customer Interests Module • Message Center Module • Notification Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • Selective Cart Module • Stock Notification Module • Timed Access Promotion Engine Module • SmartEdit

Recipe Folder	Storefront	Modules
`sap_aom_som_b2b_b2c`	• apparel • electronics • powertools	B2C Accelerator and B2B Accelerator with sync and async order management SAP integration Includes: • B2B Accelerator Module • B2B Commerce Module • Backoffice Addons Module • B2C Accelerator Module • Asynchronous Order Management Module • Coupon Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SAP Availability Module • SAP Credit Check Module • SAP Framework Core Module • SAP Model Module • SAP Customer • SAP Synchronous Pricing Module • Synchronous Order Management Module • Data Hub • SmartEdit

Recipe Folder	Storefront	Modules
`sap_oms_aom_b2b_b2c`	• apparel • electronics • powertools	Includes: • B2B Accelerator Module • B2B Commerce Module • Backoffice Addons Module • B2C Accelerator Module • Coupon Module • Order Management Services Module • Asynchronous Order Management Module • Personalization Module • Personalization Promotions Module • Personalization Search Module • Promotion Engine • Rule Engine • SAP Credit Check Module • SAP Framework Core Module • SAP Model Module • SAP Synchronous Pricing Module • Data Hub • SAP Customer

Removed Recipes

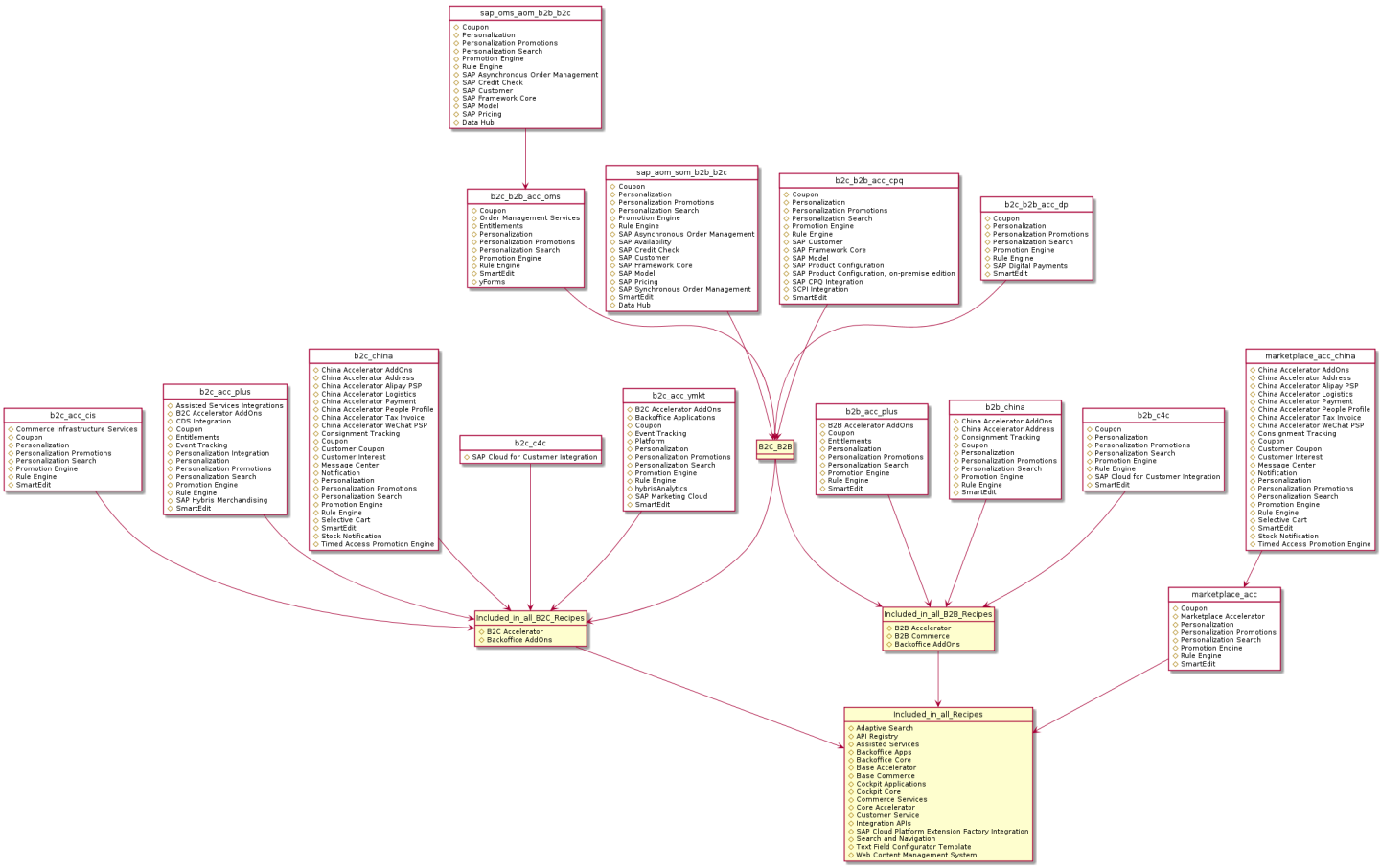
The following recipes were removed in the 1905 release:

- `b2b_acc`
- `b2c_acc`
- `b2c_acc_cpq_dockerized`
- `b2c_acc_dockerized`
- `b2c_acc_dockerized_gridfs`
- `b2c_acc_oms`
- `b2c_b2b_acc`
- `b2c_b2b_oms_dockerized`
- `b2c_oms_dockerized`
- `base_images`
- `centralized_logging`
- `platform_only`
- `platform_only_dockerized`
- `platform_only_gridfs_dockerized`

- `y2ysync_dockerized`

Recipe Layers

This diagram illustrates the layers of the various recipes (Open in a new tab to view the entire image).



Related Information

Installing SAP Commerce Using Installer Recipes
Extensions and AddOns

Creating Installer Recipes

You can create your own Gradle recipe to easily install the custom SAP Commerce configuration of your choice.

The installer is a Gradle-based project written in Groovy that simplifies installing SAP Commerce. It is a script that takes care of creating directories, moving files, updating configuration and properties files, and initializing the system.

You can create your own installer recipes to install a specific SAP Commerce configuration, with the desired applications such as B2C Accelerator, Telco Accelerator, or Datahub, for your own needs. A recipe contains all the information required to install and configure the setup, such as the following:

- Three mandatory tasks: setup, initialize, and start.
- Calls to required plugins, such as `installer-platform-plugin.jar`, `installer-addon-plugin.jar`, and `installer-coreplus-plugin.jar`.
- Local properties
- Extensions
- Database configuration

- Web archives
- Server information, for example: Apache Tomcat.

Installer Recipes

The content of installer recipes varies depending on your needs. The only mandatory content for all installer recipes are the three tasks `setup`, `initialize`, and `start`.

Your installer recipe may require additional information, such as the following:

- Plugins
- Extensions
- Local Properties
- Servers

ℹ Note

This list is not exhaustive and your recipe may require information not listed here.

Mandatory Tasks

You must include the following three tasks in your installer recipe:

- **setup**: Installs the recipe in the installer, which is primarily the copying of files. The setup task is invoked by default if no other task is specified with the install command.
- **initialize**: Initializes the application.
- **start**: Starts the application.

Additionally, the installer automatically adds a perform task that invokes all the above-mentioned tasks in the order in which they are presented.

The following example shows how these mandatory tasks might be implemented in a recipe called `platform_only`.

```
task setup << {
                    platformOnly.setup()
                    }
                    task initialize << {
                    platformOnly.build()
                    platformOnly.initialize()
                    }
                    task start << {
                    platformOnly.startInBackground()
                    }
```

Plugins

You must define the plugins that your SAP Commerce setup requires in your recipe. Most recipes include a call to the `installer-platform-plugin.jar` plugin. Other frequently referenced plugins include the following:

- `installer-addon-plugin-.jar`
- `installer-coreplus-plugin-`***`<version>`***`.jar`

You can find all available plugins in the `\hybris\installer\libs` directory. Include your required plugins at the beginning of your recipe as shown in the following example. You can then reference them later in your recipe.

```
apply plugin: 'installer-platform-plugin'
                    apply plugin: 'installer-coreplus-plugin'
```

You can also reference your required plugins elsewhere in your recipe as shown in the following example.

```
task setup << {
                    apply plugin: 'installer-platform-plugin'
                    def pl = platform(config)
                    pl.setup()
                    }
                    task initialize << {
                    apply plugin: 'installer-platform-plugin'
                    def pl = platform(config)
                    pl.build()
                    pl.initialize()
                    }

                    task start << {
                    apply plugin: 'installer-platform-plugin'
                    def pl = platform(config)
                    pl.start()
                    }
```

Extensions

You must include all required extensions in your recipe.

The following example shows how extensions are added.

```
extensions {
                    extName 'yacceleratorcockpits'
                    extName 'yacceleratorinitialdata'
                    extName 'yacceleratorstorefront'
                    extName 'yaddon'
                    extName 'ycommercewebservices'
                    extName 'electronicsstore'
                    extName 'apparelstore'
                    extName 'b2ccheckoutaddon'
                    extName 'liveeditaddon'
                    extName 'acceleratorwebservicesaddon'
                    }
                    }
```

Local Properties

You must specify any local properties that your SAP Commerce setup requires.

The following example shows how these local properties are defined.

```
localProperties {
                    property 'commerceservices.default.desktop.ui.experience', 'responsive'
                    property 'uiexperience.level.supported', 'DESKTOP'
                    }
```

Web Servers

You must specify information related to any web servers that you require.

The following example shows how the mandatory tasks are implemented.

```
def setupOMSTomcat() {
                    def CATALINA_OPTS = "-Xms4096m -Xmx4096m"
                    tomcat.instance('oms').setup {
                    ports {
                    http 8080
                    ssl  8081
                    }
                    webApps {
                    webApp 'oms-rest-webapp.war', file(suiteHome+'/hybris-oms/binary/webapp/oms-rest-w
                    }
                    libraries {
                    lib file(suiteHome+'/hybris-oms/sample-config/com.hybris.oms_oms-rest-webapp-logba
                    }

                    }.start(CATALINA_OPTS)
                    }
```

Creating Your Installer Recipe

Procedure

1. Using Groovy, write your Installer recipe following the information provided in Installer Recipes.
2. Create a folder for your recipe in the `/hybris/installer/recipes/.`

   The name of your recipe must be the same as the name of your folder.
3. Create a `README.txt` file that describes what your recipe does and the commands required to setup, initiate and start your recipe.
4. Save your recipe as a `build.gradle` file in the recipe's folder.

Related Information

Installing SAP Commerce Using Installer Recipes

Installer Platform Plugin

Groovy User Documentation

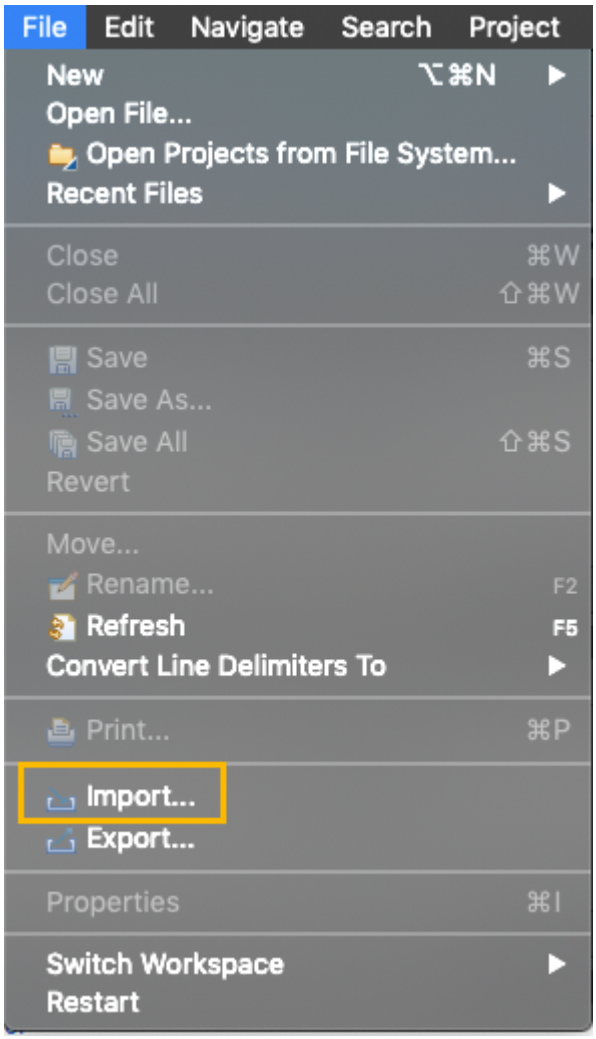
Gradle User Documentation

Setting Up an Eclipse Project

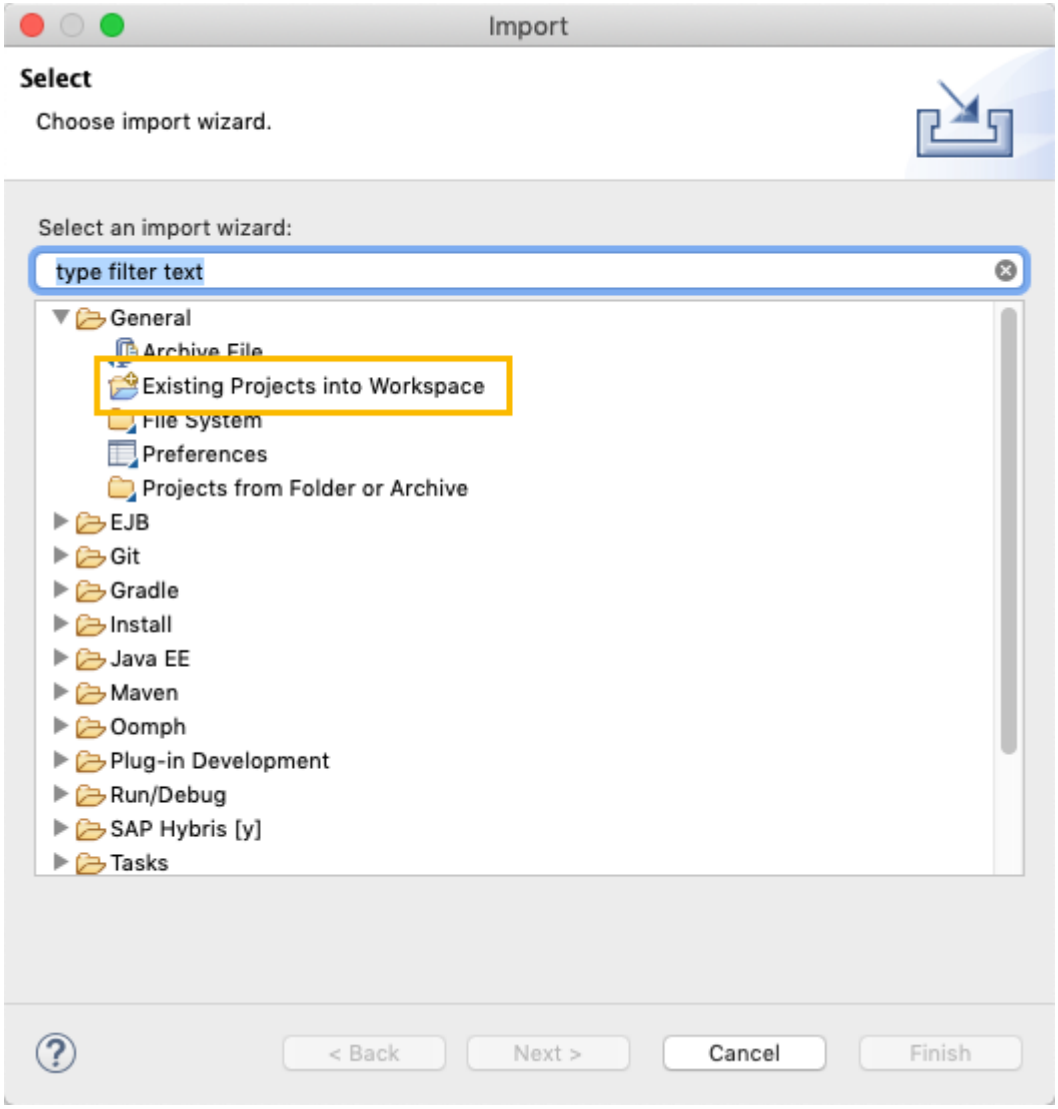
Preconfigured Eclipse `.project` and `.classpath` files allow you to import SAP Commerce and its packages into Eclipse, ready for custom development.

Procedure

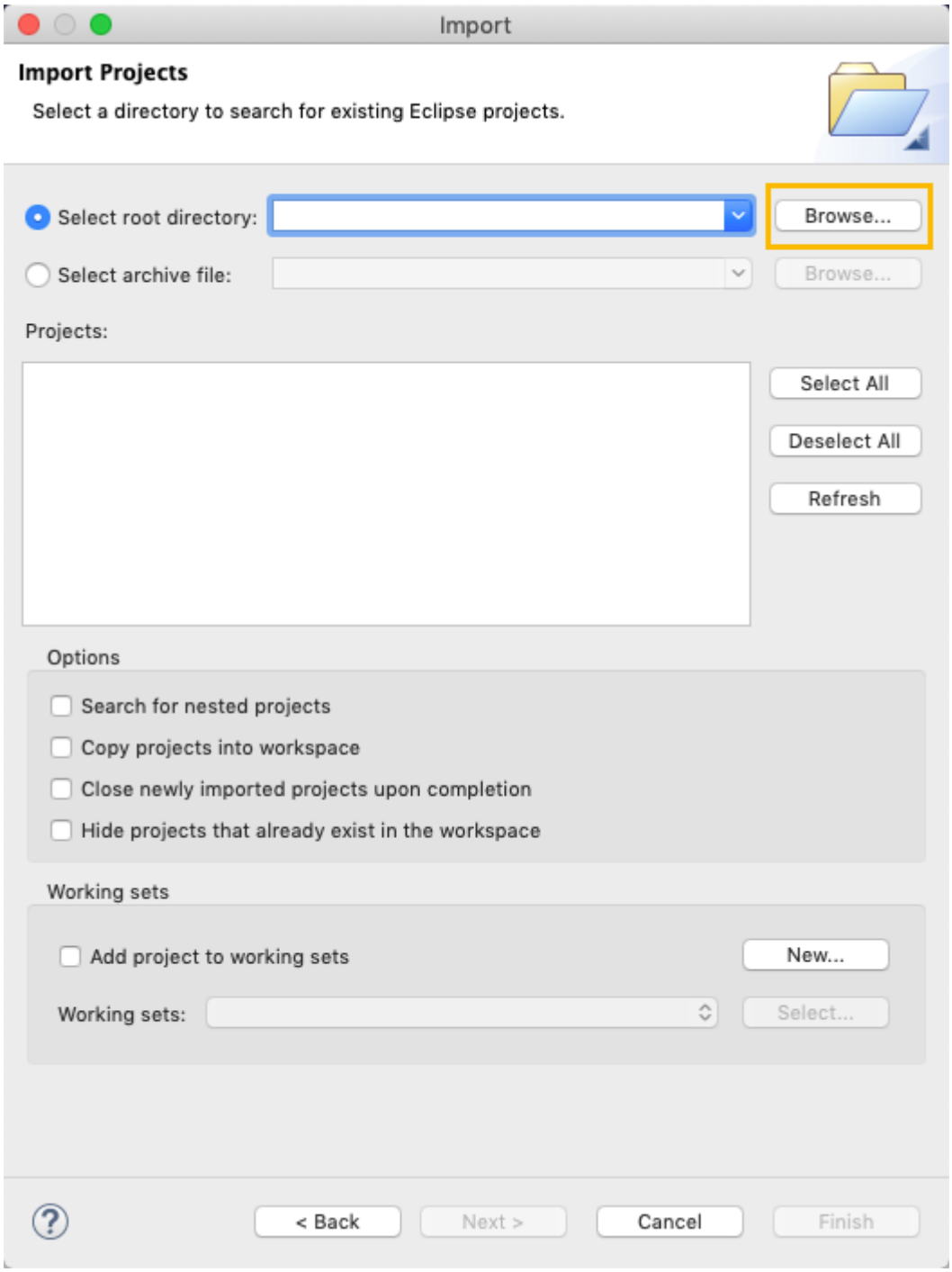
1. Download and unzip SAP Commerce and its packages into a single directory. By default this is a `hybris` directory for all packages.
2. In Eclipse, open the **File** menu and click **Import**.



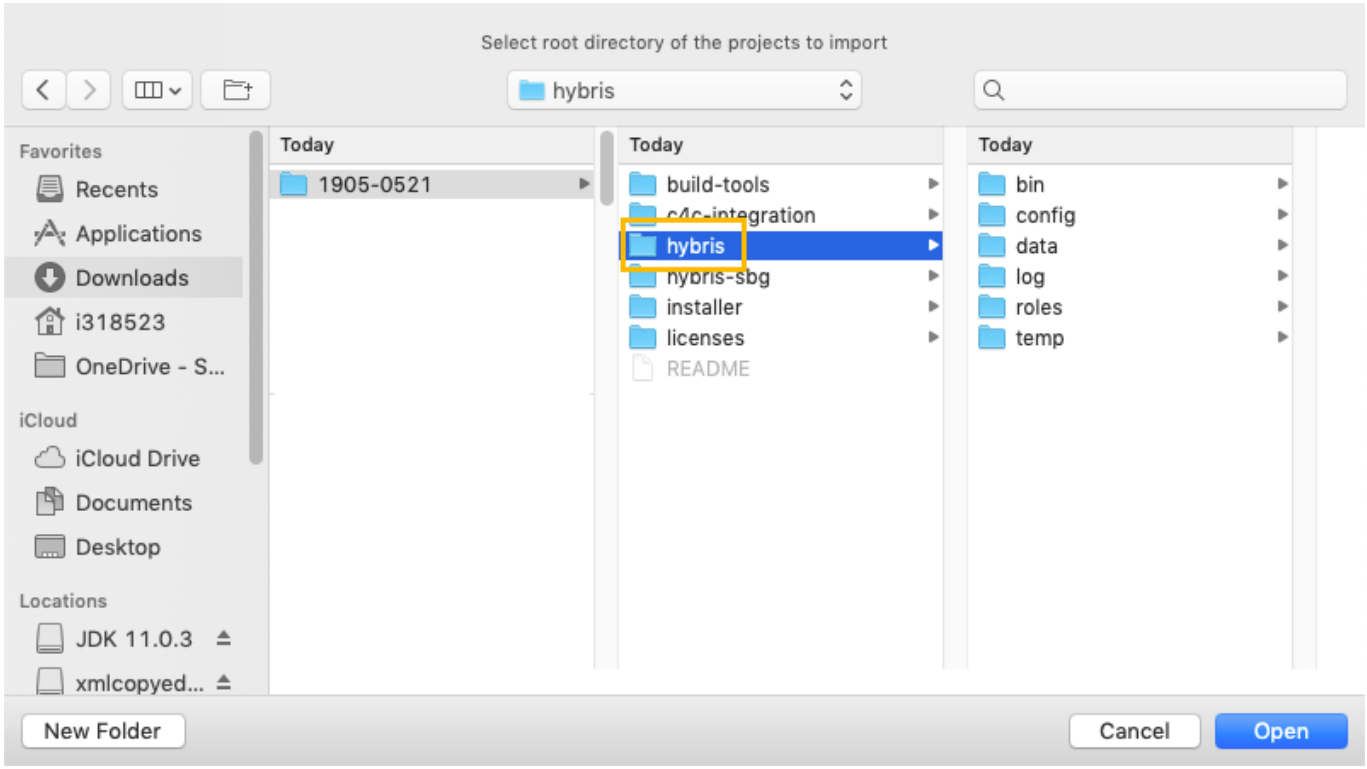
3. In the Select screen of the Import wizard, expand the **General** section, and select Existing Projects into Workspace. Click the Next button to continue.



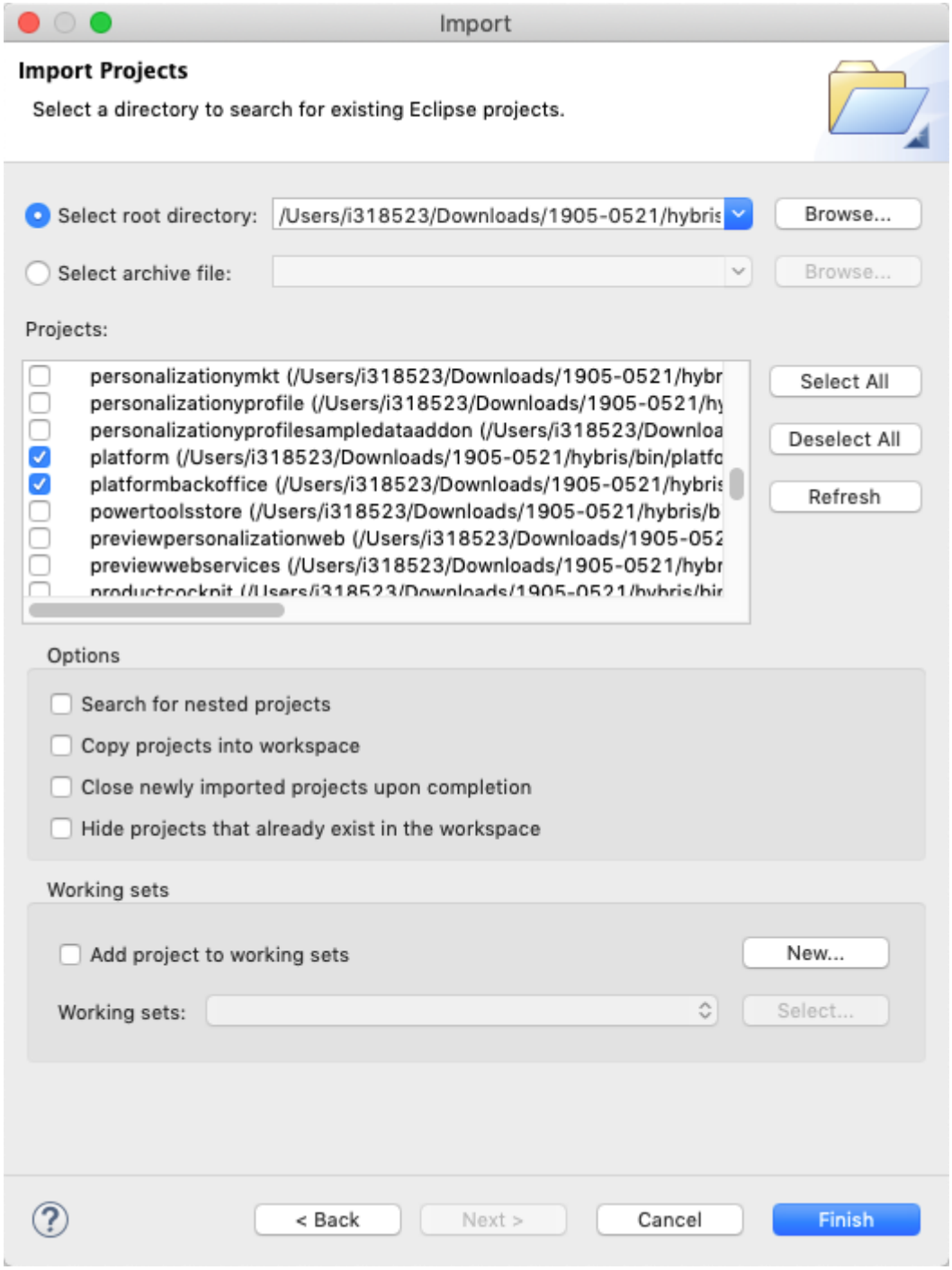
4. In the Import Projects screen of the Import wizard, click on the Browse button for the Select root directory field.



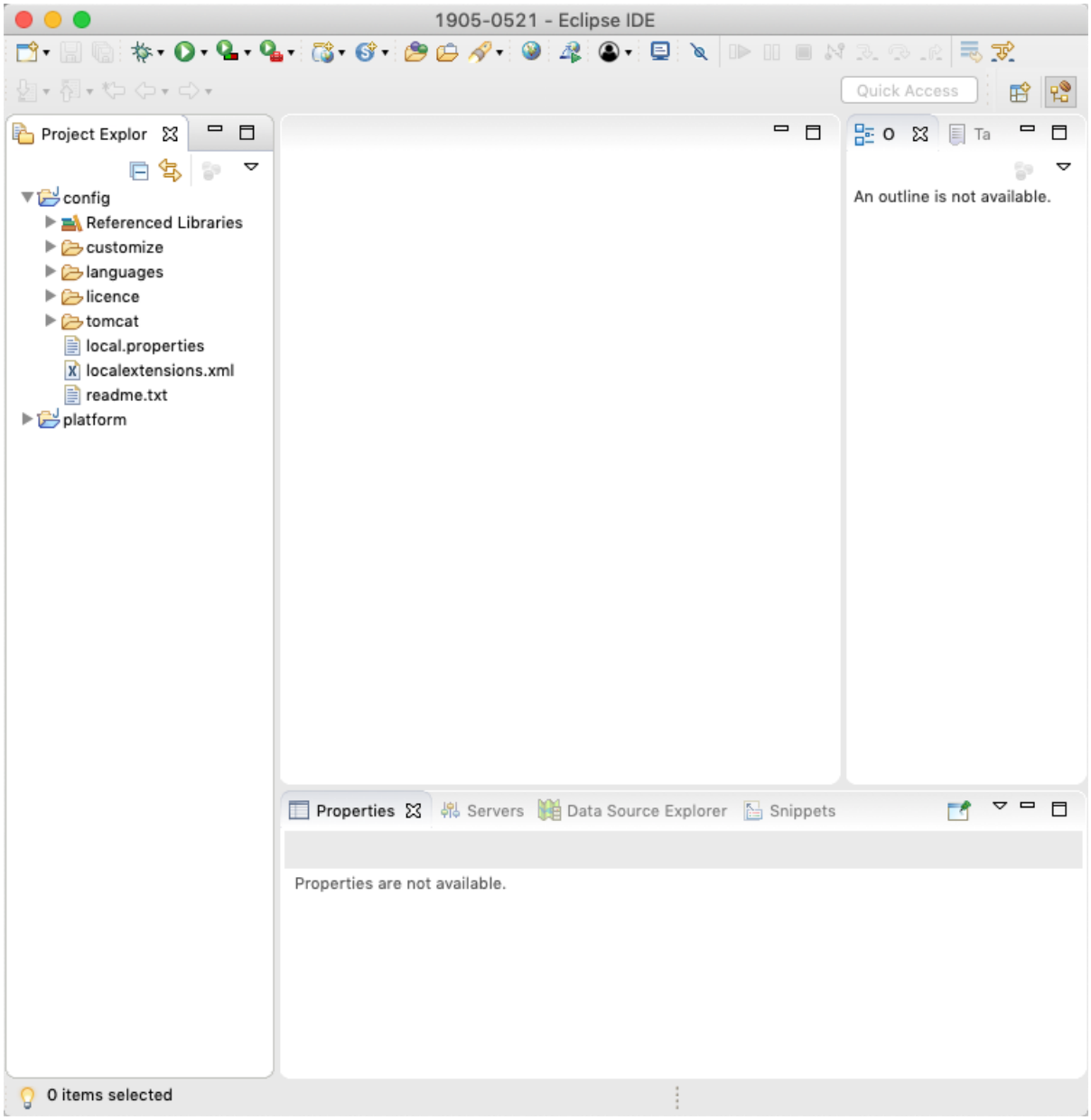
5. Navigate through the file system to the `hybris` directory into which you have unzipped the SAP Commerce files. Select the `hybris` directory and click the Open button.



6. Back on the **Import Projects** screen of the **Import** wizard, check or clear the projects Eclipse has located to include or exclude the projects you need, then click the **Finish** button.



Eclipse now includes SAP Commerce and its extensions into its currently active workspace:



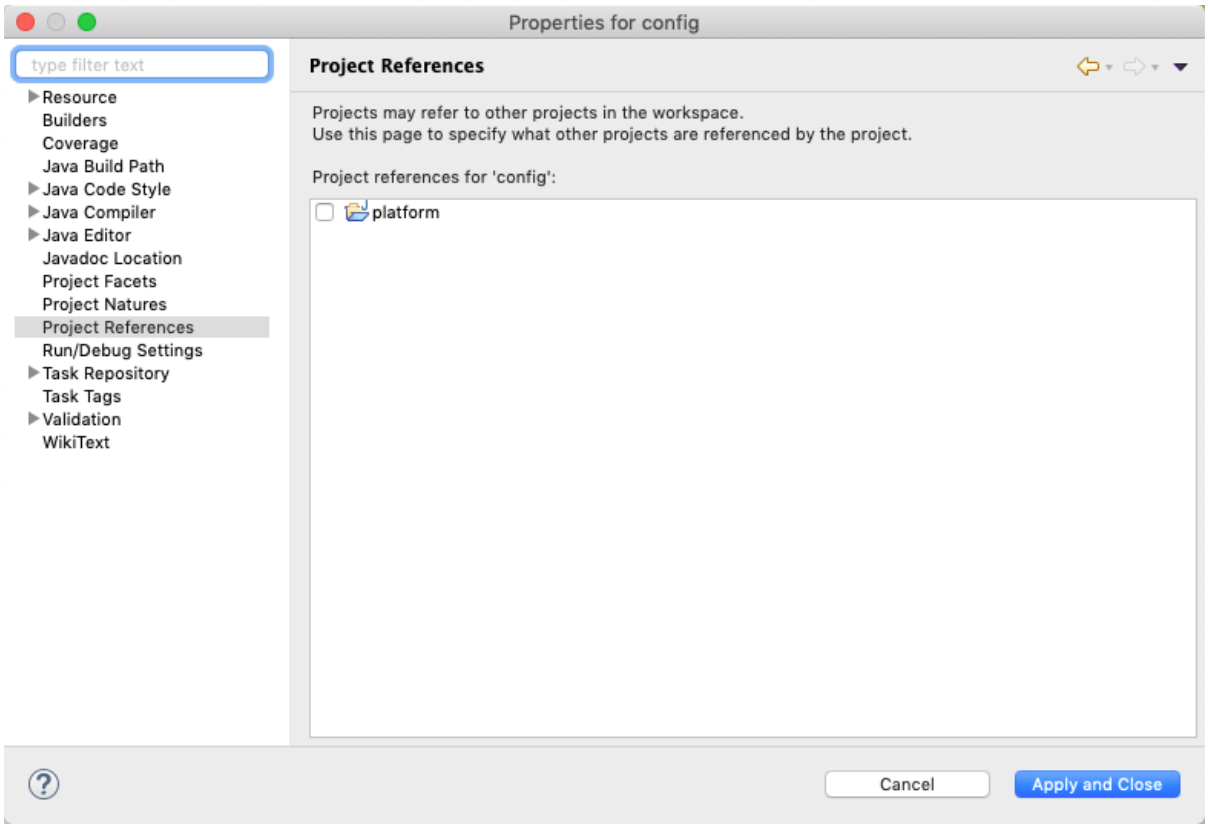
Note

All the imported Eclipse projects should be without such errors like the one in the following example:

```
Class CMSRequestHandler can not be found ...
```

If this happens, it means that you probably imported the project before you first run the initial Ant in the `platform` directory. The eclipse errors occur, because some classes are initially copied during this Ant process. So make sure that you first run Ant and then setup eclipse.

7. **Optional:** The project may depend on other projects. In Eclipse, in the Project Explorer view right-click the project name and choose Properties from the context-menu. The Properties dialog box for the chosen project appears. Go to the Project References option. Here you can check the other projects that are referenced by the current project.



Related Information

Gradle Support

Third-Party Databases

A wide variety of third-party databases may be used with SAP Commerce. This gives great flexibility, allowing you to choose a database best suited to your solution.

Supported Database Systems

SAP Commerce supports a variety of database systems. By factory default, it comes pre-bundled with and pre-configured for the HSQLDB database.

Database System	Description
HSQLDB	• Pre-bundled • Pre-configured • Runs out-of-the-box
Oracle	• Property configuration overview • Some database administration experience recommended
MySQL	• Property configuration overview • Some database administration experience recommended

Database System	Description
Microsoft SQL Server	• Property configuration overview • Some database administration experience recommended
HANA	• Property configuration overview • Some database administration experience recommended

ℹ Note

Choose a CI (case insensitive) collate when creating a database for SAP Commerce.

The commonly used collations are:

- **Latin1_General_CI_AS**:Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
- **SQL_Latin1_General_CP1_CI_AS**:- Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data

Enabled accent sensitivity otherwise 'a' will be treated the same as 'á' (note with an accent).

ℹ Note

For Microsoft SQL Server ensure, that the database has been properly set up as described in Microsoft SQL Server.

MySQL

MySQL is a common, open-source database system available for a variety of operating systems.

HSQLDB

HSQLDB is a light-weight SQL database that can be run within a Java Virtual Machine. It is useful for development environments due to its speed. SAP Commerce comes pre-bundled with a compatible version of HSQLDB.

Oracle

Oracle Database is a common enterprise database system, available for a variety of operating systems.

Microsoft SQL Server

Microsoft SQL Server is a database system available for Microsoft Windows operating systems.

SAP HANA Database

SAP HANA is a relational, in-memory database, and the main component of the SAP HANA platform. To use SAP HANA, you need to configure the Platform by adding properties to your `local.properties` configuration file.

JNDI

There are a few important JNDI-related topics you should be aware of.

MySQL

MySQL is a common, open-source database system available for a variety of operating systems.

Installation

For supported MySQLDB versions, see System Requirements.

Install MySQL as specified at http://dev.mysql.com/doc/ .

ℹ Note

The MySQL connector is not shipped with SAP Commerce. Download the file and put it in the `${HYBRIS_BIN_DIR}/platform/lib/dbdriver` folder.

Configuration Overview

The following table contains a list of all MySQL-specific configuration properties in the `${HYBRIS_HOME}/bin/platform/project.properties` file. Please refer to Configuring the Behavior of SAP Commerce for a full list of all properties.

Property Name	Property Description
`db.url`	Specifies a path to access the database, such as **`jdbc:mysql://localhost/hybris?useConfigs=maxPerformance`**. The `useConfigs=maxPerformance` parameter retrieves a number of properties optimized for the combination of MySQL and SAP Commerce. These properties are stored within the MySQL connector driver file that you need to download manually. The properties are used by default after finding `mysql-connector-java-x.x-bin.jar` in the ${HYBRIS_BIN_DIR}/platform/lib/dbdriver directory). This connector driver JAR file contains a property file with optimized settings for MySQL and SAP Commerce. This property file is stored in the driver JAR file (`com/mysql/jdbc/configs/maxPerformance.properties`). The `useConfigs=maxPerformance` parameter loads all properties of the `maxPerformance.properties` file. The settings on MySQL 8.x are similar. Potential problems related to the ssl mode and jdk11 have been observed on MySQL 8.x and Percona: MySQL 8.x tests successfully when the ssl mode is disabled: `sslMode=DISABLED` (available since MySQL 8.0.13. For more information, see https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-reference-using-ssl.html). For Percona, it is also recommended to disable the ssl mode in the `db.url` property by setting `useSSL=false`.
`db.driver`	The fully qualified name of the JDBC driver class used to access the database. To use a required driver version, in `project.properties` set: • `com.mysql.jdbc.Driver` for a 5.x driver • `com.mysql.cj.jdbc.Driver` for a 8.x driver The `project.properties` file default is `com.mysql.jdbc.Driver`.

Property Name	Property Description
`db.username`	The name of the user account used to access the database. **Note** **Do Not Use High-level System Administration User Accounts** Do not use the **root** database user account to run SAP Commerce for security reasons. Use a non-privileged database user account instead.
`db.password`	The password matching the database user account specified in **db.username** property.
`db.tableprefix`	An optional String that precedes a set of table names in a database schema. Specifying different table prefixes allows you to create different logical sets of tables within one schema, and thus running several instances of SAP Commerce on one single database schema.
`mysql.optional.tabledefs`	Allows you to specify extra run-time parameters for the MySQL database.
`mysql.tabletype`	Lets you specify the type of MySQL database tables you want to use. Please refer to the MySQL documentation on Storage Engines for details.
`mysql.allow.fractional.seconds`	Set this property to true if you're using **MySQL 5.6.4 or later**. This flag allows to create datetime columns which support fractional seconds. For more information, see https://dev.mysql.com/doc/relnotes/mysql/5.6/en/news-5-6-4.html **Note** With MySQL 5.6.4 (or later) the `mysql.allow.fractional.seconds` property **must** be defined. Otherwise java.util.Date columns may behave in the strange way of date values getting **rounded up** (means into the future) instead of being truncated. **Before** 5.6.4 you must either **omit** the mysql.allow.fractional.seconds property or set it for **false**!
`db.customsessionsql`	Because of recent changes in MySQL transaction handling, isolation level **Repeatable Read** may cause frequent deadlocks. To change the isolation level to **Read Committed**, you can set the property **db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED.**
`db.mapping.char.legacy`	With the `db.mapping.char.legacy` property set by default to `true` (legacy), the char values are mapped to integer/byte values. By setting this property to `false` (disabling the legacy flag), you force the conversion of column types to **char** values. The reason the default value is `true` is because only this setting works correctly on already existing databases, on system update, without additional, manual work. If you disable the legacy mode for system update, you need to first convert the existing columns types (for example from int to char) manually.

For copy & paste please use this template:

```
# connection
db.url=jdbc:mysql://<host>:<port>/<dbname>?useConfigs=maxPerformance&characterEncoding=utf8
db.driver=com.mysql.jdbc.Driver
db.username=<username>
db.password=<password>

# table prefix e.g. if db is shared
db.tableprefix=

# other *mandatory* settings
db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
mysql.optional.tabledefs=CHARSET=utf8 COLLATE=utf8_bin
mysql.tabletype=InnoDB

# MySQL DATETIME handling
#   from MySQL 5.6.4 -> mysql.allow.fractional.seconds=true
#   before MySQL 5.6.4 -> mysql.allow.fractional.seconds=false (or omit the property)
#
mysql.allow.fractional.seconds=true
#
# Note: after changing this property you must re-run initialization / update because
# is only has a effect if the initialization / update DDL statements are re-generated!
#
```

MySQL database tables in lowercase

SAP Commerce stores all MySQL database tables in lowercase to prevent difficulties on Unix-related systems (Linux, etc.) with mixed-case database table names.

Using UTF-8

A character set defines the list of characters that are available for an operating system or an application to choose from. An encoding defines how those available characters are transformed into bytes or groups of bytes, so that they can be sent (to another application or an external server, for example). For example, in ASCII encoding, the letter A is encoded as the number value 65. In other words, if, in a transmission, the number 65 comes up, the system knows that A is meant. Similarly, p is 112, and # is 32.

However, Unicode describes several dozens of thousands of characters, each of which requires 4 bytes to store in native Unicode representation. Many latin-based alphabets, however, make use of only a limited number of special characters that are not represented by the ASCII standard. For those alphabets, using 4 bytes for every character was considered too much, and thus encodings like UTF-16 and UTF-8 were designed. UTF-8 stores US-ASCII-based characters in one single byte and non-US-ASCII characters in two to four bytes. As it forms a compromise between compact storage and full support of all Unicode characters and allows relatively easy conversion of ASCII texts, UTF-8 is becoming a very popular encoding.

Both SAP Commerce and MySQL support UTF-8 out-of-the-box. MySQL, however, needs to know about a character set's inherent rules for comparisons and ordering - whether D is equal to d (case sensitivity), for example. This set of rules is referred to as **collation** in MySQL terms. For extra details on collations, please refer to the MySQL documentation of your version in use.

Setting Up a UTF-8-Based MySQL Database

ℹ Note

New database required

This article assumes that you create a MySQL database from scratch and do not have to convert an existing database.

In an out-of-the-box MySQL installation, the default character set is **latin1**. You need to set it to UTF-8 as SAP Commerce uses UTF-8 as native encoding.

Linux Systems

On Linux systems, MySQL is configured via instances of the `my.ini` stored as `my.cnf` in various locations. Depending on the location of the files settings are defined in, those settings' values have a certain priority. If a parameter is set in more than one place, the value from the file with the highest priority apply.

Location	Settings Apply	Priority
`/etc/my.cnf`	server-wide	lowest
`${mysql_data_dir}/my.cnf`	for the database located in `${mysql_data_dir}`	medium
`\~/.my.cnf`	to the respective user	high

Windows Systems

There are two ways to set the default encoding on Windows systems: via the graphical MySQL Server Instance Config Wizard and via the `my.ini` file (as under Linux).

The MySQL Workbench

In the MySQL Workbench, you may administer the server and set up basic values for the `my.ini` file without manual editing the file.

1. Open the MySQL Server Instance Config Wizard (by clicking on **Start > Programs > MySQL > MySQL Workbench 5.2 CE** on MySQL 5.6, for example)
2. Double-click the server on the server administration section
3. Click Options file under Configuration
4. On the General tab in the International section, be sure to set the **character-set-server** option to **utf8** and select **skip-character-set-character-set-client-handshake**.
5. On the InnoDB tab under the Logfiles section be sure to set the **innodb_flush_log_at_trx_commit** option to **0** in order to improve performance drastically.

Using the `my.ini` File

Under Windows, by default there is only one instance of the `my.ini` file located in `C:\ProgramData\MySQL\MySQL Server 5.6`. Please edit it to specify the default character set. For details, please refer to the next section.

`my.ini` file

The overriding of parameters through `my.ini` files of different priorities means that you need to check whether UTF-8 is activated (via the **status** command on the MySQL command shell, for example). If UTF-8 is activated (as displayed in the following code snippet from the MySQL command shell, you're set:

```
Server characterset:    utf8
Db     characterset:    utf8
Client characterset:    utf8
Conn.  characterset:    utf8
```

The character set is specified by the **--character-set-server** parameter.

To change the character set options in `my.ini` you have to set the following variables:

```
[client]
default-character-set=utf8

[mysql]
default-character-set=utf8

[mysqld]
character-set-server=utf8
skip-character-set-client-handshake
```

ℹ Note

Mind the Override Priorities

Make sure that this value is not overridden in any other, more highly priorized `my.ini` file.

Time zone settings

If you want to utilize time zone support in your MySQL database, use the following command: `mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql`

Add the following property to the connection string with a relevant time zone value, for example: `serverTimezone=Europe/Berlin`

Performance

In order to get a rough impression of database performance, SAP recommends using the same database configuration for testing that will be used in live operation.

MySQL comes with various database engines, referred to as **storage engines** (please refer to the MySQL documentation for more details). Depending on the storage engine used for a database table, the database table is or is not transaction-safe. For example, MyISAM database tables are not transaction-safe, InnoDB database tables and BDB database tables are transaction-safe.

Depending on the storage engine, there are different kinds of **transaction isolation levels**. A transaction isolation level specifies how strictly individual transactions are isolated, that is, how strictly the data set in the database is kept apart between different transactions. The higher the transaction isolation level, the better data is isolated and the worse for performance. In other words, with higher transaction isolation levels, you trade performance for data consistency. Estimation of transaction isolation levels of InnoDB database tables (overview by MySQL):

Transaction Isolation Level	Estimation
Serializable	Not recommended for two reasons: • there is a chance that you may run into ***phantom reads*** (A ***phantom read*** occurs when two identical queries are executed within a single transaction, but the set of rows returned from the second query is different from the first). • **serializable** consumes a lot of database performance.
Repeatable read	Recommended.
Read committed	Recommended in case **repeatable read** does not deliver enough performance or causes too many deadlocks.

Transaction Isolation Level	Estimation
Read uncommitted	Not recommended: this basically results in transactions not being isolated. In other words: InnoDB database tables running the **read uncommitted** transaction isolation level are not transaction-safe.

Using transaction-safe storage engines is recommended for development purposes. By default, SAP Commerce uses **InnoDB** instead of **MyISAM**. Using **MyISAM** database tables reduces database performance heavily compared to **InnoDB** database tables. Especially during system initialization when a lot of database statements are executed, **InnoDB** runs much quicker (about three times - or possibly more).

ℹ Note

SAP Commerce Warns Automatically If Settings Are Non-performant

In combination with **InnoDB**-based database tables, the **innodb_flush_log_at_trx_commit** setting has a massive impact on database performance. Please refer to the MySQL documentation (MySQL 5.0 version) for details. SAP Commerce determines whether the **innodb_flush_log_at_trx_commit** is set to **0** or not and displays a warning if not. For details, see PLA-6128.

Related Information

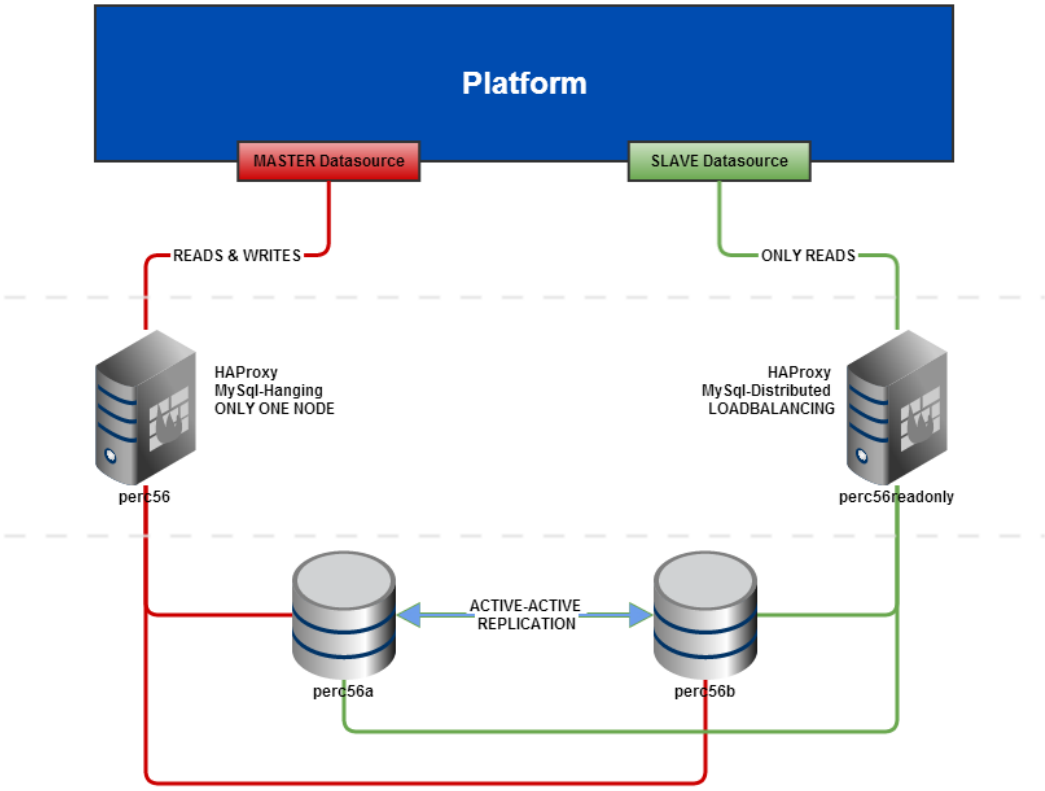
http://www.mysql.com
http://dev.mysql.com/doc/

Configuring a Percona MySQL Cluster

Percona XtraDB Cluster is a high performance active-active database cluster used in high availability, close-to-MySQL scenarios. It offers a range of enterprise database features over and above standard MySQL.

Percona XtraDB Cluster offers:

- 100% binary and API compatibility with MySQL (version numbers are aligned)
- Enterprise support by an international company Percona
- active-active clustering
- a "real" backup tool (called xtrabackup)
- ~20% higher performance than MySQL



Naming Convention

The following naming convention for nodes and proxies is used throughout this installation guide:

Separate machines

- **perc56a** is Percona cluster node 1
- **perc56b** is Percona cluster node 2
- **percquorum** is the Percona arbitrator node used to provide an even number of nodes

Separate IP addresses (virtual IP's)

- **perc56** is the Virtual IP ensuring "hanging" behavior (prefer one node to avoid race conditions)
- **perc56readonly** is the Virtual IP ensuring round robin access dedicated to read use.

Hardware Recommendations

It is recommended to use a powerful set of machines for cluster node 1 and 2 with at least two network interface cards (cluster and data network) because these machines are the database servers. Hardware sizing should be done according to MySQL / Percona specs , depending on the expected workload. Both cluster nodes keep a copy of the full database and each needs to fulfill the performance requirements separately. Please plan accordingly: the cluster described here is twice as expensive as a single node (since two nodes with data load).

The arbitrator node (**percquorum**) does not get any load except network load. A minimum set of cores (2) and RAM (2GB) are recommended, it requires one network interface card to reach the cluster node and a second one just to monitor.

→ Tip

Read optimization through replica sets

Consider a separate read-only layer also knon as standard MySQL replication. The master (and therefore the write target) in this replication set can be set up as a Percona HA active-active cluster. This combines the scaling advantages of a MySQLreplication set with the availability of the Percona active-active cluster. MySQL replicas are used to optimize read performance; write is mainly correlated to disk performance. To scale a write, the disk performance needs to be scaled up, scale out is not possible in MySQL. Because Platform mostly executes read operations, the scaling with replicas is a viable option in this environment.

→ Tip

Higher availability through higher number of nodes

The described setup is the minimum in an active-active setup. If you want to achieve higher availability, for example, while servicing a node or while backing up, you need to increase the number of nodes (typical 3, so you are still high available while servicing one node). If you increase the number of cluster members, scale hardware, too. (A three node solution is ***three*** times as expensive as a single node solution and so on)

Services Used

Make sure that the cluster works seamlessly as you have to ensure not only the Percona product is needed (in order of installation and priority):

Service	Function
ntp / ntpdate	keeping the time in sync
Percona XtraDB Cluster 5.6	**providing the database**
haproxy / mysqlproxy	controlling the load balancing data flow
keepalived	maintaining the virtual ip fail over between the nodes
inetd / xinetd	making the cluster check available to the other node(s)
clustercheck.sh	monitoring the cluster status
transparent proxying (TPROXY)	make sure IP based acces restrictions are still working
firewall with e.g. ferm	ensure the security of the box

Preparing the Machines

Platform uses default Debian stable as operating system.

1. Exchange ssh root keys (since the "valuable data" on the nodes is contained in the sql cluster and those credentials must be shared anyhow, this actually not lowerring the security). In this section we establish also a sync script for easy handling
2. Add Percona repositories to the server repos on ALL nodes

    a. add the repos:

    /etc/apt/sources.list.d/percona.list

    ```
    deb http://repo.percona.com/apt wheezy main
    deb-src http://repo.percona.com/apt wheezy main
    ```

    b. add the valid Percona key to verify the signature of packages coming from Percona

```
> gpg --keyserver  keys.gnupg.net --recv-keys CD2EFD2A
> gpg -a --export CD2EFD2A | apt-key add -
```

c. change the priority to accept Percona packages in favor of maybe outdated debian version

/etc/apt/preferences.d/percona

```
Package: *
Pin: origin repo.percona.com
Pin-Priority: 990
```

d. make sure your repositories are uptodate

```
> apt-get update
```

3. Sync the time between the nodes and make them peer each other (already a drift of 0.5s is a major issue for the cluster)

a. install ntp

```
> apt-get install ntp lockfile-progs
```

(lockfile-progs is needed in debian to ensure a proper restart)

b. modify the ntp config file

/etc/ntp.conf

```
# select an adequate time server "near by"
server time1.fra.hybris.com iburst
server time2.fra.hybris.com iburst

# make sure to peer between servers
peer perc56a.database.fra.hybris.com iburst
peer perc56b.database.fra.hybris.com iburst
peer percquorum.database.fra.hybris.com iburst

# By default, exchange time with everybody, but don't allow configuration and peering.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

# allow peering
restrict perc56a.database.fra.hybris.com peer
restrict perc56b.database.fra.hybris.com peer
restrict percquorum.database.fra.hybris.com peer

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
```

(this file is valid on all servers, the reference to the server itself is automatically ignored, so distribute on ALL servers without modification)

→ Tip

the percquorum server could be used in different clusters, make sure it is peering to all nodes of all clusters it is servicing.

c. Restart ntp

```
> service ntp restart
```

d. Verify correct operation

```
> ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*timec.fra.hybri 195.50.171.101   3 u    6   64  377    0.376    5.721   4.997
```

```
-timeb.fra.hybri 129.70.132.33    3 u   62   64  377    0.374   14.931   4.704
+perc56a.databas 10.8.1.148       4 u   26   64  377    0.789    2.112   4.491
+percquorum.data 10.8.2.149       4 u   37   64  377    0.829    1.962   3.758
```

4. Make sure network confiuration is working
5. Exchange ss

Install Percona Cluster Node

1. Install Percona on the cluster nodes 1 and 2 :

```
apt-get install percona-xtradb-cluster-56
```

(it will ask you for the initial database password. The one you specified on node 1 will be important later in this document)

2. To be able to use the slave data source in Platform, add the following configuration to your `local.properties` file:

```
db.url=jdbc:mysql://perc56.<youraddress>.com/<dbname>?useConfigs=maxPerformance&characterEncodin
db.driver=com.mysql.jdbc.Driver
db.username=<username>
db.password=<password>
db.tableprefix=
mysql.optional.tabledefs=CHARSET=utf8 COLLATE=utf8_bin
mysql.tabletype=InnoDB
db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
mysql.allow.fractional.seconds=true
slave.datasource.1.db.url=jdbc:mysql://perc56read.<youraddress>.com/<dbname>?useConfigs=maxPerfo
slave.datasource.1.db.driver=com.mysql.jdbc.Driver
slave.datasource.1.db.username=<username>
slave.datasource.1.db.password=<password>
slave.datasource.1.db.tableprefix=
slave.datasource.1.mysql.optional.tabledefs=CHARSET=utf8 COLLATE=utf8_bin
slave.datasource.1.mysql.tabletype=InnoDB
slave.datasource.1.db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
slave.datasource.1.mysql.allow.fractional.seconds=true
```

3. Install an external disk as `/dev/vdb` for mysql data on nodes 1 and 2 in /etc/fstab:

```
LABEL=data      /media/data0    ext4    noatime,nobarrier,discard       0       1


/dev/vdb        197G 1015M  186G   1% /media/data0
```

4. Configure percona on nodes 1 and 2 in `/etc/mysql/my.cnf`:
   - set the log to `/var/syslog`

     my.cnf

     ```
     syslog
     syslog-facility=local0
     ```

   - configure the cluster and nodes. Put this configuration on node 1

     my.cnf

     ```
     wsrep_cluster_address=gcomm://perc56b.<yourinternaladdress>.com,sqlquorum.<yourinternaladd
     wsrep_cluster_name=<yourclustername, for example perc_mygreatcluster>
     wsrep_node_name=perc56a.<yourinternaladdress>.com
     wsrep_data_home_dir=/var/lib/wsrep
     wsrep_provider='/usr/lib/libgalera_smm.so'
     wsrep_provider_options='gcache.size=1G'
     wsrep_replicate_myisam=1
     wsrep_slave_threads=8
     wsrep_sst_method=xtrabackup
     wsrep_sst_auth=root:{password}
     ```

and this configuration on node 2

my.cnf

```
wsrep_cluster_address=gcomm://perc56a.<yourinternaladdress>.com,sqlquorum.<yourinternaladd
wsrep_cluster_name=<yourclustername, for example perc_mygreatcluster>
wsrep_node_name=perc56a.<yourinternaladdress>.com
wsrep_data_home_dir=/var/lib/wsrep
wsrep_provider='/usr/lib/libgalera_smm.so'
wsrep_provider_options='gcache.size=1G'
wsrep_replicate_myisam=1
wsrep_slave_threads=8
wsrep_sst_method=xtrabackup
wsrep_sst_auth=root:{password}
```

- binlog_format = ROW (this is the default setup)
- data dir on external disk

  my.cnf

  ```
  datadir=/srv/mysql
  ```

- remove the following parameters:

  my.cnf

  ```
  #innodb_locks_unsafe_for_binlog=1
  #table_cache=1M
  ```

5. Copy `debian.cfg` to the root on node 2. Use the following command:

   ```
   scp /etc/mysql/debian.cnf eit@perc56b
   ```

6. Create a **clustercheck** user in MySQL for monitoring purposes.

   ```
   GRANT PROCESS ON *.* TO 'clustercheckuser'@'localhost' IDENTIFIED BY 'clustercheckpassword!';
   ```

7. Install the following necessary tools:

   ```
   apt-get install haproxy xinetd httpcheck keepalived ntpdate
   ```

   - haproxy (load balancer) on nodes 1 and 2. To access haproxy statistics, use the following address pattern:

     ```
     http://perc56.<yourinternaladdress>.com/haproxy/stats
     ```

   - xinetd on port 9200 (ENABLED = 1)
   - httpcheck, which checks if the server is running
   - keepalived (failover) on nodes 1 and 2
   - ntpdate to synchronize the time with an ntp server, set this parameter in /etc/default/ntpdate:

     ```
     NTPSERVERS=<address of your ntp server>
     ```

8. Configure DHCP (DNS), to be able to use URLs to access nodes
9. To perform an operation specifically on the slave data source, use the following example code:

   ```
   public static void doOnSlaveDataSource(final Runnable action)
           {
                   final Tenant tenant = Registry.getCurrentTenantNoFallback();
                   ((AbstractTenant) tenant).cancelForceMasterMode();
                   tenant.activateSlaveDataSource();
                   try
   ```

```
                {
                        requireNonNull(action).run();
                }
                finally
                {
                        Registry.getCurrentTenantNoFallback().deactivateAlternativeDataSource();
                }
        }
```

Related Information

http://www.percona.com/doc/percona-xtradb-cluster/5.6/howtos/virt_sandbox.html

HSQLDB

HSQLDB is a light-weight SQL database that can be run within a Java Virtual Machine. It is useful for development environments due to its speed. SAP Commerce comes pre-bundled with a compatible version of HSQLDB.

Installation

No installation necessary, HSQLDB comes pre-bundled with SAP Commerce.

Configuration Overview

The following table contains a list of all HSQLDB-specific configuration properties. Please refer to Configuring the Behavior of SAP Commerce for a full list of all properties.

Property name	Property description
`db.url`	Specifies a path to access the database, such as `jdbc:hsqldb:file:/mydb;hsqldb.cache_scale=16;shutdown=true`.
`db.driver`	The fully qualified name of the JDBC driver class used to access the database. The `project.properties` file default is **org.hsqldb.jdbcDriver**.
`db.username`	The name of the user account used to access the database. On production systems, we recommend using a database system different from HSQLDB.
`db.password`	The password matching the database user account specified in `db.username`
`db.tableprefix`	An optional String that precedes a set of table names in a database schema. Specifying different table prefixes allows you to create different logical sets of tables within one schema, and thus running several instances of SAP Commerce on one single database schema.
`hsqldb.usecachedtables`	Please refer to http://hsqldb.sourceforge.net/doc/guide/ch01.html#N1023C for additional information on cached tables.
`db.mapping.char.legacy`	With the `db.mapping.char.legacy` property set by default to `true` (legacy), the char values are mapped to integer/byte values. By setting this property to `false` (disabling the legacy flag), you force the conversion of column types to **char** values. The reason the default value is `true` is because only this setting works correctly on already existing databases, on system update, without additional, manual work. If you disable the legacy mode for system update, you need to first convert the existing columns types (for example from int to char) manually.

Example Configuration for HSQLDB Database

```
db.url=jdbc:hsqldb:file:<at:var at:name="EXPLODED_EAR" />/mydb;hsqldb.cache_scale=16;shutdown=true
                         db.driver=org.hsqldb.jdbcDriver
                         db.username=sa
                         db.password=
                         db.tableprefix=
                         hsqldb.usecachedtables=false
```

If you want to connect to an HSQLDB server, try this database URL setting:

```
db.url=jdbc:wrapper:hybris:org.hsqldb.jdbcDriver:jdbc:hsqldb:hsql://localhost:9000/myinstance
```

Related Information

http://hsqldb.org/

Configuring the Behavior of SAP Commerce

Oracle

Oracle Database is a common enterprise database system, available for a variety of operating systems.

For a list of the Oracle database versions that are supported, see System Requirements.

Install Oracle as specified by the Oracle documentation .

Oracle Configuration

SAP Commerce works well in combination with the default Oracle configuration. The following notes are recommendations, but the best settings can vary depending on used hardware, on expected database size or load.

- SAP Commerce uses a configurable connection pool and hence does not open a lot of connections to the database. However it might be useful to raise the settings for `SESSIONS` and `PROCESSES` to approx 1000 to make sure that you are not running out of connections especially you have a cluster with multiple nodes.
- For Oracle 10 / 11, we recommend using the Automated Shared Memory Management ASMM .
- If you want to set the memory manually, make sure that the Buffer cache has at least 60% of total SGA.
- We do not make use of special Oracle features like XML or Java in database. We do not need a high `Java` or `Large Pool`.

  Example configuration if having an SGA of approx. 1.5 GB (small server)

SGA-Component	Current (MB)
Shared Pool	152
Buffer Cache	1232
Large Pool	8
Java Pool	8
Other	8

- For normal sized systems you do not need to create own tablespaces.

  Create a user account with `RESOURCE` and `CONNECT` privileges that writes to the default `USERS` tablespace.

ℹ Note

Do not use high-level system administration user accounts

Do not use the **SYS, SYSTEM, or the SYSMAN** database user accounts to run the SAP Commerce for security reasons. Use a non-privileged database user account instead.

- Make sure that there is a registered listener with TCP (defaults to port 1521) that can be accessed from outside. This is done by the default listener configuration.

Configuring Transparent Application Failover

Transparent Application Failover (TAF) is a feature of the Java Database Connectivity (JDBC) Oracle Call Interface (OCI) driver. It enables the application to automatically reconnect to a database, if the database instance to which the connection is made fails. In this case, the active transactions roll back.

When an instance to which a connection is established fails or is shut down, the connection on the client side becomes stale and throws exceptions to the caller trying to use it. TAF enables the application to transparently reconnect to a preconfigured secondary instance creating a fresh connection, but identical to the connection that was established on the first original instance. That is, the connection properties are the same as that of the earlier connection. This is true regardless of how the connection was lost.

Configuring OCI Drivers

Context

To configure an OCI driver:

Procedure

1. Install Oracle Client 11.2.0.3 or later.
2. Configure the **TNS** name:

   tnsnames.ora

   ```
   HYBRIS =
      (DESCRIPTION =
         (ADDRESS = (PROTOCOL = TCP)(HOST = mynode1)(PORT = 1521))
         (CONNECT_DATA =
         (SERVER = DEDICATED)
         (SERVICE_NAME = HYBRIS)
      )
      )
   ```

3. Modify hybris user profile and log in again.

   .bash_profile

   ```
   export ORACLE_HOME=/appli/oracle/product/11.2.0/client
                                    export LD_LIBRARY_PATH=/appli/oracle/product/11.2.0/client/lib
   ```

4. Check that the hybris user has read access to Oracle client.
5. Add Oracle library path to `wrapper.conf`:

   wrapper.conf

   ```
   wrapper.java.library.path.2=/appli/oracle/product/11.2.0/client/lib
   ```

6. Check the relevant Oracle driver library version in the Tomcat library, for example:

   ```
   <hybris_home>/bin/platform/lib/dbdriver/ojdbc6-11.2.0.2.0.jar
   ```

7. Add the db connection url to the `local.properties` file:

   ```
   db.url=jdbc:oracle:oci:@HYBRIS
   db.driver=oracle.jdbc.driver.OracleDriver
   db.username=
   db.password=
   oracle.statementcachesize=0
   db.pool.maxActive=90
   db.pool.maxIdle=90
   ```

8. Disable the automatic restart when connection error in the `local.properties` file:

   ```
   tenant.restart.on.connection.error=false
   ```

9. Build and restart:

   ```
   ant build all
   ```

10. Check startup logs.

Configuring TAF on the Database Server

Context

To configure TAF on the database server:

Procedure

1. Change TNS name to use TAF:

   tnsnames.ora

   ```
   HYBRIS =
      (DESCRIPTION_LIST =
         (FAILOVER = true)
         (LOAD_BALANCE = true)
         (DESCRIPTION =
            (ADDRESS = (PROTOCOL = TCP)(HOST = mynode1)(PORT = 1521))
            (ADDRESS = (PROTOCOL = TCP)(HOST = mynode2)(PORT = 1521))
            (CONNECT_DATA =
               (SERVER = DEDICATED)
               (SERVICE_NAME = HYBRIS)
               (FAILOVER_MODE =
                  (TYPE=select)
                  (METHOD=preconnect)
                  (RETRIES=20)
                  (DELAY=3)
            )
         )
      )
   )
   ```

2. Restart and test TAF:

   When we tested this configuration, TAF worked fine but all the connections were created on the first node at the startup (no load-balancing). A workaround is to switch node 1 and node 2 order in the ***tnsname.ora*** of each application server.

SAP Commerce Configuration Properties

See a list of all Oracle DB-specific configuration properties.

Property name	Property description

`db.url`	Specifies a path to access the database, such as `jdbc:oracle:thin:@oracle_server:1521:hyb`
	
	Note
	When using Oracle RAC, you have to specify every single node of the Oracle RAC cluster explicitly. Nod
	
	The following `project.properties` sample entry gives an example on this. Every single $Node entr
	
	`db.url=jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=ON)(FAILOVER=ON)(ENAB`
	`(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(Host=$Node_1)(Port=1521)) ... (ADDR`
	`(CONNECT_DATA=(SERVICE_NAME=$ORACLE_SERVICE_NAME)))`
	
	Or, for the oci driver
	
	`db.url=jdbc:oracle:oci:@(DESCRIPTION=(LOAD_BALANCE=ON)(FAILOVER=ON)(ENABL`
	`(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(Host=$Node_1)(Port=1521)) ... (ADDR`
	`(CONNECT_DATA=(SERVICE_NAME=$ORACLE_SERVICE_NAME)(FAILOVER_MODE=(TYPE=SEL`

`db.driver`	The fully qualified name of the JDBC driver class used to access the database. The `project.propert`
`db.username`	The name of the user account used to access the database. **Note** Do not use high-level system administration user accounts. Do not use the **SYS**, **SYSTEM**, or the **SYSMAN** database user accounts to run SAP Commerce for secu instead.
`db.password`	The password matching the database user account specified in `db.username`.
`db.tableprefix`	An optional String that precedes a set of table names in a database schema. Specifying different table p one schema, and thus running several instances of SAP Commerce on one single database schema.
`oracle.statementcachesize`	This is the number of statements to store in the cache for each connection. Do not set any other value th update.
`oracle.dataTS`	Allows you to specify a tablespace for data files. By not specifying this property (`project.propertie` and thus the Oracle database uses its default value.
`oracle.indexTS`	Allows you to specify a tablespace for index files. By not specifying this property (`project.propertie` and thus the Oracle database uses its default value.
`db.mapping.char.legacy`	With the `db.mapping.char.legacy` property set by default to `true` (legacy), the char values are ma (disabling the legacy flag), you force the conversion of column types to **char** values. The reason the defa already existing databases, on system update, without additional, manual work. If you disable the legacy columns types (for example from int to char) manually.

For a full list of all SAP Commerce properties, see the global `project.properties` file in *<HYBRIS_BIN_DIR>*`/platform`.

Related Information

Configuring the Behavior of SAP Commerce
Third-Party Compatibility
Microsoft SQL Server
MySQL

Oracle

HSQLDB

Microsoft SQL Server

Microsoft SQL Server is a database system available for Microsoft Windows operating systems.

Installation

For supported Microsoft SQL Server versions see System Requirements.

Install Microsoft SQL Server as specified by the Microsoft documentation .

Configuration Overview

The following table contains a list of all Microsoft SQL Server-specific configuration properties.

Property name	Property description
`db.url`	Specifies the url to access the database. To check which driver version is supported with SAP Commerce, se Third-Party Compatibility. Example for using the Microsoft JDBC driver: `db.url=jdbc:sqlserver://<host>:1433;responseBuffering=adaptive;loginTimeout=1`
`db.driver`	The fully qualified name of the JDBC driver class used to access the database. `db.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver`
`db.username`	The name of the user account used to access the database. **Note** **Do not use high-level system administration user accounts** Do not use the **sa** database user account to run the SAP Commerce Suite for security reasons. Use a non-privileged database user account instead.
`db.password`	The password matching the database user account specified in **db.username**
`db.tableprefix`	An optional String that precedes a set of table names in a database schema. Specifying different table prefix allows you to create different logical sets of tables within one schema, and thus running several instances of SAP Commerce on one single database schema.
`#sqlserver.pkindextype`	Advanced setting. Do not edit the default value unless you are aware of the effects. Possible values: **CLUSTERED** and **NONCLUSTERED**. As a rule of thumb, every table should have a clustered index. Generally, but not always, the clustered index should be on a column that monotonically increases - su as the primary key column. However - SELECTs on tables with **CLUSTERED** primary keys lead to more database locking when reading data. Our tests on many systems and project experience has shown that usir **NONCLUSTERED** as pkindextype is the better choice. Default is **NONCLUSTERED**.

Note

Insufficient Isolation Might Cause Deadlocks on Microsoft SQL Server on Multi-Threaded Imports

If you use ImpEx to run multi-threaded imports, you may experience deadlocks on Microsoft SQL Server with **ALLOW_SNAPSHOT_ISOLATION OFF** and **READ_COMMITTED_SNAPSHOT OFF** on the database table. If so, try setting

ALLOW_SNAPSHOT_ISOLATION ON and **READ_COMMITTED_SNAPSHOT ON**, such as:

```
ALTER DATABASE hybris SET READ_COMMITTED_SNAPSHOT ON;
          ALTER DATABASE hybris SET ALLOW_SNAPSHOT_ISOLATION ON;
```

See also

- http://msdn.microsoft.com/en-us/library/ms175095.aspx

Server Settings

There are some steps to perform in addition to installing the MS SQL Server in order for it to function properly with the platform.

Create Database

You must create a custom database for the platform:

1. Start the Microsoft SQL Server Management Studio (Start > All Programs > Microsoft SQL Server > Microsoft SQL Server Management Studio).
2. Connect to the Database Engine using Windows Authentication.
3. Right-click on Databases > New Database...
4. In the window, set the owner to an existing account or the one created in the next precedure.

   SAP Commerce requires databases with **CI collations** due to queries treating table and column names case insensitively. Therefore please choose a CI collate when creating the database, like:

   - **Latin1_General_CI_AS:Latin1**: General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
   - **SQL_Latin1_General_CP1_CI_AS**: Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data

Create Account

You must assign a custom SQL Server account for the platform.

1. Start Microsoft SQL Server Management Studio (Start > All Programs > Microsoft SQL Server > Microsoft SQL Server Management Studio).
2. Connect to the Database Engine using Windows Authentication.
3. Go to Security > Logins.
4. Right-click on Logins -> New Login...
5. In the window, use a custom login name.
6. Use an SQL Server authentication.
7. Specify a password.
8. Set default database to the database created in the previous procedure.

Enable SQL Server Login

By default the login to SQL Server is only possible with Windows user accounts. You must change this to also accept SQL users.

1. Start Microsoft SQL Server Management Studio (Start > All Programs > Microsoft SQL Server > Microsoft SQL Server Management Studio).
2. Connect to the Database Engine using Windows Authentication.

3. Right-click on the top-level (database instance) and select **Properties.**
4. Go to Security.
5. Change the Server authentication to SQL Server and Windows Authentication mode.

Enable TCP/IP protocol

It is important to enable the TCP/IP protocol in the MS SQL Server because the platform uses it to connect to the server.

1. Start SQL Server Configuration Manager (Start > All Programs > Microsoft SQL Server > Configuration Tools > SQL Server Configuration Manager).
2. Go to SQL Server Network Configuration > Protocols for ***<instance>.***
3. Right-click on TCP/IP > Enable.
4. Start Microsoft SQL Server Management Studio (Start > All Programs > Microsoft SQL Server > Microsoft SQL Server Management Studio).
5. Connect to the Database Engine using Windows Authentication.
6. Right-click on root of the tree > Restart.

Listening Port

To find out on which port the MS SQL Server is listening:

1. Start the Microsoft SQL Server Management Studio (Start > All Programs > Microsoft SQL Server > Microsoft SQL Server Management Studio).
2. Connect to the Database Engine using Windows Authentication.
3. Go to Management > SQL Server Logs.
4. Open the current log by double-clicking it.
5. Search the log for "listening on". An output line similar to "Server is listening on['any' ip4 port] will be displayed.

→ Tip

You can configure a custom static port; for more information, see the Microsoft documentation at http://support.microsoft.com/kb/823938 .

A helpful website to troubleshoot why a connection it not working: http://social.technet.microsoft.com/wiki/contents/articles/2102.how-to-troubleshoot-connecting-to-the-sql-server-database-engine.aspx .

Related Information

Configuring the Behavior of SAP Commerce

http://www.microsoft.com/sql/default.mspx

http://social.technet.microsoft.com/wiki/contents/articles/2102.how-to-troubleshoot-connecting-to-the-sql-server-database-engine.aspx

Microsoft SQL Server

MySQL

Oracle

HSQLDB

SAP HANA Database

SAP HANA is a relational, in-memory database, and the main component of the SAP HANA platform. To use SAP HANA, you need to configure the Platform by adding properties to your `local.properties` configuration file.

Installation

SAP HANA database is included in the SAP HANA Platform, which is deployable on-premise as an appliance, in the cloud, or as a hybrid of the two.

The supported SAP HANA version can be found within System Requirements.

For installation details, please refer to https://www.sap.com/products/hana/implementation/resources.html.

Configuration Overview

The following table contains a list of all SAP HANA DB - specific configuration properties. Please refer to Configuring the Behavior of SAP Commerce for a full list of all properties.

Property name	Property description
`db.url`	Specifies a path to access the database, such as `db.url=jdbc:sap://<host>:<port>?reconnect=true&statementCacheSize=<value>`
`db.driver`	The fully qualified name of the JDBC driver class used to access the database. The `project.properties` file default is `com.sap.db.jdbc.Driver`.
`db.username`	The name of the user account used to access the database.
`db.password`	The password matching the database user account specified in `db.username`
`db.mapping.char.legacy`	With the `db.mapping.char.legacy` property set by default to `true` (legacy), the char values are mapped to integer/byte values. By setting this property to `false` (disabling the legacy flag), you force the conversion of column types to **char** values. The reason the default value is `true` is because only this setting works correctly on already existing databases, on system update, without additional, manual work. If you disable the legacy mode for system update, you need to first convert the existing columns types (for example from int to char) manually.

<port>

Configure `<port>` by using the `3xxyy` pattern where `xx` is the SAP HANA instance number and `yy` is a two-digit number as follows:

- Port `3xx15` is for an SAP HANA single-container system.
- Port `3xx13` is for the system database of an SAP HANA multitenant database container.

The SQL port numbers for tenant databases are normally increased by increments of 3, starting from 41: `3xx41`, `3xx44`, or `3xx47`, and so on.

statementCacheSize

The `statementCacheSize` parameter defines the maximum number of SQL statements to be cached. Configure the `statementCacheSize` parameter as required. In SAP Commerce this parameter is by default set to `256`, `statementCacheSize=256`. When set to `0`, statement caching is off. It is also the minimum value. The maximum value is `1000`.

Example Configuration for SAP HANA Database:

```
db.url=jdbc:sap://10.8.26.8:30015/?reconnect=true
db.driver=com.sap.db.jdbc.Driver
db.username=user
db.password=password
```

Column and Row-Based Storage

Apart from the traditional row-based data storage, SAP HANA also allows the column-based style.

Note

Column-based style is recommended, because the SAP HANA Platform is optimized to work with it.

`True` is the default value. Setting the value to `False` changes the configuration to row-based style.

```
hanadb.storage.columnbased=true
```

DDL/DML Settings

It is possible to change the ddl and dml batch sizes:

```
bootstrap.init.type.system.ddl.sql.batch.size=
bootstrap.init.type.system.dml.sql.batch.size=
```

Note

It is strongly recommended to set the ddl batch size to `1`. Any other values may cause issues with dropping tables.

JDBC Driver

The SAP HANA jdbc driver (`hybris/bin/platform/lib/dbdriver/ngdbc-*.jar`) is delivered with platform and should be working out of the box.

Since 2.3.13, the SAP HANA JDBC driver is built with JDK 9 with a target of Java 7. If you face Java 9 - related problems, switch to SAP HANA JDBC driver 2.2.x .

SQL Client

The most recommended option is to use the latest SAP HANA Client (see https://tools.hana.ondemand.com/#hanatools).

Related Information

http://www.saphana.com
Configuring the Behavior of SAP Commerce
Third-Party Compatibility
Microsoft SQL Server
MySQL
Oracle
HSQLDB

Specifying HANA as Deployment Type for Single Items

In addition to setting the storage for all tables to HANA, you can add a special property for a specific item in order to change its deployment to HANA.

Introduction

The property `hanadb.storage.columnbased` changes the table type for all tables. However, you can also override this property for a specific deployment. To do this, add a custom property to the item type which holds the deployment declaration.

Examples

Column based deployment

```
<itemtype code="MyItem" autocreate="true" generate="false">
        <deployment table="MyDeployment" typecode="102938"/>
        <custom-properties>
        <property name="deployment.hana.tabletype"><value>"COLUMN"</value></property>
        </custom-properties>
        <attributes>
        ...
</itemtype>
```

Row based deployment

```
<itemtype code="MyItem" autocreate="true" generate="false">
        <deployment table="MyDeployment" typecode="102938"/>
        <custom-properties>
        <property name="deployment.hana.tabletype"><value>"ROW"</value></property>
        </custom-properties>
        <attributes>
        ...
</itemtype>
```

Verification

You can verify table types by executing the following statement on the HANA database

```
select * from
(
        select SCHEMA_NAME, TABLE_NAME, 'ROW' as TABLE_TYPE from M_RS_TABLES
        union
        select SCHEMA_NAME, TABLE_NAME, 'COLUMN' as TABLE_TYPE from M_CS_TABLES
) where SCHEMA_NAME='{YOUR_SCHEMA}' order by TABLE_NAME
```

Related Information

http://www.saphana.com

Configuring the Behavior of SAP Commerce

Third-Party Compatibility

Microsoft SQL Server

MySQL

Oracle

HSQLDB

JNDI

There are a few important JNDI-related topics you should be aware of.

Handling Connection Pool

SAP Commerce handles database connections from JNDI and non-JNDI sources in a unified way. In both cases SAP Commerce manages its own connection pool. Acquired connections are stored in the SAP Commerce pool and are not released unless there's a connection error. The JNDI pool can't have settings that are incompatible with the SAP Commerce pool. For example JNDI Pool cannot offer fewer connections than required by SAP Commerce. You must disable any JNDI advanced management features such as closing inactive connections because these connections are being managed by SAP Commerce.

Database Connection

When using `local.properties` for database connection configuration, the database password and other information is stored in plain text. This is insecure. Instead, use JDNI for database configuration.

Do the following to use a JNDI data source:

1. Define the data source for your application server. Check the instructions specified by the manufacturer of the application server.
   - JNDI Resources HOW-TO for Tomcat 7

   Note

   SAP Commerce does not support JNDI Datasource defined in the `context.xml` file.

   Note

   To make the JNDI configuration visible for Platform, `HybrisGlobalResourcesLifecycleListener` should be defined in the Tomcat configuration file, like it is configured in the configuration provided in the `config` folder under `config/tomcat/conf/server.xml`.
2. Configure SAP Commerce to use the JNDI data source
3. Configure the name of the JNDI data source using the `config` property in `local.properties`.

   ```
   db.pool.fromJNDI=<JNDI datasource name>
   ```
4. Call `ant deploy` and start the application server.

Parameters used during JNDI configuration are XMLENCODED, therefore you should encode all characters which are not allowed in XML but which are part of the configuration (passwords and URLs often contain characters which need to be escaped in XML).

Example: If a password contains an ampersand, encode it as `&` to avoid an XML parse exception.

Third-Party Web Servers

Some scenarios may require that no direct connection from the Internet be established to your application servers. To this end, you use web servers to act as reverse proxies to forward incoming requests to your application servers.

Read more in Web Server Proxy Setup.

Related Information

Third-Party Compatibility

Web Server Proxy Setup

Some scenarios may require that no direct connection from the internet be established to your application servers. To this end, you use web servers to act as reverse proxies to forward incoming requests to your application servers.

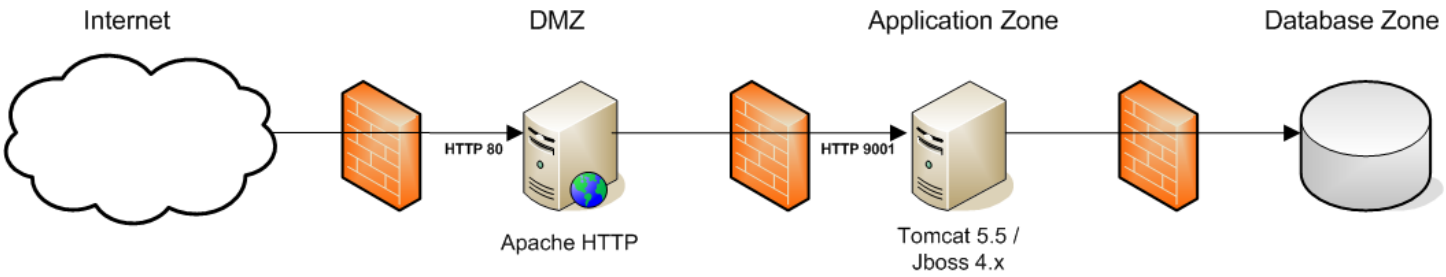
You can either use HTTP or AJP13 as the protocol between the web servers and the application servers, below you find setup instruction for both HTTP and AJP13.

Protocol	Performance	Transparency
AJP13	+	−
HTTP	−	+

Setting up an HTTP Proxy

Requirements:

- Apache HTTP Server 1.3.x / 2.x
  - proxy_module
  - proxy_http_module
- Platform using Tomcat



Apache 1.3/2.x supports an optional module (mod_proxy) that configures the web server to act as a proxy server. This can be used to forward requests for a particular web application to a Tomcat instance, without having to configure a web connector such as mod_jk. To accomplish this, you need to perform the following tasks. This example will forward all request to the context path /webmc of your apache server to the webmc of an SAP Commerce E-Business Platform running on the same machine on port 9001.

1. Configure your copy of Apache so that it includes the **mod_proxy** and **proxy_http_module** modules. If you are building from source, the easiest way to do this is to include the --enable-module=proxy directive on the ./configure command line.
2. If not already added for you, make sure that you are loading the **mod_proxy** and the **proxy_http_module** module at Apache startup time, by using the following directives in your `httpd.conf` file:

```
LoadModule proxy_module  modules/mod_proxy.so
                                AddModule  mod_proxy.c #only needed for Apache 1.3.x
                                LoadModule proxy_http_module  modules/mod_proxy_http.so
                                AddModule  mod_http_proxy.c #only needed for Apache 1.3.x
```

3. Include two directives in your `httpd.conf` file for each web application that you wish to forward to Tomcat. For example, to forward an application at context path /webmc:

```
ProxyPass         /webmc  http://localhost:9001/webmc
                                ProxyPassReverse  /webmc  http://localhost:9001/webmc
```

   which tells Apache to forward URLs of the form `http://localhost/webmc/*` to the Tomcat connector listening on port 9001.

4. Configure your copy of Tomcat to include a special <Connector> element, with appropriate proxy settings in the `server.xml` of your Tomcat instance, for example:

```
<Connector port="9001" ...
                                proxyName="www.mycompany.com"
                                proxyPort="80"/>
```

which will cause servlets inside this web application to think that all proxied requests were directed to www.mycompany.com on port 80.

5. It is legal to omit the proxyName attribute from the <Connector> element. If you do so, the value returned by request.getServerName() will by the host name on which Tomcat is running. In the example above, it would be localhost.
6. If you also have a <Connector> listening on port another port (e.g. 8080) (nested within the same Service element), the requests to either port will share the same set of virtual hosts and web applications.
7. You might wish to use the IP filtering features of your operating system to restrict connections to port 9001 (in this example) to be allowed only from the server that is running Apache.
8. When requests are proxied by Apache, the web server will be recording these requests in its access log. Therefore, you will generally want to disable any access logging performed by Tomcat itself.

When requests are proxied in this manner, all requests for the configured web applications will be processed by Tomcat (including requests for static content). You can improve performance by using the mod_jk web connector instead of mod_proxy. mod_jk can be configured so that the web server serves static content that is not processed by filters or security constraints defined within the web application's deployment descriptor (/WEB-INF/web.xml).

ℹ Note

Alternative Configuration

Alternatively, you can set up a series of web applications that are only available via proxying, as follows:

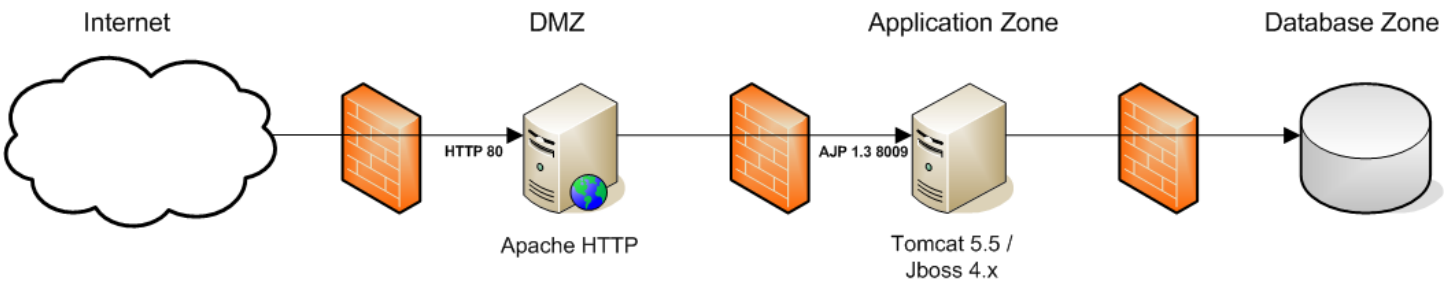
- Configure another <Service> that contains only a <Connector> for the proxy port.
- Configure appropriate Engine, Host, and Context elements for the virtual hosts and web applications accessible via proxying.
- Optionally, protect port 8081 with IP filters as described earlier.

Setting Up an AJP 1.3 Proxy

The AJP13 protocol is packet-oriented. A binary format was presumably chosen over the more readable plain text for reasons of performance. The web server communicates with the servlet container over TCP connections. To cut down on the expensive process of socket creation, the web server will attempt to maintain persistent TCP connections to the servlet container, and to reuse a connection for multiple request/response cycles.

Requirements:

- Apache HTTP Server 1.3.x / 2.x
  - proxy_module
  - proxy_ajp_module
- E-Business Platform using Tomcat



Apache 1.3/2.x supports an optional module (mod_proxy) that configures the web server to act as a proxy server. This can be used to forward requests for a particular web application to a Tomcat instance, without having to configure a web connector such as mod_jk. To accomplish this, you need to perform the following tasks. This example will forward all request to the context path

/webmc of your apache server to the webmc of an SAP Commerce E-Business Platform running on the same machine on port 9001.

1. Configure your copy of Apache so that it includes the **mod_proxy** and **mod_ajp_proxy** modules. If you are building from source, the easiest way to do this is to include the --enable-module=proxy directive on the ./configure command line.
2. If not already added for you, make sure that you are loading the **mod_proxy** and the **mod_http_proxy** module at Apache startup time, by using the following directives in your **httpd.conf** file:

   ```
   LoadModule proxy_module  modules/mod_proxy.so
                                    AddModule  mod_proxy.c #only needed for Apache 1.3.x
                                    LoadModule proxy_ajp_module  modules/mod_proxy_ajp.so
                                    AddModule  mod_ajp_proxy.c #only needed for Apache 1.3.x
   ```

3. Include two directives in your **httpd.conf** file for each web application that you wish to forward to Tomcat. For example, to forward an application at context path /webmc:

   ```
   ProxyPass         /webmc  ajp://localhost:8009/webmc
                                    ProxyPassReverse  /webmc  ajp://localhost:8009/webmc
   ```

   which tells Apache to forward URLs of the form `http://localhost/webmc/*` to the Tomcat listening on port 9001.
4. Configure your copy of Tomcat to include a special <Connector> element, with appropriate proxy settings in the `server.xml` of your Tomcat instance, for example:

   ```
   <Connector port="8009"
                                    enableLookups="false"
                                    redirectPort="8443"
                                    proxyName="www.mycompany.com"
                                    proxyPort="80"
                                    protocol="AJP/1.3"/>
   ```

   which will cause servlets inside this web application to think that all proxied requests were directed to www.mycompany.com on port 80.
5. It is legal to omit the proxyName attribute from the <Connector> element. If you do so, the value returned by request.getServerName() will by the host name on which Tomcat is running. In the example above, it would be localhost.
6. If you also have a <Connector> listening on port on port another port (e.g. 8080) (nested within the same Service element), the requests to either port will share the same set of virtual hosts and web applications.
7. You might wish to use the IP filtering features of your operating system to restrict connections to port 9001 (in this example) to be allowed only from the server that is running Apache.
8. When requests are proxied by Apache, the web server will be recording these requests in its access log. Therefore, you will generally want to disable any access logging performed by Tomcat itself.

When requests are proxied in this manner, all requests for the configured web applications will be processed by Tomcat (including requests for static content). You can improve performance by using the mod_jk web connector instead of mod_proxy. mod_jk can be configured so that the web server serves static content that is not processed by filters or security constraints defined within the web application's deployment descriptor (/WEB-INF/web.xml).

ℹ Note

Alternative Configuration

Alternatively, you can set up a series of web applications that are only available via proxying, as follows:

- Configure another <Service> that contains only a <Connector> for the proxy port.
- Configure appropriate Engine, Host, and Context elements for the virtual hosts and web applications accessible via proxying.
- Optionally, protect port 8081 with IP filters as described earlier.

Related Information

http://tomcat.apache.org/tomcat-6.0-doc/proxy-howto.html
http://tomcat.apache.org/tomcat-6.0-doc/config/http.html
http://tomcat.apache.org/tomcat-6.0-doc/config/ajp.html

Setting up SAP JVM

Perform some additional steps to set up SAP JVM to run correctly with SAP Commerce, and avoid potential errors.

Prerequisites

- Install and configure SAP JVM
- Install and configure svn, ant, wget, unzip

Context

SAP JVM is a certified Java Virtual Machine (JVM) and Java Development Kit (JDK), compliant to the Java Standard Editions 1.4.2, 5, 6, and 7. SAP JVM does not come with a default javascript engine implementation for JSR 223. Attempting to run it without a JavaScript engine produces test failures such as the following:

```
java.lang.IllegalStateException: No such engine identified by: DefaultScriptEngineType{scriptName='jav
                        at de.hybris.platform.scripting.engine.impl.DefaultScriptingLanguagesService.c
                        at de.hybris.platform.scripting.engine.impl.DefaultScriptingLanguagesService.g
                        at de.hybris.platform.scripting.engine.impl.DefaultScriptingLanguagesService.b
                        at de.hybris.platform.scripting.engine.impl.DefaultScriptingLanguagesService.g
```

The solution is to put a JSR 223-compliant javascript engine implementation on the classpath. The following steps will guide you through this process on a linux operating system:

Procedure

1. Download the latest version of Rhino JavaScript Engine.

   ```
   wget https://github.com/mozilla/rhino/releases/download/Rhino1_7_9_Release/rhino-1.7.9.zip
   unzip rhino1.7.9.zip
   ```

2. Get the sources of the JSR 223 adapter for Rhino.

   ```
   svn checkout https://svn.java.net/svn/scripting~svn/trunk
   ```

3. Build the adapter.

   ```
   cp rhino1_7R4/js.jar trunk/engines/javascript/lib/js.jar
   ant -f trunk/engines/javascript/make/build.xml clean all
   ```

4. Copy the adapter and Rhino to the platform.

   ```
   mkdir -p ${HYBRIS_BIN_DIR}/platform/ext/scripting/lib
   cp trunk/engines/javascript/build/js-engine.jar ${HYBRIS_BIN_DIR}/platform/ext/scripting/lib/
   cp trunk/engines/javascript/lib/js.jar ${HYBRIS_BIN_DIR}/platform/ext/scripting/lib/
   ```

Setting Up a Production System

For demonstration and development you will typically install SAP Commerce locally, but for production systems you may have a distributed system to set up.

Related Information

Deploying New SAP Commerce Software Versions
Deployment of SAP Commerce Server

Deployment of SAP Commerce Server

SAP Commerce offers an easy way of deploying to a production server. You create deployment ZIP files which contain everything needed to deploy your system.

SAP Commerce Server is based on Apache Tomcat, which is an open source implementation of the Java Servlet and JavaServer Pages technologies.

The deployment concept for this web application technology is based on WAR (Web archive) files, which consist of one web application per WAR file. This is not sufficient for a deployment of a SAP Commerce system, as a SAP Commerce system consists of the core application with all its configured extensions and a number of web applications.

To fully understand the process for deploying SAP Commerce, you should learn about:

- Creating Deployment ZIP Files: Describes the process of creating the deployment ZIP files which you use for installation of a new SAP Commerce system or update a running SAP Commerce system.
- Installing Deployment ZIP Files: Describes how to install created deployment ZIP files to a clean system.
- Updating Deployment ZIP Files: Describes how to update a running system which was already configured.

Related Information

Download

Creating Deployment ZIP Files

You can create deployment ZIP files for easily deploying your SAP Commerce to the SAP Commerce Server (preconfigured Tomcat).

Prerequisites

In order to be able to create the deployment ZIP files, you need a pre-configured SAP Commerce system (for example your development, test, or integration system) which has all the extensions configured. These extensions are part of the deployment ZIP files.

The deployment ZIP files are created by the ant task **production**. When you execute `ant production` in the `platform` directory of your SAP Commerce system, two deployment ZIP files are created. These ZIP files do not contain any configuration details, they just contain the binaries of your SAP Commerce system:

- `hybrisServer-Platform.zip`:

  This ZIP file contains the complete `platform` directory including the embedded SAP Commerce Server.

- `hybrisServer-AllExtensions.zip`:

  This ZIP file contains all the configured extensions of your SAP Commerce system.

Context

To create deployment ZIP files:

Procedure

1. Open a Terminal/Command Prompt window.
2. Navigate to the Platform directory; for example: `cd <hybris installation directory>/bin/platform`
3. Set up the `ant` environment.
   - For Windows systems: `setantenv.bat`
   - For Linux/Unix systems: `./setantenv.sh`
4. Execute `ant production`.

   After the ant production task finishes, you find the resulting ZIP file under `HYBRIS_TEMP/hybris/hybrisServer` (see Environment Variables for details). The absolute path is printed out in the last line of the ant production task output in your console.

```
ant production
[echo] ---------------------------------------------------------------
     [echo] hybris Platform Environment
     [echo] ---------------------------------------------------------------
     [echo] OS family:                 unix
     [echo] Java platform:             Java(TM) SE Runtime Environment, 11.0.3+12-LTS
     [echo] Java compiler:             org.eclipse.jdt.core.JDTCompilerAdapter
     [echo] Build target:              11
     [echo] Compilation mode:          strict
     [echo] hybris Platform directory: /Users/i318523/Downloads/maj17/hybris/bin/platform
     [echo] hybris Platform version:   1905.0
     [echo] Ant version:               Apache Ant(TM) version 1.10.5 compiled on July 10 2018
     [echo] Ant memory settings:       -Xmx2g -Dfile.encoding=UTF-8 -Djdk.util.jar.enableMultiRe
     [echo] ---------------------------------------------------------------
     [echo] Extension file used: /Users/i318523/Downloads/maj17/hybris/bin/platform/extensions.x
     [echo] Platform home: /Users/i318523/Downloads/maj17/hybris/bin/platform
     [echo] Log folder: /Users/i318523/Downloads/maj17/hybris/log
     [echo] Data folder: /Users/i318523/Downloads/maj17/hybris/data
     [echo] Config folder: /Users/i318523/Downloads/maj17/hybris/config
     [echo] Bin folder: /Users/i318523/Downloads/maj17/hybris/bin
     [echo] Temp folder: /Users/i318523/Downloads/maj17/hybris/temp/hybris
     [echo] Boostrap bin folder: /Users/i318523/Downloads/maj17/hybris/bin/platform/bootstrap/bi
     [echo] Roles folder: /Users/i318523/Downloads/maj17/hybris/bin/platform/../../roles
     [echo] ---------------------------------------------------------------
...
production:
     [echo] Building hybris Server Production Zips to '/Users/i318523/Downloads/maj17/hybris/tem
     [echo] Validation of *-items.xml files is enabled. It can be disabled by setting build.deve
     [echo] Validation - Step 1: Deployments
     [echo] checking file: /Users/i318523/Downloads/maj17/hybris/bin/platform/ext/core/resources
...
[ycodegenerator] 13:26:39,664 [main] INFO  CodeGenerator - Starting code generation ...
[ycodegenerator] 13:26:40,056 [main] INFO  CodeGenerator -    No changes found, skipping code ge
[ycodegenerator] 13:26:40,057 [main] INFO  CodeGenerator - Code generation done in 392 ms.
     [echo] preparing...
     [echo] building extension 'core'...
     [echo] building extension 'paymentstandard'...
...
 [echo] ************************************************
     [echo] - copy_extension_for_hybris_server
     [echo] ------------------------------------------------
     [echo] - srcdir:   /Users/i318523/Downloads/maj17/hybris/bin/modules/platform/yempty
     [echo] - destdir:  /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/bin/modul
     [echo] - extname:  yempty
     [echo] ************************************************
     [echo]
     [copy] Copying 40 files to /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/b
     [copy] Copied 16 empty directories to 1 empty directory under /Users/i318523/Downloads/maj1
    [mkdir] Created dir: /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/bin/modu
      [jar] Building jar: /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/bin/mod
    [mkdir] Created dir: /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/tomcat
      [zip] Building zip: /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/hybrisS
      [zip] Building zip: /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/hybrisS
      [zip] Building zip: /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/hybrisS
      [zip] Building zip: /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/hybrisS
     [copy] Copying 1 file to /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer
```

```
     [delete] Deleting directory /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/bi
     [delete] Deleting directory /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/co
     [delete] Deleting directory /Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/to
       [echo] Finished creation of '/Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer

BUILD SUCCESSFUL
Total time: 26 seconds



    [echo] Finished creation of '/Users/i318523/Downloads/maj17/hybris/temp/hybris/hybrisServer/hyb
```

Related Information

Download

Installing Deployment ZIP Files

You can install an SAP Commerce system on a production server using the deployment ZIP files created by **ant production**.

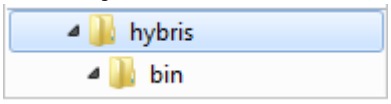
Prerequisites

The process of creating deployment ZIP files is described in Creating Deployment ZIP Files.

First you have to make sure that your production system complies to the system requirements which are described in Third-Party Compatibility.

Before you begin with the installation, make sure that you are familiar with the SAP Commerce directory structure. You can find detailed information in SAP Commerce Directory Structure.

Procedure

1. Copy the deployment ZIP files that you want to install to the production system.
2. Extract deployment ZIP files. Extract both ZIP files (`hybrisServer-Platform.zip` and `hybrisServer-AllExtensions.zip`) to the directory the SAP Commerce system is installed in. This directory is referred to as the **SAP Commerce installation directory**. As both ZIP files contain the common root directory `bin` you should end up with a directory structure like this: 

3. Configure SAP Commerce System. If this is the first time your installing the SAP Commerce system on your production system, you need to either copy the configuration which should be used from another system or configure it starting with one of the two configuration templates.

Configuring Using the Configuration of Your Development System

Procedure

1. Zip your `config` directory on your development system. As the deployment ZIP files intentionally do not contain any configuration settings, you need to create a ZIP file of the `config` directory of the SAP Commerce system you used to create the deployment ZIP files before.
2. Copy the ZIP file to your production system.
3. Extract the ZIP file on your production system.

4. Extract the configuration ZIP file to your SAP Commerce installation directory. In the end, the configuration files have to be located under: <SAP Commerce `installation directory>/config`

Configuring from Scratch Using a Configuration Template

Procedure

1. Open a command shell.
2. Go to the SAP Commerce `platform` directory: `cd <SAP Commerce installation directory>/bin/platform.`
3. Set up the ant environment:
   - For Windows systems: `setantenv.bat`
   - For Linux/Unix systems: `./setantenv.sh`
4. Execute ant.

   When you run ant for the first time, the SAP Commerce build system detects that there is no configuration directory. Therefore, it asks you which configuration template you want to use. You can find detailed information on configuration templates (see Configuration Templates). Select a configuration template and ant creates a configuration directory in <SAP Commerce `installation directory>/config` and copy over the needed configuration files.

```
C:\hybris\bin\platform>ant
Buildfile: build.xml
    [echo]
    [input]
    [input]  **** NO CONFIG FOLDER FOUND ****
    [input]
    [input]  No config folder was found at C:\hybris\config.
    [input]  A "fresh" folder containing basic configuration files and the SAP Commerces
    [input]  demo license will be created for your convenience.
    [input]  Please adjust and review the configuration files (and license) and
    [input]  call 'ant' again. This directory will never be overridden or
    [input]  touched again. Always use this configuration folder for configuration
    [input]  of platform, do not change anything within the platform folder.
    [input]
    [input]  Please choose the configuration template.
    [input]  Press [Enter] to use the default value ([develop], production)
```

Adjusting the Configuration

Now you do all the adjustments to the copied configuration in order to match the production system environment. This may include:

Configuration file	Description	Documentation
<SAP Commerce installation directory>/config/ `localextensions.xml`	Extension Configuration	Installation Based on Specified Extensions
<SAP Commerce installation directory>/config/ `local.properties`	All other configuration aspects, for example: Database, JVM, Port and so on.	Configuration Properties

Building SAP Commerce System

Procedure

1. Open a command prompt.
2. Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform` directory.
3. Make sure that a compliant version is used.
    a. On the Windows operating system, call the *<${HYBRIS_BIN_DIR}>* `/platform/setantenv.bat` file. Do not close the command prompt after this call as the settings are transient and would get lost if the command prompt is closed.
    b. On the Unix operating system, call the *<${HYBRIS_BIN_DIR}>* `/platform/setantenv.sh` file, such as: `. ./setantenv.sh`.
4. Call `ant all` to build the entire SAP Commerce solution.

Additional Steps

To automatically start a SAP Commerce system at system boot time, SAP Commerce Server uses the Tanuki Java Service Wrapper. In order to start your SAP Commerce system during startup of your production system, you need to follow the scripts described here:

- Linux / Unix systems
- Windows systems

Updating Deployment ZIP Files

You may update your SAP Commerce installation using deployment ZIP files. This involves shutting down your current system, removing the old binaries, extracting the new files, then rebuilding the system.

Shut Down the SAP Commerce System

Make sure that you shut down the SAP Commerce system before you proceed with the update process.

Remove Old Binaries

First you have to determine if the new deployment ZIP files contain a new SAP Commerce version or not. If you need to deploy a new Platform version you have to delete some folders according to the table below:

Description	Directory to be deleted
`hybrisServer-Platform.zip` contains a new Platform version.	*<installation directory>*`/bin/platform`
`hybrisServer-AllExtensions.zip` contains new extension versions.	• *<installation directory>*`/bin/modules` • *< installation directory>*`/bin/custom` • *<installation directory>*`/bin/ext-partner`

Make sure that you extract the deployment ZIP files to the right directory. Both ZIP files contain the `bin` directory, therefore you need to extract them to the SAP Commerce installation directory.

Extract Deployment ZIP Files

Depending on which directories you have deleted in the step before, you now need to extract one or both deployment ZIP files.

Deployment ZIP File	Description
`hybrisServer-Platform.zip`	Extract this ZIP file, if you need to deploy a new SAP Commerce version.
`hybrisServer-AllExtensions.zip`	Extract this ZIP file, if you need to deploy new extension versions.

Build SAP Commerce System

1. Open a command prompt.
2. Navigate to the `<HYBRIS_BIN_DIR>/platform` directory.
3. Make sure that a compliant version is used:
    - On the Windows operating system, call the `<HYBRIS_BIN_DIR>/platform/setantenv.bat` file. Do not close the command prompt after this call as the settings are transient and would get lost if the command prompt is closed.
    - On the Unix operating system, call the `<HYBRIS_BIN_DIR>/platform/setantenv.sh` file, for example `. ./setantenv.sh`.
4. Call `ant clean all` to build the entire SAP Commerce solution.

Ant Production Improvements

SAP Commerce provides several ant production flags that provide improvements or new features you may use with ant production for generating production packages.

For details see:

- Running SAP Commerce Server on Production Environment without Calling Any Ant Target
- Creating the Production Package without Tomcat Application Server
- Validating Packages Created by Ant Production Target
- Externalizing Generated Classes from a Production Package
- Creating Separate Tomcat Configuration Files for Extensions with Web Module
- How Old and New Ant Production Related Flags Work Together

Running SAP Commerce Server on Production Environment without Calling Any Ant Target

production.legacy.mode flag

To create an ant production zip package that can be unpacked and run without the need to execute any ant target on production environment, you can use the boolean flag `production.legacy.mode` (defaults to true).

When set to false, all SAP Commerce paths in `server*.xml` files and `wrapper*.conf` files are PLATFORM_HOME relative, where PLATFORM_HOME is an environment variable.

When an ant production zip package (created in non legacy mode) is unpacked on a production environment, the SAP Commerce Server can be started without the need to call any ant target, provided that the PLATFORM_HOME and JAVA_HOME environment variables are set correctly.

To create an ant production package in non-legacy mode call:

```
ant production -Dproduction.legacy.mode=false
```

Depending on the `production.include.tomcat` flag, four or five files will be created:

- hybrisServer-Platform.zip
- hybrisServer-AllExtensions.zip
- hybrisServer-Config.zip
- hybrisServer-Licence.zip
- hybrisServer-TomcatConfig.zip

For more details on hybrisServer-TomcatConfig.zip file please consult *Creating the Production Package without Tomcat Application Server* section below.

Note

For information on how ant production related flags work together please read *Ant Production Related Flags Compatibility Matrix* section below

Next, all generated files need to be transfered to the production environment, unpacked and two environment variables need to be set up:

- JAVA_HOME - should be already set - points to JDK or JRE installation
- PLATFORM_HOME - should point to the extracted `hybris/bin/platform` directory

To start SAP Commerce Server please call:

```
./hybrisserver.sh
```

Creating the Production Package without Tomcat Application Server

production.include.tomcat flag

To create ant production zip package that contains no tomcat application server new flag was added:

- production.include.tomcat (defaults to true)

When set to false ant production package will not contain tomcat application server.

Additionally new `hybrisServer-TomcatConfig.zip` file containing SAP Commerce related tomcat configurations will be created.

To create ant production package without tomcat please call:

```
ant production -Dproduction.include.tomcat=false
```

Depending on `tomcat.legacy.deployment` flag setting created `hybrisServer-TomcatConfig.zip` file will contain:

- server.xml file with all SAP Commerce web applications configured inside (tomcat.legacy.deployment=true)

or

- separate configuration files for each of web SAP Commerce applications (`tomcat.legacy.deployment=false`)

additionally `ytomcat.jar` file will be included in both cases

Note

For information on how ant production related flags work together please read *Ant Production Related Flags Compatibility Matrix* section below

To run SAP Commerce Server on standalone tomcat server:

- unpack production package files (see *Creating the Production Package without Tomcat Application Server* section above)
- copy SAP Commerce Server configuration files from `hybrisServer-TomcatConfig.zip` to standalone tomcat directory

To start standalone tomcat, please go to tomcat's bin directory and call:

```
./catalina.sh run
```

Validating Packages Created by Ant Production Target

Use the `production.validate.packages` flag to automatically validate the production packages generated by the ant production target.

production.validate.packages flag

The automatic validation of production packages generated by ant production target is disabled by default. To enable it please call:

```
ant production -Dproduction.validate.packages=true
```

Validation uses `validation.txt` file with rules divided into three sections:

- include ext - list of extension (one per line) names that must be included inside production packages
- exclude ext - list of extension (one per line) names that must NOT be included inside production packages
- exclude dir - list of directories (one per line) that must NOT be included inside production packages (like testsrc or src)

Each section name is preceded with a ';' character, comments starts with '#' character.

For example, putting following content into `validation.txt` file:

```
;include ext
hac
scripting
;exclude ext
mediaconversion
;exclude dir
testsrc
src
```

and running ant production will fail when hac or scripting extension is not included or when `mediaconversion` extension is included, or if any of `testsrc` or `src` directories is present in ant production packages.

Externalizing Generated Classes from a Production Package

production.mutable.platform.separate flag

Since **any** extension may affect generated code (models, DTOs), you must upload it to live servers whenever you make a change to that extension.

Ant production allows you to package generated code (`models.jar`) separately, outside of `hybrisServer-Platform.zip`, so that production update flows can decide to selectively upload only specific extensions.

For example, after you make a change to an extension items.xml file, you don't have to unpack the `hybrisServer-Platform.zip` on a production environment at all. Instead, you only unpack your extension and the generated code delivered in a separate `hybrisServer-MutablePlatform.zip` file.

To generate `hybrisServer-MutablePlatform.zip` file with model and DTO classes, set the `production.mutable.platform.separate` flag to `true`.

Creating Separate Tomcat Configuration Files for Extensions with Web Module

The `tomcat.legacy.deployment` flag controls the way Tomcat context files are generated.

Separate Context Files for Extensions with Web Modules

Each extension with a web module needs a separate context entry in Tomcat's configuration.

Tomcat supports it in two ways:

- all entries can be located in `tomcat/conf/server.xml` file

  or

- there can be a separate xml file per entry

You can use the `tomcat.legacy.deployment` flag to control the way Tomcat context files are generated.

When set to `true`, all context entries will be put into the Tomcat's server.xml file (current and default behavior).

When set to `false`, a separate context xml file is generated. The name of the xml file is created according to the webroot parameter from extensions' `extensioninfo.xml` file.

Additionally, there are custom properties allowing context configuration.

Entering the following properties into `local.properties` file:

`local.properties`

```
tomcat.legacy.deployment=false
backoffice.tomcat.context.Resources.name=jdbc/EmployeeDB
backoffice.tomcat.context.Resources.auth=Container
backoffice.tomcat.context.Resources.type=javax.sql.DataSource
backoffice.tomcat.context.Resources.description=Employees Database for HR Applications
backoffice.tomcat.context.Listener.className=com.sap.core.services.accessor.tomcat.support.NamingResou
backoffice.tomcat.context.Listener.factoryClassName=com.sap.cloud.runtime.kotyo.tomcat.support.Delegat
backoffice.tomcat.context.Listener.subelement.attribute1=someAttribute
backoffice.tomcat.context.Listener.subelement.attribute2=otherAttribute
```

and triggering *ant server* creates `backoffice.xml` file in the directory `tomcat/conf/Catalina/localhost`:

`backoffice.xml`

```
<!-- 'backoffice' extension's context for tenant 'master' -->
   <Context path="/backoffice" docBase="/path/to/hybris/bin/ext-backoffice/backoffice/web/webroot" >
     <Loader className="de.hybris.tomcat.HybrisWebappLoader" platformHome="/path/to/hybris/bin/platfor
     <Resources description="Employees Database for HR Applications" name="jdbc/EmployeeDB" type="java
     <Listener factoryClassName="com.sap.cloud.runtime.kotyo.tomcat.support.DelegatingObjectFactory" c
```

```
  <subelement attribute1="someAttribute" attribute2="otherAttribute"/>
    </Context>
```

The `config/tomcat/tomcat_context.tpl` velocity template is used for context file generation and can be adjusted if needed.

ℹ Note

File containing the context for the root web application (`webroot=""`) is named `ROOT.xml`.

How Old and New Ant Production Related Flags Work Together

Ant Production Related Flags Explained

`tomcat.legacy.deployment`

true - configuration of all SAP Commerce web applications is generated into `server.xml`

false - configuration of all SAP Commerce web applications is generated into separate configuration files

`production.legacy.mode`

true - paths inside server*.xml and wrapper*.conf files are not changed

false - path inside server*.xml and wrapper*.conf files are changed to PLATFORM_HOME related, where PLATFORM_HOME cat be set using an environment variable

`production.include.tomcat`

true - production package contains tomcat application server

false - production package does not contain tomcat application server

Ant Production Related Flags Compatibility Matrix

Table below shows which flags go together in ant production

tomcat.legacy.deployment	production.legacy.mode	production.include.tomcat	usage on production	actions needed to run on production
false	false	false	✅	set JAVA_OPTS=-DPLATFORM_HOME=/path/to/hybris environment variable
false	false	true	✅	set PLATFORM_HOME=/path/to/hybris environment variable, set JAVA_HOME
false	true	false	⛔	flags used together don't make sense on production environment
false	true	true	✅	call ant server
true	false	false	⚠️	outdated server.xml needs to be adjusted

tomcat.legacy.deployment	production.legacy.mode	production.include.tomcat	usage on production	actions needed to run on production
true	false	true	✅	set PLATFORM_HOME=/path/to/hybris environment variable, SET JAVA_HOME
true	true	false	⛔	flags used together don't make sense on production environment
true	true	true	✅	call ant server

Customize Packaging of Extensions

For a efficient roll-out it'd be most useful to split up the default (big) zip files into several smaller ones to **minimize the files to uploaded** simply because he's likely to **change only a few** extensions at a time.

This can be achieved by placing a `ant.production.package.<extensionname>` in the platform configuration, like this example:

```
ant.production.package.yacceleratorstorefront=acc
```

As a result running `ant production` will place the `yacceleratorstorefront` extension in a `hybrisServer-Acc.zip` file instead of `hybrisServer-AllExtensions.zip`.

Preparing Data Hub for a Production Environment

There are several things you must do to complete your installation of Data Hub and prepare it for a production environment.

This section provides the final steps to make your Data Hub installation production ready. Follow the steps in the Related Links section to tune your Tomcat installation, install and integrate a relational database with Data Hub, enable encryption, and define your database cleanup strategy.

ℹ Note

If you have not already installed Data Hub and configured it according to the steps in Start Your Data Hub Journey, do so before completing the steps in this section. If you have not done so, start with Install the Basic Prerequisites.

Related Information

Required Skills
Tuning Tomcat
Choosing a Database
Auto Init Mode
Activating Data Hub Database Cleanup
Data Hub Installation Using Recipes
Upgrading Data Hub from 1811 to 1905
Summary: Installation

Install the Basic Prerequisites

Get set up to install Data Hub.

Context

Data Hub is a Java web application that uses a relational database. The following steps are required for minimal installation and, unless otherwise stated, are **not** valid for third-party software versions.

Procedure

1. Download the latest Data Hub ZIP file containing the DH web application. See the Download section of this guide for further information.
2. Install SapMachine JDK 11 from https://sap.github.io/SapMachine/ . Data Hub supports 64-bit Java.
3. Install Apache Tomcat 9.0.x from https://tomcat.apache.org/download-90.cgi .

   Follow the instructions in the Apache Tomcat documentation to install.
4. Configure Tomcat to add SSL authentication. Follow the instructions in the Tomcat SSL/TLS Configuration How-To guide.

   ### → Tip

   The folders you create in the next steps are required for the Hello World tutorials and for using Data Hub.

5. Create an `/opt/datahub` folder.
6. In the `datahub` folder, create the following new folders:

   a. `config`, used for the files `local.properties` and `datahub.encryption.key.txt`

   b. `extensions`
7. Create an XML file.

   ### → Remember

   All of the Data Hub documentation is based on the `datahub-webapp.xml` file you create in the following steps.

8. Open a new file in a text editor and copy the following into the file:

```
<?xml version='1.0' encoding='utf-8'?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application -->
<Context antiResourceLocking="true"
                docBase="{full path to your Data Hub installation}/hybris/bin/ext-integration/da
                reloadable="true">

    <Resources className="org.apache.catalina.webresources.StandardRoot" >
      <PostResources className="org.apache.catalina.webresources.DirResourceSet"
                    base="/opt/datahub/config" internalPath="/" webAppMount="/WEB-INF/classes" /
      <PostResources className="org.apache.catalina.webresources.DirResourceSet"
                    base="/opt/datahub/extensions" webAppMount="/WEB-INF/lib" />
    </Resources>

</Context>
```

9. Edit the *<docBase>* parameter to reflect the full path to your SAP Commerce installation.
10. Save the file as `datahub-webapp.xml` into the *<TOMCAT_HOME>*`/conf/Catalina/localhost` folder. **Note:** the *<TOMCAT_HOME>*`/conf/Catalina/localhost` does not exist until Tomcat is run.
11. Install the cURL command-line tool.

    From https://curl.haxx.se/ , download and install the cURL software that is appropriate for your operating system.

Related Information

Install SAP Commerce Data Hub

Tuning Tomcat

Tomcat can be tuned to use memory more efficiently, which improves performance and helps Data Hub move more data.

Create your Tomcat startup file. There is a Linux example and a Windows example listed. Just copy and paste to your system. Note that the *<CATALINA_OPTS>* described in the following is just an example. Do not use it **as is** in a production environment. Determine your own best configuration.

Linux - datahubserver.sh

```
#!/bin/bash
# resolve links - $0 may be a softlink
PRG="$0"
while [ -h "$PRG" ]; do
        ls=`ls -ld "$PRG"`
        link=`expr "$ls" : '.*-> \(.*\)$'`
        if expr "$link" : '/.*' > /dev/null; then
                PRG="$link"
        else
                PRG=`dirname "$PRG"`/"$link"
        fi
done
# Get standard environment
PRGDIR=`dirname "$PRG"`
# Explanation of settings:
# * Set the minimum memory to 8192 mb
# * Set the maximum memory to 8192 mb
# * Use the ParNew garbage collector for the young generation heap
# * Use the ConcurrentMarkSweep garbage collector for the old generation heap
# * Tell the JVM to touch all memory pages during JVM initialization
# * Disable explicit garbage collection (for example, through the System.gc() method)
export CATALINA_OPTS="-Xms8192m -Xmx8192m -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:+AlwaysPreTouch
# Run Tomcat in the foreground
$PRGDIR/catalina.sh run
```

Windows - datahubserver.bat

```
@echo off

rem -----------------------------------------------------------------------------
rem Start script for the Data Hub Server
rem -----------------------------------------------------------------------------
```

```
setlocal

rem Explanation of settings in CATALINA_OPTS:
rem * Set the minimum memory to 8192 mb
rem * Set the maximum memory to 8192 mb
rem * Use the ParNew garbage collector for the young generation heap
rem * Use the ConcurrentMarkSweep garbage collector for the old generation heap
rem * Tell the JVM to touch all memory pages during JVM initialization
rem * Disable explicit garbage collection (for example, through the System.gc() method)

set CATALINA_OPTS=-Xms8192m -Xmx8192m -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:+AlwaysPreTouch -X

rem Guess CATALINA_HOME if not defined
set "CURRENT_DIR=%cd%"
if not "%CATALINA_HOME%" == "" goto gotHome
set "CATALINA_HOME=%CURRENT_DIR%"
if exist "%CATALINA_HOME%\bin\catalina.bat" goto okHome
cd ..
set "CATALINA_HOME=%cd%"
cd "%CURRENT_DIR%"
:gotHome
if exist "%CATALINA_HOME%\bin\catalina.bat" goto okHome
echo The CATALINA_HOME environment variable is not defined correctly
echo This environment variable is necessary to run this program
goto end
:okHome

set "EXECUTABLE=%CATALINA_HOME%\bin\catalina.bat"

rem Check that target executable exists
if exist "%EXECUTABLE%" goto okExec
echo Cannot find "%EXECUTABLE%"
echo This file is necessary to run this program
goto end
:okExec

rem Get remaining unshifted command line arguments and save them in the
set CMD_LINE_ARGS=
:setArgs
if ""%1""==""""" goto doneSetArgs
set CMD_LINE_ARGS=%CMD_LINE_ARGS% %1
shift
goto setArgs
:doneSetArgs

call "%EXECUTABLE%" start %CMD_LINE_ARGS%

:end
```

⚠ Caution

If you decide to run Tomcat as a service, the current working directory is switched to the root. During start-up, Data Hub may fail to create the `brokerdb` directory because of a permission issue writing to the root. To resolve the start-up failure, open your Data Hub `local.properties` file. Set the `datahub.event.broker.persistence.directory` property with

the absolute path to the `brokerdb` directory where the Tomcat user has write permissions. For example `datahub.event.broker.persistence.directory=%TOMCAT_HOME%/bin/brokerdb`.

Required Reconfiguration of Tomcat to Support SSL

Add SSL authentication by reconfiguring Tomcat. See HTTPS and SSL.

Adding a Reason to Tomcat's HTTP Response

Tomcat 8.5.x no longer adds a reason to the HTTP response. The lack of response is a change from Tomcat 7 and 8.0. It could possibly affect any tests you have that rely on the reason phrase.

You can enable the reason by adding an attribute `sendReasonPhrase` to a <connector> in the `server.xml` file. Set its value of true (it is false by default). For example:

```
<Connector port="8080" protocol="HTTP/1.1"
              connectionTimeout="20000"
              redirectPort="8443"
              sendReasonPhrase="true" />
```

Related Information

Choosing a Database

Choosing a Database

Data Hub requires a dedicated database for staging data, and saving item metadata and statuses related to load, composition, and publication actions. Data Hub can be configured to use several common relational databases. Flexibility of this type provides you the opportunity to select the ideal database solution for your data integration project.

Overview

Data Hub is a data integration and staging platform that works primarily asynchronously. Raw item data is first loaded from source systems. It is then composed into canonical items. Finally, it is published to one or more target systems in a form suitable for those systems. Each of these stages occurs independently. Because load, composition, and publication events may be triggered at any time, Data Hub must store the following:

- Raw items during the load phase, to be ready for composition.
- Canonical items during the composition phase, to be ready for publication.

In addition, each item includes metadata that defines both its structure, and its relationship to the next state in the data transformation workflow. The workflow being raw to canonical, or canonical to target. Additional metadata is also required to describe each target system.

Thirdly, each item is marked with a status that indicates its progress in the Data Hub workflow. Which would be the outcome of any load, composition, or publication event. Statuses are also recorded for target system publication events.

Persist all of these data types - data items, metadata, and statuses. In sum, they not only enable the function of Data Hub, but also constitute a complete history of all data transformations. A sort of **golden record** history of Data Hub events for auditing purposes.

Persisting this data requires a dedicated database.

Performance

Data Hub employs highly concurrent processing for maximum efficiency and throughput. It uses hibernate for non performance-critical transactions and has its own implementation of a jdbc repository for performance-critical transactions. These Data Hub features provide a level of persistence abstraction that is compatible with a range of common relational databases. Some of the databases may have their own performance limitations, depending on configuration. The choice of database does not affect Data Hub performance in any significant way.

Please refer to the related links section about the individual database topics. Consult with a DBA for the right choice of database. The DBA can help you create a performance-related configuration tailored for the needs of your data integration project.

Data Retention

Any data from completed publications remains in the database, forming a complete auditing record of your data transformation history, as described previously. You may wish to keep the auditing record indefinitely, but over time it can affect Data Hub performance. Previous auditing records can be cleaned up, either manually or automatically, using the provided Data Hub clean up extension. You may also wish to develop your own extension to perform the clean-up according to your requirements. See Activating Data Hub Database Cleanup.

Database Schema

During initialization, Data Hub creates its own schema and initializes this schema with the metadata loaded from its extensions. By default, the `kernel.autoInitMode` is set to `update` to prevent data loss. To refresh the database at any time, drop and create the database manually. Then restart Data Hub to regenerate the schema.

Supported Databases

By default, Data Hub is configured to use the HSQL database. However, HSQL is not a supported database for production deployments.

Note

Data Hub is case sensitive, so your chosen database must also be case sensitive. Of the databases supported by Data Hub, only MySQL is not natively case sensitive. Instructions for configuring it to be case sensitive are included.

For production, Data Hub supports several relational databases. They include:

- MySQL
- Oracle
- SAP HANA DB
- MSSQL

For further information related to the setup and configuration of one of these databases for your Data Hub installation, review Related Links.

Note

To avoid potential issues in certain cases, ensure that your database supports case-sensitive queries. More information is provided in the individual database topics.

Using MySQL

MySQL is a popular, open-source relational database system. Data Hub can be easily configured to use MySQL.

Context

Complete the following steps to configure your Data Hub installation to use a MySQL database.

ⓘ Note

By default, MySQL performs case-insensitive queries, which may be an issue in some cases. Case sensitivity is set using the collate parameter. To enable case-sensitive queries in Data Hub, create the schema so it is configured as follows:

```
CREATE SCHEMA `integration`
DEFAULT CHARACTER SET utf8
COLLATE utf8_bin ;
```

Procedure

1. When you download MySQL, select the MySQL Community Server product.
2. Create your database.

   The default Data Hub installation relies on a database instance with the name of **integration**, with an administrative user named **hybris** and the password **hybris**. You can change the database instance name as well as the username and password. Reflect the changes in your database connection information located in the `local.properties` file. A user with sufficient privileges to grant the rights for the Data Hub database creates the database. Including full schema privileges to the database instance.

   The database can be created from the command line using the **mysql** command-line tool as follows:

   ```
   create database integration;
   ```

3. Create your database user.

   Using the **mysql** command-line tool, enter:

   ```
   CREATE USER 'hybris'@<host>' IDENTIFIED BY 'hybris';
   ```

4. Grant all necessary permissions for the `hybris` user.

   To grant full schema privileges to the `integration` database, enter:

   ```
   GRANT ALL PRIVILEGES ON integration.* TO 'hybris'@'<host>'
       ->     WITH GRANT OPTION;
   ```

5. Add the MySQL database driver to the Tomcat server classpath.

   For example, place `mysql-connector-java-5.1.x-bin.jar` in the <*TOMCAT_HOME*>`/lib` directory.

6. Update your `local.properties` file to reflect your database settings.

   After Data Hub application is deployed in the Web server, a `local.properties` file is created by you to reflect your local setup. The content of the `local.properties` file should reflect your database choice as is shown in the following example:

   `local.properties`

```
dataSource.className=com.mysql.jdbc.jdbc2.optional.MysqlDataSource
dataSource.jdbcUrl=jdbc:mysql://localhost/integration?useConfigs=maxPerformance&rewriteBatchedSt
dataSource.username=...
dataSource.password=...
```

The `local.properties` file is stored in `/opt/datahub/config` in accordance with earlier setup procedures.

7. **Optional:** Provide your database name for JNDI.

   If you are using the JNDI API, add the following property to your `local.properties` file:

   ```
   dataSource.jndiName=database_name
   ```

8. Restart the Tomcat server for the changes to take effect.

Related Information

Auto Init Mode

Using Oracle

Oracle is an enterprise database management system (DBMS) produced by the Oracle Corporation. Data Hub can be easily configured to use an Oracle database.

Context

Complete the following steps to configure your Data Hub installation to use an Oracle database.

! Restriction

When using Oracle SE, Data Hub and SAP Commerce cannot share the same Oracle SE Instance.

i Note

Oracle is case sensitive by default.

Procedure

1. Create your database.

   The default Data Hub installation relies on a database instance with the name of **integration**, with an administrative user named **hybris** and the password **hybris**. You can change the database instance name as well as the username and password. Reflect the changes in your database connection information located in the `local.properties` file. A user with sufficient privileges to grant the rights for the Data Hub database creates the database. Including full schema privileges to the database instance.

   To create an Oracle database and define the administrator, do the following:

   a. Open the DBCA application in the Database Operation window

   b. Select **Create Database**

   c. Click **Next** to start the guided workflow for creating a database

2. Add a `hybris` user.

   Once you create the integration database, add the `hybris` admin user from the command line as follows:

   ```
   CREATE USER hybris IDENTIFIED BY hybris;
   ```

3. Grant all necessary permissions for the `hybris` user.

   The `hybris` user needs all of the following permissions, which can be granted from the command line as follows:

   ```
   grant
   CREATE SESSION, ALTER SESSION, CREATE DATABASE LINK, CREATE MATERIALIZED VIEW,
   CREATE PROCEDURE, CREATE PUBLIC SYNONYM, CREATE ROLE, CREATE SEQUENCE, CREATE SYNONYM,
   CREATE TABLE, CREATE TRIGGER, CREATE TYPE, CREATE VIEW, UNLIMITED TABLESPACE to hybris;
   ```

4. Add an Oracle database driver to the Tomcat server classpath.

   For example, `ojbdc7.jar` would be placed in the `<TOMCAT_HOME>/lib` directory.

5. Update your `local.properties` file to reflect your database settings.

   After Data Hub application is deployed in the web server, a `local.properties` file is created by you to reflect your local setup. The content of the `local.properties` file should reflect your database choice as is shown in the following example:

   `local.properties`

   ```
   dataSource.className=oracle.jdbc.pool.OracleDataSource
   dataSource.jdbcUrl=jdbc:oracle:thin:@rddb01.yrdci.fra.hybris.com:1521:rddb01
   dataSource.username=...
   dataSource.password=...
   ```

   The `local.properties` file is stored in `/opt/datahub/config` in accordance with earlier setup procedures.

6. **Optional:** Provide your database name for JNDI.

   If you are using the JNDI API, add the following property to your `local.properties` file:

   ```
   dataSource.jndiName=database_name
   ```

7. Restart the Tomcat server for the changes to take effect.

Related Information

Auto Init Mode

Using SAP HANA

SAP HANA DB is a high-performance, in-memory database that is part of the SAP HANA platform. Data Hub can be easily configured to use HANA DB.

Context

Complete the following steps to configure your Data Hub installation to use an SAP HANA database.

> **Note**
> SAP HANA is case sensitive by default.

Procedure

1. Create your database and assign a SYSTEM user.

The default Data Hub installation relies on a database instance with the name of **integration**, with an administrative user named **hybris** and the password **hybris**. You can change the database instance name as well as the username and password. Reflect the changes in your database connection information located in the `local.properties` file. A user with sufficient privileges to grant the rights for the Data Hub database creates the database. Including full schema privileges to the database instance.

To create a HANA database and define the administrator:

a. Open the **Manage Databases** application by clicking the tile of the same name on the homepage of the SAP HANA cockpit

b. Choose Create Tenant Database in the footer toolbar

c. Enter the name of the new database and the password of the SYSTEM user

2. Add a HANA database driver to the Tomcat server classpath. SAP Commerce recommends using the latest available HANA database driver.
3. Update your `local.properties` file.

   After Data Hub application is deployed in the web server, a `local.properties` file is created by you to reflect your local setup. The content of the `local.properties` file should reflect your database choice as is shown in the following example:

   `local.properties`

   ```
   dataSource.className=com.sap.db.jdbcext.DataSourceSAP
   dataSource.jdbcUrl=jdbc:sap://10.8.27.8:30115/?reconnect=true
   dataSource.username=...
   dataSource.password=...
   ```

   The `local.properties` file is stored in `/opt/datahub/config` in accordance with earlier setup procedures.

4. **Optional:** Provide your database name for JNDI.

   If you are using the JNDI API, add the following property to your `local.properties` file:

   ```
   dataSource.jndiName=database_name
   ```

5. Restart the Tomcat server for the changes to take effect.

Related Information

Auto Init Mode

Using MSSQL

MSSQL is a relational database management system (DBMS) produced by Microsoft. Data Hub can be easily configured to use MSSQL.

Context

Complete the following steps to configure your Data Hub installation to use a MSSQL database.

ℹ Note

MSSQL is case sensitive by default.

Procedure

1. Create your database.

   The default Data Hub installation relies on a database instance with the name of **integration**, with an administrative user named **hybris** and the password **hybris**. You can change the database instance name as well as the username and password. Reflect the changes in your database connection information located in the `local.properties` file. A user with sufficient privileges to grant the rights for the Data Hub database creates the database. Including full schema privileges to the database instance.

   To create an MSSQL database and define the administrator, open the **SQL Server Management Studio** application and right click the Databases folder icon. Choose New Database. Enter the name of the new database and click OK.

   > **Note**
   >
   > The MSSQL default schema is `dbo`. If you change this schema name, add `kernel.jdbc.schemaName=newschemaname` to your `local.properties` file.

2. Define an administrator user.

   To define a `hybris` administrator account, expand the Security option at the server level. Now right click Logins and click New Login. Click User Mappings, located the integration database, and assign database administrator permissions to the `hybris` user.

3. Add an MSSQL database driver to the Tomcat server classpath.

   For example, `sqljdbc41.jar` would be placed in the `<TOMCAT_HOME>/lib` directory. The correct driver version is chosen depending on the version of MSSQL and Java SDK being used.

4. Update your `local.properties` file.

   After Data Hub application is deployed in the Web server, a `local.properties` file is created by you to reflect your local setup. The content of the `local.properties` file should reflect your database choice as is shown in the following example:

   `local.properties`

   ```
   dataSource.className=com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource
   dataSource.jdbcUrl=jdbc:sqlserver://localhost:1433
   dataSource.username=...
   dataSource.password=...
   ```

   The `local.properties` file is stored in `/opt/datahub/config` in accordance with earlier setup procedures.

5. **Optional:** Provide your database name for JNDI.

   If you are using the JNDI API, add the following property to your `local.properties` file:

   ```
   dataSource.jndiName=database_name
   ```

6. Restart the Tomcat server for the changes to take effect.

Related Information

Auto Init Mode

Install SAP Commerce Data Hub

The following steps help you install Data Hub for use with the Hello World tutorial.

Prerequisites

Before proceeding with the following installation steps, complete the steps described in Install the Basic Prerequisites.

Context

Follow this procedure to install the SAP Commerce Data Hub web application.

Procedure

1. If you do not already have the latest SAP Commerce, download the ZIP file.

   For more information, see Download.

2. Expand the ZIP file on your file system.
3. Download the separate Data Hub ZIP file from the same location, and expand it into your SAP Commerce directory.

   a. Access and download the Data Hub software from the Download Center of SAP Service Marketplace. You may have to contact your S-User (SAP Super User). https://support.sap.com/en/my-support/users.html

   b. Copy the downloaded Data Hub ZIP file into your expanded SAP Commerce directory, at the same level as the `hybris` directory.

   c. From the command line, extract the Data Hub ZIP archive with the command `unzip -u <filename.zip>`, replacing `filename.zip` with the actual name of the downloaded file.

      The `-u` option **updates** the existing `hybris` directory, and places all necessary Data Hub files in the `hybris/bin/ext-integration/datahub` subdirectory.

4. Find the Data Hub version number in the name of the WAR file located in `<HYBRIS_HOME>/hybris/bin/modules/datahub/web-app`. Update the `datahub-webapp.xml` file created in Install the Basic Prerequisites.
5. Data Hub implements HTTPS for all communications to its endpoints and with Tomcat. The Data Hub side of the HTTPS communications is on by default. However, you must add HTTPS authentication to Tomcat by reconfiguring. See https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html .
6. Go to a command line and change directory to *<TOMCAT_HOME>*`/bin`.
7. Launch Tomcat by running *<TOMCAT_HOME>*`/bin/startup.sh` from the command line. When Tomcat launches, the Data Hub WAR file is expanded.
8. Stop Tomcat by typing CTRL-C and then running *<TOMCAT_HOME>*`/bin/shutdown.sh` from the command line. With Tomcat stopped, you can further configure Data Hub.
9. From the command line, change directory to `/opt/datahub/config`.
10. With your favorite text editor, create a file named `local.properties` and save it to `/opt/datahub/config`.
11. Open the `local.properties` file. By default, the following properties are set to false.

```
datahub.cleanup.rawitem.enabled=true
datahub.cleanup.canonicalitem.enabled=true
datahub.cleanup.targetitem.enabled=true
datahub.cleanup.publisheditems.enabled=true
```

i Note

Data Hub includes a set of default local properties. For an example, see datahub-sample.gradle. For a list of available configuration properties for Data Hub, see Configuration Properties.

→ Remember

The cleanup extension is absolutely critical to maintaining a Data Hub that performs well. For additional information, see Activating Data Hub Database Cleanup.

Auto Init Mode

Data Hub provides a configuration property that allows you to control what happens to the database schema during start-up.

Auto-initialization of the Data Hub database schema is possible during the start up cycle of Data Hub. To specify how and if this auto initialization occurs, add the property `datahub.autoInitMode` to your `local.properties` file, as follows:

```
datahub.autoInitMode=create
```

The auto initialization mode has the following options:

- `ignore`: When `datahub.autoInitMode` is set to ignore, the schema is not initialized during startup. If a compatible schema exists, Data Hub uses that schema. If no schema exists, or the schema is incompatible, Data Hub startup fails.
- `create`: When `datahub.autoInitMode` is set to create and there is no existing schema, a new schema is created. If a compatible schema already exists, Data Hub starts up normally.
- `create-drop`: When `datahub.autoInitMode` is set to create-drop, any existing schema is automatically dropped, and a new schema is created. This mode is not suitable for production environments, as loss of data may occur.

The default value for `datahub.autoInitMode` is `create`.

Auto Init Mode and Metadata

The loading of metadata is independent of the `autoInitMode` setting. Any new or changed metadata is always loaded when starting up Data Hub, even in `ignore` mode. The only time the `autoInitMode` setting affects existing metadata is in `create-drop` mode. In this case, drop the entire database schema, and all persisted data is deleted, including metadata. Data Hub then loads any metadata it finds into the fresh schema as part of the normal start-up routine.

The Version Table

The Data Hub database includes a version table, `DataHubVersion`, which holds the current Data Hub version number. This table is used during initialization to check the existing schema against the schema of the Data Hub version starting up. If `datahub.autoInitMode` is set to either `ignore` or `create`, and an incompatible version of the schema is found, Data Hub fails to start. In this case a warning appears in the logs.

```
2016-10-24 12:03:56,273 [ERROR] [o.s.w.c.ContextLoader] Context initialization failed
...
The database is already initialized, but in a different version [expected=6.3.0, actual=6.2.0].
```

In case the version table does not exist, the following appears in the logs.

```
2016-10-24 12:07:24,994 [DEBUG] [c.h.d.p.j.u.i.DefaultJpaSchemaGenerationStrategy] Version table does
2016-10-24 12:07:24,994 [WARN] [c.h.d.p.j.u.i.DefaultAutoInitPropertyMapper] If migrating, please refe
*****************************************************************************************************
* kernel.autoInitMode  * datahub.autoInitMode  * Schema Dropped  * Schema Created  * Schema Updated  *
* ******************  * ********************  * **************  * **************  * **************  *
* create-drop          * create-drop           * Yes              * Yes              * No              *
* update               * create                * No               * If needed       * No              *
* ignore               * ignore                * No               * No              * No              *
*****************************************************************************************************
```

In both these cases, follow the prescribed migration process to upgrade your Data Hub to the current version before proceeding.

When `datahub.autoInitMode` is set to `create-drop`, the database schema always drops, and the database creates a fresh, compatible schema at startup.

Extending the DDL Scripts

If needed for your Data Hub extension, you can extend the provided DDL scripts to include additional tables, data columns, or attributes. During startup, Data Hub uses the scripts provided in the `WEB-INF/classes/META-INF/sql` directory of the

exploded `datahub-webapp.war` file to generate the database schema. This directory has two subdirectories:

- `WEB-INF/classes/META-INF/sql/drop-source` - contains scripts with DROP statements for all tables.
- `WEB-INF/classes/META-INF/sql/create-source` - contains scripts with CREATE statements for all tables.

You can modify these scripts to extend the default schema created at startup. Do not remove any existing tables, columns, or attributes, as Data Hub requires them for normal operation.

Related Information

Securing Your Data Hub Application

Securing Your Data Hub Application

Secure your Data Hub application using simple configuration. Complete these steps to ensure basic end-to-end security of REST endpoints and data attributes.

Configure the Default Security Profile

Context

Data Hub provides a default Spring security profile. Provide authentication credentials for the roles defined in this profile.

Procedure

1. Add the following properties to your `local.properties` file.

   ```
   datahub.security.basic.admin.user=<adminuser>
   datahub.security.basic.admin.password=<adminpass>
   datahub.security.basic.read_only.user=<rouser>
   datahub.security.basic.read_only.password=<ropass>
   ```

2. Provide username and password credentials for each of the two roles defined by the default security profile. Use secure passwords that you do not use for any other purpose.

   Note

   The `datahub.security.basic.admin.user` name and the `datahub.security.basic.read_only.user` name must be different.

Provide Credentials for Data Hub Adapter

Context

If you are using Data Hub with SAP Commerce, provide connection credentials for the Data Hub Adapter so it can connect to Data Hub.

Procedure

Add Data Hub Adapter connection credentials to the `local.properties` file of your SAP Commerce installation. Data Hub Adapter requires both GET and POST access to all Data Hub REST endpoints. Use the credentials you defined for the admin user to allow this access.

```
# The username and password for basic authentication of Data Hub adapter with the Data Hub server inst
datahubadapter.datahuboutbound.user=<adminuser>
datahubadapter.datahuboutbound.password=<adminpass>
```

Create an OAuth Client for Data Hub Adapter

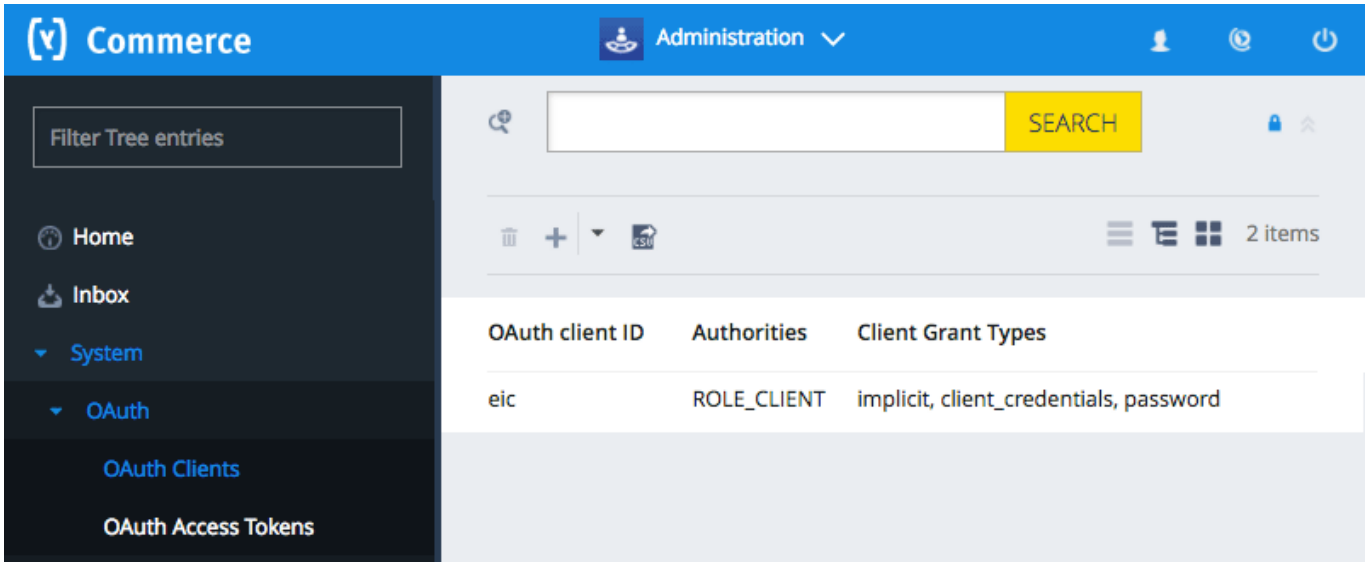
Context

If you are using Data Hub with SAP Commerce and the Data Hub Adapter, configure a dedicated OAuth client for Data Hub. The configuration is done in the Backoffice Administration Cockpit.

Procedure

1. Log into the Backoffice Administration Cockpit.
2. In the Administration perspective, click **System** **OAuth** **OAuth Clients**.

   The list of clients that appears includes the default **eic** client.



3. Click the plus sign next to **OAuth Client Details**.
4. In the client details window, enter a unique ID in *<ClientID>* and the client secret. Click **Next**.

   A new form appears.
5. Click the plus sign under **Authorities**, and enter the authority name `ROLE_CLIENT`.
6. Click the plus sign under **Client Grant Types**, and add the following three grant types, exactly as shown:
   - implicit
   - client-credentials
   - password
7. In a similar way, add the OAuth resource ID `hybris`, and the Redirect URI `https://<yoururl>:9002/auth2_implicit_callback`.

   The base of the Redirect URI can change to match your installation, but `auth2_implicit_callback` must be the same as shown.
8. Click **Next**.
9. Add the OAuth scope `basic`.

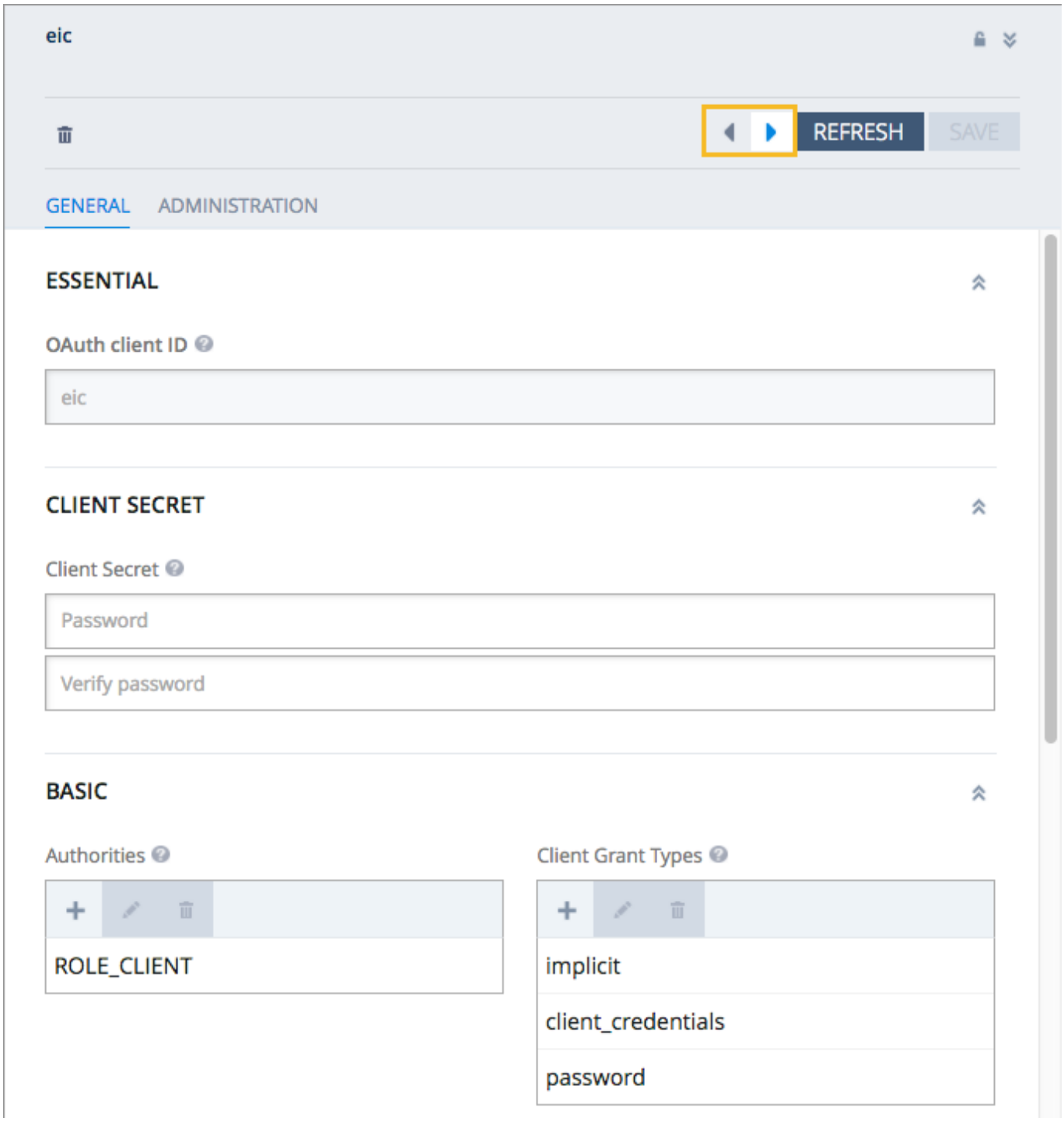
10. Click Done.

    A banner appears, confirming the creation of the new client.

11. To see your new client in the list, click Search to refresh the page.

> **Note**
>
> Except for the ClientID and secret, all properties of your new client should exactly match the properties of the default client, **eic**. To verify, carry out the following steps.

12. Click the line listing for your client to bring up the client details.



13. Use the forward and backward arrows, highlighted in the previous image, to switch between your client and the next.

    You should see no change in any of the properties between your client and the **eic** client, except for the ClientID. The client secret is not shown in cleartext.

14. Configure Data Hub to use the new client, by adding the following properties to your `local.properties` file.

```
datahub.core.export.service.clientid=demo
datahub.core.export.service.clientsecret=secret
```

Restart Data Hub for the properties to take effect.

Set Up Encryption

Context

Data Hub comes with some built-in encryption capabilities for attributes that you wish to keep secure in the data store. Use this service for such items as passwords and other sensitive data. This is a mandatory step, as target system passwords are encrypted by default.

Procedure

1. Generate an encryption key.

   Data Hub uses **AES/ECB/PKCS5Padding** symmetric key encryption with a 128-bit key by default. There are many different ways you can generate an AES key. If you have openSSL installed on your machine, you can generate a 128-bit AES key with a simple terminal command.

   a. Generate a 128-bit AES key using the following terminal command:

   ```
   openssl enc -aes-128-ecb -k secret -P -md sha1
   ```

   The command produces output that looks something like the following:

   ```
   salt=5AFFD797B75FAE1A
   key=ED5D133A8364EA35FC0F4E485C0C0066
   ```

   > **Note**
   > Use only the actual string value of the key; do **not** paste all the output.

   b. Store the key in `/opt/datahub/security-dir/`.

2. Specify the path to the file that contains the key by setting the `datahub.encryption.key.path` property in your `local.properties` file.

   ```
   datahub.encryption.key.path=path/to/encryption-key.txt
   ```

3. **Optional:** Provide a value for the attribute mask. This value is the text that is returned in place of the actual attribute value, for example, in log output.

   Secured attribute value masking can be configured with the following properties. The default is *******.

   ```
   # enable/disable secured attribute value masking
   datahub.secure.data.masking.mode=true

   # set the masking value
   datahub.secure.data.masking.value=*******
   ```

Define Encrypted Attributes

Context

After you configure encryption and store your key, you can specify which attributes to secure.

Procedure

To secure an attribute, set its secured property to `true` in your `raw.xml` and `canonical.xml` data definitions.

```
<attribute>
     <name>attributeName</name>
     <model>
          <type>String</type>
          <secured>true</secured>
      </model>
</attribute>
```

Upon restart, the attribute is encrypted in the Data Hub database, and masked in log files.

Testing Your Secured Configuration

Context

You can test your configuration of the production environment by starting Tomcat and runningData Hub.

Procedure

1. In a command line, change to the Tomcat directory using the command `cd <TOMCAT_HOME>/bin`.
2. Run `<TOMCAT_HOME>/bin/startup.sh` to launch Tomcat.
3. In a new command line window, run the following command to start Data Hub.

   ```
   curl -G https://localhost:8443/datahub-webapp/v1/version
   ```

   The system returns the version number when the configuration is successful.

   Note

   It might take a minute or two for Data Hub to start.

Activating Data Hub Database Cleanup

Over time, a Data Hub database accumulates database records that are used for auditing, but as these records accumulate, they can also affect performance. The following document describes these records and which database tables are affected. Additionally, privacy rules can necessitate the ability to permanently delete consumer data such as orders and personally identifiable information. Database cleanup allows you to remove data related to these situations.

Prerequisites: For additional background or context regarding the following material, see Item Statuses.

When a Data Hub instance has been running for a long time, too many audit records accumulate in the Data Hub database. These records **do** affect performance. If not needed in the active Data Hub database, the historical auditing database records can be migrated to an archive database before elimination. They can also be eliminated without affecting the current state of Data Hub. These operations can be performed even when there are processes currently being performed by Data Hub on the database. The only consequence is that removing the records also removes the audit history for the records. The audit records show how and when they have been imported, composed, and published.

Item Type	Table Name	Status Code for Items to be Deleted
Raw	RawItem	PROCESSED
Canonical	CanonicalItem	ARCHIVED
	CanItemPubStatus	These records have corresponding records in CanonicalItem. After removing the CanonicalItem record, its corresponding record can be removed from this table.
		If there are many successfully published canonical items in your Data Hub database, performance is affected. Set the property `datahub.cleanup.publisheditems.enabled` to true in your local.properties file. The property configures the `datahub-cleanup` extension to delete them. **⚠ Caution** Setting the `datahub.cleanup.publisheditems.enabled` property to true deletes all of your successfully published canonical items except for those explicitly set as excluded. So, the history that might be useful in MDM scenarios is gone. A use case for removing these records is in point-to-point data integration scenarios where the history is typically not needed. Only activate this feature after carefully considering your need for these historical records. For more information about excluded items, see **Excluding Certain Types from Cleanup**, in this topic.
TargetItem	TargetItem	Given the targetsystempublication field refers to an entry in the TargetSystemPub table. Given the status field of that entry does not equal "IN_PROGRESS". The record can be removed.
	CanItemPubStatusErrors	To delete any of these records, the targetsystempublication field must refer to an entry in the TargetSystemPub table where the status field does not equal "IN_PROGRESS".
	PublicationError	

ⓘ Note

If you are using the Data Hub publication in memory option, the `datahub-cleanup` extension processes normally. However, with the publication in memory option, target items do not persist, so the `datahub-cleanup` extension does not do anything with the publication data records.

Using Data Hub's Built-In Extension

The `datahub-cleanup` extension is deployed with the Data Hub WAR file.

Out of the box, the `datahub-cleanup` extension executes a set of default deletion behaviors. The default behavior deletes the following:

- All raw items after they are processed in a composition

- All archived canonical items and their associated publication errors and status after a composition
- All target items after they are finished being used during a publication

To enable the default behavior, set the corresponding properties to **true** in the deployed `local.properties` file as follows:

```
datahub.cleanup.rawitem.enabled=true
datahub.cleanup.canonicalitem.enabled=true
datahub.cleanup.targetitem.enabled=true
datahub.cleanup.publisheditems.enabled=true
```

Data Hub is shipped with these property values set to false.

> ⚠ **Caution**
> Turning on this feature turns off the relationship from Raw to Canonical and Canonical to Raw.

Defining the Cleanup Batch Size

The `datahub-cleanup` extension processes all audit item deletes. The extension is triggered by an event and does its work in batches. The split of one large transaction into multiple smaller batches comes at the cost of total time for deletion. However, it gives the benefit of a more responsive and robust system. It also avoids potential issues with limitations certain databases may impose on the number of unique record IDs included in the IN clause of a query. For example, with Oracle, this number is limited to 1000.

The property `datahub.sql.maxUpdateBatchSize` defines the batch size. Set it to a reasonable value. n is the number of database records deleted in a single query. This property can be set in your `local.properties` file.

```
datahub.sql.maxUpdateBatchSize=500
```

If not explicitly specified, the default batch size is 1000 - the Oracle maximum. You can set the default batch size to any positive integer, but it cannot be set to a negative value.

Defining Canonical Item Cleanup Delay Times

When Data Hub publishes, it gathers data from two groups:

- all the new canonical items
- all canonical items that have previously failed to publish but have not reached their max retry limit

All of these canonical items in the pool are processed with the publication. The publication comprises of one or more publication actions. If there is a maximum publication action size, then the publication action is limited to that number of items. Whatever the number of publication actions used to publish all items, Data Hub creates them and queues them. One set of input data has the possibility of being split across several publication actions, because items are not preassigned to specific publication actions.

The cleanup extension is required for any Data Hub installation, because it has a powerful, positive impact on performance. However, if it is misconfigured, it can have negative impacts. You use the following two timeout properties to configure it. The two timeout properties described in the following are critical for a proper configuration. The properties must be in your `local.properties` file at Data Hub startup. If you activate the cleanup extension without specifying a value for these properties, they each default to 12 hours.

- `datahub.cleanup.canonicalitem.time.delay` – Set the property to a time beyond the time of the longest running publication. The value is usually tuned to something significantly larger than the longest running publication. Such a tuning accounts for spikes in data loads, increased volume over time, and so on. The time value is provided in whole seconds.

```
datahub.cleanup.canonicalitem.time.delay=43200
```

- `datahub.cleanup.publisheditems.time.delay` – Set the property to a time beyond the longest throughput time of any related and/or dependent items. For example, assume that the Customer and Address items are loaded independently, so they may be sent to a target system across two publications. Tune the `datahub.cleanup.publisheditems.time.delay` property, so that these items are still present during the second publication. The time value is provided in whole seconds.

```
datahub.cleanup.publisheditems.time.delay=43200
```

Excluding Certain Types from Cleanup

In some cases, you might want to exclude some canonical item types from deletion by the cleanup extension. There are no rules about which data you should exclude from cleanup. Whether you exclude items depends on your use of the data. For example, you can exclude data that you need to reuse such as configuration data, lookup data, or sales area data. You should exclude any data that is functionally required to be permanently retained in Data Hub. For example, `CanonicalAttributeCategoryRelation`.

To exclude items, use the property `datahub.cleanup.publisheditems.excluded.types`. This property takes a comma-separated list of canonical item types and excludes them from the published item cleanup routine. For example:

```
datahub.cleanup.publisheditems.excluded.types=CanonicalProduct,CanonicalAttributeCategoryRelation
```

Performance is affected when a high volume of successfully published canonical items are in your Data Hub database. To improve performance, the `datahub-cleanup` extension can be activated to delete successfully published canonical items. However, for out-of-the-box implementations, for example, when replicating SAP product and customer data, some canonical item types are required for the subsequent publications to be completed successfully. The following canonical items should not be cleaned up:

sapproduct

- CanonicalAttribute
- CanonicalCategoryProduct
- CanonicalProductTax
- CanonicalProduct
- CanonicalProductSales
- CanonicalProductUnit

sapidocintegration & sapcustomer

- CanonicalParty
- CanonicalAddress

sapproductconfiguration (cpq scenario)

- CanonicalOrder

⚠ Caution

Types added to the exclusion list property accumulate in the database and may cause performance issues over time. To resolve this issue, you can do one of the following:

- Restart Data Hub in create-drop mode.

- Manually remove types from the database through SQL queries.

Related Information

Exploring the Impact of the Clean Up Extension

Where Do I Go from Here?

Data Hub is installed. What do I do now?

Go to Build Essential Data Hub Knowledge and begin working through the feature documentation.

Running SAP Commerce

The SAP Commerce comes with a preconfigured and highly optimized server based on Apache Tomcat for development and production.

SAP Commerce comes prebundled with a Apache Tomcat web container plus configuration settings. The Apache Tomcat can be used out-of-the-box for development and production use. This concept of a prebundled Apache Tomcat and configuration settings is called the SAP Commerce Server. Internally, `embeddedserver` extension provides an API to run an embedded servlet container, while the `tomcatembeddedserver` extension provides a Tomcat-based implementation of this API. To deploy new versions of SAP Commerce, you can replace the entire SAP Commerce Server with a newer version.

About the SAP Commerce Server

The concept of the SAP Commerce Server brings a lot of advantages and benefits for you:

- You can use proven standards (Tomcat) but get professional support, documentation and know-how from SAP
- Very fast deployment cycles
- Simplified hosting
- Setup and configuration process is exactly the same on development machines and production environments.
- SAP Commerce does not need extended JEE features like EJB or JMS. The complete SAP Commerce stack is very lightweight and only needs a Servlet 2.5 compliant web server. SAP Commerce does not need a JEE compliant application server. For executing background tasks like CronJobs or administration tasks, you can even run SAP Commerce as a standalone Java application - completely without any application server or web server.
- With various enhancements like automatic updating functionality , a refurbished directory layout and a completely preconfigured and optimized Server based on standard Apache Tomcat technology, SAP Commerce dramatically simplifies the deployment of productive systems. SAP Commerce fully supports virtualized environments and helps you by providing preconfigured virtual images for development and production to further reducing the effort of installing and updating the SAP Commerce software.

Starting the SAP Commerce Server

Follow the steps to start the SAP Commerce Server.

Basically, you can start the SAP Commerce Server in these ways:

- From the command line
- As a system service

Starting the SAP Commerce Server From the Command Line

Normal operation mode:

1. Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform` directory.
2. To start the SAP Commerce Server:
   - On Windows systems call the `hybrisserver.bat` file.
   - On Unix systems call the `hybrisserver.sh` file, such as: `./hybrisserver.sh`

Debug operation mode, requiring `develop` configuration template:

1. Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform` directory.
2. To start the SAP Commerce Server:
   - On Windows systems run the `hybrisserver.bat` file with the debug parameter, such as `hybrisserver.bat debug`.
   - On Unix systems call the `hybrisserver.sh` file with the debug parameter, such as `./hybrisserver.sh debug`.

For more information, see Configuration Templates.

Starting the SAP Commerce Server as a Service

Refer to http://technet.microsoft.com/en-us/library/cc736564(WS.10).aspx : Microsoft Technet on using services.

To run the SAP Commerce Server as a system service, you need to install it.

Installing the SAP Commerce Server as a Service

Under Microsoft Windows systems:

- Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform/tomcat-6/bin` directory.
- Call the `InstallTomcatService.bat` file.

Removing the SAP Commerce Server Service

To remove the SAP Commerce Server service:

- Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform/tomcat-6/bin` directory.
- Call the `UninstallTomcatService.bat` file.

Using the SAP Commerce Server

You can use the SAP Commerce Server in development and production scenarios.

ℹ Note

The SAP Commerce Server binaries are bundled with the Platform. You do not need to download the SAP Commerce Server separately.

You can use the SAP Commerce Server in two scenarios without major re-configuration in between:

- Development:

  You run the SAP Commerce Server on the same machine on which you develop. An explicit deployment is not necessary.

- Production use:

  The SAP Commerce Server is run on a machine different from the one on which you develop. Usually, some sort of deployment process is used.

Development Operation

Basically, the setup procedure is the same for both development and production scenario. Refer to Installing a Local Instance for details on installing SAP Commerce.

During the first SAP Commerce build, select the `develop` configuration template (see Configuration Templates).

After that, you can develop your extension(s). To "deploy" modified extensions or configuration, simply restart the SAP Commerce Server. During restart, the modified extensions and configuration will be loaded automatically.

Production Operation

Production operation differs from development mainly in the fact that you will want to deploy SAP Commerce on a remote server.

During the first SAP Commerce build, select the `production` configuration template (see Configuration Templates).

Installation

Basically, the setup procedure is the same for both development and production scenario. Refer to the Installing a Local Instance document for details on installing SAP Commerce.

Deployment

Deployment takes place in these phases:

- Building the deployment files; see Building the Deployment Files
- Deploying the deployment files; see Deploying the Deployment Files
- Building and restarting SAP Commerce; see Building and Restarting SAP Commerce

Running SAP Commerce Server as a Service

SAP Commerce Server can optionally be installed as a service. To run as a service, SAP Commerce Server can be started, stopped, and restarted more conveniently than when run in a shell, as this is done by the `hybrisserver.bat` and `hybrisserver.sh` files.

Building the Deployment Files

Calling *ant production* on the development machine results in creating two deployment files.

The files include:

- `hybrisServer-Platform.zip` - This file contains the SAP Commerce Server. You will only need to deploy this file if you want to update the SAP Commerce Server. For example, to install a new version of SAP Commerce.
- `hybrisServer-AllExtensions.zip` - This file contains the extensions of SAP Commerce. It does not contain the Platform, which is contained in the `hybrisServer-Platform.zip` file as part of the SAP Commerce Server.

Deploying the Deployment Files

Follow the steps to deploy the `hybrisServer-AllExtensions.zip` and `hybrisServer-Platform.zip` files.

1. Stop SAP Commerce Server on the remote server if running.
2. Copy the `hybrisServer-AllExtensions.zip` file to the remote server.
3. Delete the directories in the *<${HYBRIS_BIN_DIR}>* directory, except for `/platform`.
4. Unzip the `hybrisServer-AllExtensions.zip` file into the *<${HYBRIS_BIN_DIR}>* directory. This way, you will replace the existing non-platform extensions with the new versions.
5. If you want to deploy SAP Commerce Server as well:
    a. Copy the `hybrisServer-Platform.zip` file to the remote server.
    b. Delete the `/platform` directory in the *<${HYBRIS_BIN_DIR}>* directory.
    c. Unzip the `hybrisServer-Platform.zip` file into the *<${HYBRIS_BIN_DIR}>* directory. This way, you will replace the existing SAP Commerce Server including the Platform with the new versions.

Building and Restarting SAP Commerce

Follow the steps to build SAP Commerce on the remote server, and restart SAP Commerce Server.

1. Build SAP Commerce on the remote server:
    - Open a command shell.
    - Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform` directory.
    - Make sure that a compliant Apache Ant version is used:
        - On Windows systems, call the *<${HYBRIS_BIN_DIR}>* `/platform/setantenv.bat` file. Do not close the command shell after this call as the settings are transient and would get lost if the command shell is closed.
        - On Unix systems, call the *<${HYBRIS_BIN_DIR}>* `/platform/setantenv.sh` file, such as: `. ./setantenv.sh`
    - Call *ant clean all* to build the entire SAP Commerce solution.
2. Restart SAP Commerce Server:

    Normal operation mode:

    a. Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform` directory.
    b. To start SAP Commerce Server:
        - On Windows systems, call the `hybrisserver.bat` file.
        - On Unix systems, call the `hybrisserver.sh` file, such as: `./hybrisserver.sh`

    Debug operation mode, requiring `develop` configuration template:

    a. Navigate to the *<${HYBRIS_BIN_DIR}>* `/platform` directory.
    b. To start SAP Commerce Server:
        - On Windows systems, run the `hybrisserver.bat` file with the debug parameter, such as `hybrisserver.bat debug`.
        - On Unix systems, call the `hybrisserver.sh` file with the debug parameter, such as `./hybrisserver.sh debug`.

    For more information, see Configuration Templates.

Configuring SAP Commerce Server

All of the SAP Commerce Server settings come in the form of properties and can be defined or overridden using the `local.properties` file.

They affect different aspects of the SAP Commerce Server:

- Tomcat-specific properties
- SAP Commerce specific properties
- Log4J-specific properties

The process of how these properties are loaded and take effect have similiarites and differences:

- All kinds of properties are defined (or overridden) in the `local.properties` file (see Configuring the Behavior of SAP Commerce
- All kinds of properties are loaded on the SAP Commerce Server start-up from the file system.
- Tomcat-specific properties are written to a Tomcat configuration file during the SAP Commerce build, while the SAP Commerce specific properties and Log4J-specific properties are not written to any other file.

  Although the Tomcat-specific properties will be available by SAP Commerce runtime and can be read and set, modifying these values will have no effect: by the time you could set the values "from inside", the Java Virtual Machine is already running.

The aspect that properties are read in during the SAP Commerce Server start-up means that you will not have to re-build the SAP Commerce after you have made the SAP Commerce specific changes to the `local.properties` file. Instead, restarting the SAP Commerce Server will do; the SAP Commerce Server will pick up the modifications on restart. If you have modified Tomcat-specific properties in the `local.properties` file, you will have to call *ant deploy* for the changes to take effect. Refer to Building SAP Commerce for details.

Apache Tomcat Properties

See a list of configuration properties specific to the Apache Tomcat embedded in SAP Commerce. You can see the defult settings in the `project.properties` file.

You can override the factory default values using the `local.properties` file (see Configuring the Behavior of SAP Commerce).

Property name	Property description
`tomcat.http.port`	Specifies the port that allows access to SAP Commerce via an unsecured connection.
`tomcat.ssl.port`	Specifies the port that allows access to SAP Commerce via an SSL-secured connection.
`tomcat.jmx.port`	Specifies the port that allows access to SAP Commerce via JMX.
`tomcat.javaoptions`	Java runtime options for the SAP Commerce Server in normal operation mode (`hybrisserver.bat` / `hybrisserver.sh`)
`tomcat.debugjavaoptions`	Java runtime options for the SAP Commerce Server in debug mode (`hybrisserver.bat debug` / `hybrisserver.sh debug`)
`tomcat.generaloptions`	Java runtime options for the SAP Commerce Server used in both the normal operation mode and debug mode.
`production.output.path`	The directory into which the files created by `ant production` will be created.

→ Tip

There may be situations in which you might have to modify advanced, Apache Tomcat-specific settings. In this case, you can directly modify the configuration of the SAP Commerce Server Apache Tomcat. The *<${HYBRIS_CONFIG_DIR}>* `/tomcat/conf` directory contains configuration files for the Apache Tomcat and the Tanuki Java Wrapper, such as `server.xml`. Modifying these files is for advanced users and in most situations, you will not have to do that.

The Tomcat's `useStatementFacade` property is set to `true` by default. SAP Commerce doesn't support `useStatementFacade` and our workaround sets it to `false`. If you use a non-bundled Tomcat, we recommend that you set `useStatementFacade` to `false`.

Separate Context Files for Extensions with Web Modules

Each extension with a web module needs a separate context entry in Tomcat's configuration.

Tomcat supports it in two ways:

- all entries can be located in `tomcat/conf/server.xml` file

  or

- there can be a separate xml file per entry

You can use the `tomcat.legacy.deployment` flag to control the way Tomcat context files are generated.

When set to `true`, all context entries will be put into the Tomcat's server.xml file (current and default behavior).

When set to `false`, a separate context xml file is generated. The name of the xml file is created according to the webroot parameter from extensions' `extensioninfo.xml` file.

Additionally, there are custom properties allowing context configuration.

Entering the following properties into `local.properties` file:

`local.properties`

```
tomcat.legacy.deployment=false
backoffice.tomcat.context.Resources.name=jdbc/EmployeeDB
backoffice.tomcat.context.Resources.auth=Container
backoffice.tomcat.context.Resources.type=javax.sql.DataSource
backoffice.tomcat.context.Resources.description=Employees Database for HR Applications
backoffice.tomcat.context.Listener.className=com.sap.core.services.accessor.tomcat.support.NamingResou
backoffice.tomcat.context.Listener.factoryClassName=com.sap.cloud.runtime.kotyo.tomcat.support.Delegat
backoffice.tomcat.context.Listener.subelement.attribute1=someAttribute
backoffice.tomcat.context.Listener.subelement.attribute2=otherAttribute
```

and triggering *ant server* creates `backoffice.xml` file in the directory `tomcat/conf/Catalina/localhost`:

`backoffice.xml`

```
<!-- 'backoffice' extension's context for tenant 'master' -->
   <Context path="/backoffice" docBase="/path/to/hybris/bin/ext-backoffice/backoffice/web/webroot" >
     <Loader className="de.hybris.tomcat.HybrisWebappLoader" platformHome="/path/to/hybris/bin/platfor
     <Resources description="Employees Database for HR Applications" name="jdbc/EmployeeDB" type="java
     <Listener factoryClassName="com.sap.cloud.runtime.kotyo.tomcat.support.DelegatingObjectFactory" c
 <subelement attribute1="someAttribute" attribute2="otherAttribute"/>
   </Context>
```

The `config/tomcat/tomcat_context.tpl` velocity template is used for context file generation and can be adjusted if needed.

Note

File containing the context for the root web application (`webroot=""`) is named `ROOT.xml`.

Running Platform on an External Tomcat

You can run the Platform either on an external Tomcat server or on the one provided in the Platform.

By default, the Platform runs on a Tomcat server provided in the Platform. You must first configure the Platform to run on an external Tomcat server. You must appropriately set up two variables before you can run the Platform on an external Tomcat server. These variables allow you to point to the location of a Tomcat server to be used.

Set up both the *<${bundled.tomcat.home}>* and *<${bundled.tomcat.base}>* variables.

By default, the variables are set to *<${platformhome}\tomcat>*. The `${platformhome}\tomcat` attribute points specifically to the Platform's `tomcat` folder containing Tomcat server files provided with the Platform:

The variables are set up to point to the Platform's tomcat folder.

```
bundled.tomcat.home=${platformhome}/tomcat
bundled.tomcat.base=${platformhome}/tomcat
```

To run the Platform on an external Tomcat server, follow the steps listed in these two sections:

- Configuring the Platform to run on an external Tomcat server
- Starting the Platform on an external Tomcat server

Configuring the Platform to run on an external Tomcat server

To configure the Platform to run on an external Tomcat server, set up the *<${bundled.tomcat.home}>* and *<${bundled.tomcat.base}>* variables to point to an external Tomcat server's location, as shown in the example:

The variables are set up to point to an external tomcat folder

```
bundled.tomcat.home=c:/someDirectory/tomcat
bundled.tomcat.base=c:/someDirectory/tomcat
```

Starting the Platform on an external Tomcat server

To start the Platform on an external Tomcat server, instead of running *hybrisserver.[bat|sh]* use the *catalina run* command from the `${bundled.tomcat.home}\bin` directory.

Downloading a Tomcat Server

In case the external or Platform's **tomcat** folder is empy, you can download the required version of a Tomcat server during the Platform's build process.

Use the **${bundled.tomcat.version}** variable to set up the required version of a Tomcat server, as shown in the example:

Setting up a required version of a Tomcat server to download

```
bundled.tomcat.version=7.0.59
```

Use the **${bundled.tomcat.home}** and **${bundled.tomcat.base}** variables to set up the directory into which a Tomcat server should be downloaded, as shown in the example:

Setting up a location for a Tomcat server

```
bundled.server.type=tomcat
bundled.tomcat.home=c:/someDirectory/tomcat
bundled.tomcat.base=c:/someDirectory/tomcat
bundled.tomcat.version=7.0.59
```

Note

Files `catalina-jmx-remote.jar` and `tomcat-juli-adapters.jar` need to be inside the tomcat lib folder. Otherwise, you get the following error:

Output Code

```
Launching a JVM...
WrapperManager: Initializing...
May 13, 2014 10:00:20 AM org.apache.tomcat.util.digester.Digester startElement
SEVERE: Begin event threw exception
java.lang.ClassNotFoundException: org.apache.catalina.mbeans.JmxRemoteLifecycleListener
at java.net.URLClassLoader$1.run(URLClassLoader. java:366 )
at java.net.URLClassLoader$1.run(URLClassLoader. java:355 )
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader. java:354 )
at java.lang.ClassLoader.loadClass(ClassLoader. java:424 )
at java.lang.ClassLoader.loadClass(ClassLoader. java:357 )
at org.apache.tomcat.util.digester.ObjectCreateRule.begin(ObjectCreateRule. java:144 )
at org.apache.tomcat.util.digester.Digester.startElement(Digester. java:1288 )
at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.startElement(AbstractSAXParser. jav
at com.sun.org.apache.xerces.internal.parsers.AbstractXMLDocumentParser.emptyElement(AbstractXMLDoc
at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanStartElement(XMLDocum
at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl$FragmentContentDriver.nex
at com.sun.org.apache.xerces.internal.impl.XMLDocumentScannerImpl.next(XMLDocumentScannerImpl. java
at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentF
at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration. java:846
at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration. java:775
at com.sun.org.apache.xerces.internal.parsers.XMLParser.parse(XMLParser. java:123 )
at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.parse(AbstractSAXParser. java:1210
at com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser.parse(SAXParserImpl. java:62
at org.apache.tomcat.util.digester.Digester.parse(Digester. java:1561 )
at org.apache.catalina.startup.Catalina.load(Catalina. java:615 )
at org.apache.catalina.startup.Catalina.load(Catalina. java:663 )
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method. java:606 )
at org.apache.catalina.startup.Bootstrap.load(Bootstrap. java:280 )
at org.apache.catalina.startup.Bootstrap.main(Bootstrap. java:454 )
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

```
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method. java:606 )
at org.tanukisoftware.wrapper.WrapperSimpleApp.run(WrapperSimpleApp. java:290 )
at java.lang.Thread.run(Thread. java:724 )
May 13, 2014 10:00:20 AM org.apache.catalina.startup.Catalina load
WARNING: Catalina.start using conf/server.xml: Error at (6, 141) : org.apache.catalina.mbeans.JmxRe
May 13, 2014 10:00:20 AM org.apache.tomcat.util.digester.Digester startElement
SEVERE: Begin event threw exception
java.lang.ClassNotFoundException: org.apache.catalina.mbeans.JmxRemoteLifecycleListener
at java.net.URLClassLoader$1.run(URLClassLoader. java:366 )
at java.net.URLClassLoader$1.run(URLClassLoader. java:355 )
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader. java:354 )
at java.lang.ClassLoader.loadClass(ClassLoader. java:424 )
at java.lang.ClassLoader.loadClass(ClassLoader. java:357 )
at org.apache.tomcat.util.digester.ObjectCreateRule.begin(ObjectCreateRule. java:144 )
at org.apache.tomcat.util.digester.Digester.startElement(Digester. java:1288 )
at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.startElement(AbstractSAXParser. jav
at com.sun.org.apache.xerces.internal.parsers.AbstractXMLDocumentParser.emptyElement(AbstractXMLDoc
at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanStartElement(XMLDocum
at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl$FragmentContentDriver.nex
at com.sun.org.apache.xerces.internal.impl.XMLDocumentScannerImpl.next(XMLDocumentScannerImpl. java
at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentF
at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration. java:846
at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration. java:775
at com.sun.org.apache.xerces.internal.parsers.XMLParser.parse(XMLParser. java:123 )
at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.parse(AbstractSAXParser. java:1210
at com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser.parse(SAXParserImpl. java:62
at org.apache.tomcat.util.digester.Digester.parse(Digester. java:1561 )
at org.apache.catalina.startup.Catalina.load(Catalina. java:615 )
at org.apache.catalina.startup.Catalina.start(Catalina. java:677 )
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method. java:606 )
at org.apache.catalina.startup.Bootstrap.start(Bootstrap. java:321 )
at org.apache.catalina.startup.Bootstrap.main(Bootstrap. java:455 )
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method. java:606 )
at org.tanukisoftware.wrapper.WrapperSimpleApp.run(WrapperSimpleApp. java:290 )
at java.lang.Thread.run(Thread. java:724 )
May 13, 2014 10:00:20 AM org.apache.catalina.startup.Catalina load
WARNING: Catalina.start using conf/server.xml: Error at (6, 141) : org.apache.catalina.mbeans.JmxRe
May 13, 2014 10:00:20 AM org.apache.catalina.startup.Catalina start
SEVERE: Cannot start server. Server instance is not configured.
<-- Wrapper Stopped
```

SAP Commerce Specific Properties

You can define runtime parameters for the SAP Commerce server, such as cache size, CronJob, or LDAP settings.

You can override the factory default values using the `local.properties` file.

The settings are for SAP Commerce and do not affect the application server directly. These settings are not discussed here in depth, refer to Configuration Properties instead.

Log4J-Specific Properties

Although they are a subset of the SAP Commerce specific properties, the Log4J-related properties are handled in a special way.

Log4J-related settings are loaded prior to the SAP Commerce specific properties, but are not written to any persistent file (as is the case with Tomcat-specific properties).

You can override the factory default values using the `local.properties` file. For more information, see Configuring the Behavior of SAP Commerce.

Upgrading Tomcat

Follow the instructions to upgrade the Tomcat server shipped with the Platform. You may need to do it, for example, if any security issues have been solved in a recent Tomcat version.

Procedure

1. Copy the following files to a backup directory (paths are relative to the directory `platform/tomcat`):

   From the `bin` folder:

   - `InstallTomcatService.bat`
   - `UninstallTomcatService.bat`
   - `catalina.bat`
   - `catalina.sh`
   - `debug.bat`
   - `debug.sh`
   - `jdk_logging.properties`
   - `o.bat`
   - `o.sh`
   - `wrapper-aix-ppc-32`
   - `wrapper-aix-ppc-64`
   - `wrapper-freebsd-x86-32`
   - `wrapper-freebsd-x86-64`
   - `wrapper-hpux-ia-32`
   - `wrapper-hpux-ia-64`
   - `wrapper-hpux-parisc-32`
   - `wrapper-hpux-parisc-64`
   - `wrapper-linux-390-32`
   - `wrapper-linux-390-64`

- wrapper-linux-armel-32
- wrapper-linux-armhf-32
- wrapper-linux-ppc-32
- wrapper-linux-ppc-64
- wrapper-linux-x86-32
- wrapper-linux-x86-64
- wrapper-macosx-universal-32
- wrapper-macosx-universal-64
- wrapper-solaris-sparc-32
- wrapper-solaris-sparc-64
- wrapper-solaris-x86-32
- wrapper-solaris-x86-64
- wrapper-windows-ia-64.exe
- wrapper-windows-x86-32.exe
- wrapper-windows-x86-64.exe
- wrapper.bat
- wrapper.sh

From the `conf` folder:

- README.txt
- hybris-wrapper-license.conf
- i3-log.conf

From the `lib` folder:

- keystore
- libwrapper-aix-ppc-32.a
- libwrapper-aix-ppc-64.a
- libwrapper-freebsd-x86-32.so
- libwrapper-freebsd-x86-64.so
- libwrapper-hpux-ia-32.so
- libwrapper-hpux-ia-64.so
- libwrapper-hpux-parisc-32.sl
- libwrapper-hpux-parisc-64.sl
- libwrapper-linux-390-32.so
- libwrapper-linux-390-64.so
- libwrapper-linux-armel-32.so
- libwrapper-linux-armhf-32.so
- libwrapper-linux-ia-64.so
- libwrapper-linux-ppc-32.so
- libwrapper-linux-ppc-64.so
- libwrapper-linux-x86-32.so

- libwrapper-linux-x86-64.so
- libwrapper-macosx-universal-32.jnilib
- libwrapper-macosx-universal-64.jnilib
- libwrapper-solaris-sparc-32.so
- libwrapper-solaris-sparc-64.so
- libwrapper-solaris-x86-32.so
- libwrapper-solaris-x86-64.so
- wrapper-version.txt
- wrapper-windows-ia-64.dll
- wrapper-windows-x86-32.dll
- wrapper-windows-x86-64.dll
- wrapper.jar

From the `logs` folder:

- WHERE_ARE_THE_LOGS.txt

2. Delete the `tomcat` directory.
3. Download and unzip your new Tomcat to the `platform/tomcat` directory.
4. Download and unzip tomcat extras (`catalina-jmx-remote, tomcat-juli-adapters`). If necessary, rename the files to `catalina-jmx-remote.jar` and `tomcat-juli-adapters.jar`.
5. Copy files from step 1 back to the `tomcat` directory.

Deployment of External WAR files

Add the `webapp` element to `(local)extensions.xml` to enable the deployment of external `.war` files.

Add the webapp element to `localextensions.xml` as shown in the following examples.

```
<extensions>

        <extension .../>

    <!-- .war file -->
    <webapp contextroot="foo" path="${HYBRIS_BIN_DIR}/path/to/webapp.war"/>
        <!-- exploded war archive -->
    <webapp contextroot="bar" path="/path/to/exploded-webapp"/>
        <!-- context file -->
        <webapp context="/path/to/context.xml" />

 </extensions>
```

The `webapp` element can have either one (`context`) or two (`contextroot`, `path`) attributes. Their meaning is explained in the following table.

webapp attribute name	description
contextroot	The webroot for web application (without trailing "/")
path	The path either to the WAR file or to an exploded `webapp` directory
context	The path to `context.xml` file with `webapp` description

The system recognizes placeholders like *<${HYBRIS_BIN_DIR}>* in paths. External web applications are extracted and copied to the *<HYBRIS_BIN_DIR>*`/bin/custom` directory.

The context.xml File

The `context.xml` file must contain `path` and `docBase` attributes. Its content is copied into the embedded Tomcat deployment descriptor, either `server.xml` or a separate `context.xml` file depending on the `tomcat.legacy.deployment` flag.

`context.xml`

```
<Context path="/sample" docBase="/path/to/sample.war" />
```

Related Information

Configuring the Behavior of SAP Commerce
Clustered Environment
Building SAP Commerce
http://tomcat.apache.org

Deploying New SAP Commerce Software Versions

You can deploy a new version of SAP Commerce on a single server machine or in a cluster. When deploying to a cluster environment, you must install the new software version one node at a time.

SAP Commerce is regularly released with new versions that contain bug fixes and improvements. You may also want to deploy self-implemented versions of SAP Commerce software on a test system, for example, during a development project. For information about upgrading to a specific SAP Commerce version, see Upgrading SAP Commerce.

Creating a Deployment-ready Version of a New SAP Commerce Version

This section is executed on your local machine. You do not require access to the target deployment system to perform these steps.

Procedure

To create a deployment-ready version of the new SAP Commerce software version:

1. On the target machine, create the folder where you will extract the SAP Commerce installation ZIP file.
2. Download SAP Commerce to the target machine.

   To download the SAP Commerce installation ZIP file, you require a valid license key. If you do not have one, contact your SAP sales representative. For more details, see Download.
3. Extract the ZIP file to the newly-created installation folder, for example: `C:\hybris`.
4. Optionally, if you have a hotfix to install:
   a. Unzip your hotfix file to your *<customize>* folder.
   b. Call `ant customize` to replace selected files within the *<HYBRIS_BIN_DIR>* directory with custom versions of the file.

   For details, see Building SAP Commerce.
5. Build SAP Commerce.

a. Open a command prompt.

b. Navigate to the `<HYBRIS_BIN_DIR>/platform` directory.

c. Make sure that a compliant version is used.

- On a Windows operating system, call the `<HYBRIS_BIN_DIR>/platform/setantenv.bat` file. Do not close the command prompt after this call as the settings are transient and would get lost if the command prompt is closed.
- On a Unix operating system, call the `<HYBRIS_BIN_DIR>/platform/setantenv.sh` file, such as: `. ./setantenv.sh`.

a. Call `ant clean all` to build the entire SAP Commerce solution.

6. Build the deployment files of SAP Commerce.

   There are two kinds of deployment files, depending on whether you use the SAP Commerce Server (see Running SAP Commerce) or a third-party application server:

   - For the SAP Commerce Server, deploy new versions of SAP Commerce in the form of two ZIP files. Call `ant production` in the `<HYBRIS_BIN_DIR>/platform` directory.
   - For third-party application servers, deploy new versions of SAP Commerce in a single EAR file. Call `ant ear` in the `<HYBRIS_BIN_DIR>/platform` directory.

Deploying the New SAP Commerce Software Version in a SAP Commerce Cluster

This section is executed on the deployment target system.

Context

This procedure assumes your SAP Commerce cluster consists of three nodes. However, the basic procedure remains the same regardless of the number of cluster nodes.

Procedure

Phase 1: Deploy the new SAP Commerce software version on a single node.

1. Shut down the individual node from cluster operation.

   → Tip

   If possible, select a node that is not connected to the outside world and is not used by customers. This allows you to shut down the node with as little disruption as possible. If you shut down a live cluster node, all the user sessions instantiated on that node will be lost. This means that customers will lose their session (cart, etc) if you shut down the node.

   You can avoid customer data loss by:

   a. Removing the node from management using the load balancer.

   b. Allowing the user sessions time out automatically. This may take some time, depending on your settings; the default setting for a session lifetime is one hour.

   If your implementation uses SessionFailover, no user session information will be lost because the sessions will be persisted in the database. The load balancer will distribute the sessions across the remaining, live cluster nodes.

   a. Select a single node on which you will run the system update. For example, Node 1.

   b. If Node 1 is managed by a loadbalancer: Remove Node 1 from being managed by the system load balancer.

c. Stop the SAP Commerce running on Node 1. Do not stop the other nodes in the cluster. Do not stop the database.

2. Deploy the new SAP Commerce software version.

   a. Back up the version of SAP Commerce that is currently on Node 1. For example, rename the `platform` directory into `platform-6_0`.

   b. Create a database dump to have a backup of the database that contains the data for the currently installed version of the SAP Commerce software.

   c. Deploy the new SAP Commerce version on Node 1.

3. Restart the node.

   a. Start the newly installed version of SAP Commerce on Node 1.

Phase 2: Perform an upgrade using the single node.

4. Update your version of SAP Commerce as described in Upgrading SAP Commerce.

5. If the updated node was managed by a load balancer prior to deployment of the new SAP Commerce software version, put the node back under the management of the load balancer.

Phase 3: Deploy the new SAP Commerce software version on the other nodes of the cluster.

6. Execute the following procedure on each node within the cluster. Upgrade the nodes one at a time.

   a. If the node is managed by a load balancer, remove a node that is running the existing version of the SAP Commerce from the management of the system load balancer.

   b. Shut down the node.

   c. Deploy the new version of the SAP Commerce software on the node.

   d. Restart the node.

   e. If the updated node was managed by a load balancer prior to deployment of the new SAP Commerce software version, put the node back under the management of the load balancer.

API, Data Model, and Spring Framework Changes

The 1905 release of SAP Commerce brings important API, data model, and Spring framework changes. See the tables and follow the links to the generated reports to discover the details.

API Changes

View API changes between 1811 and 1905 of SAP Commerce in the following HTML report: API Compatibility Report 1811 to 1905

Data Model Changes

The following sections list the data model changes in SAP Commerce between 1811 and 1905.

Type Comparer Report of Problems

The following table lists results from the Type Comparer Report. The first column lists the extension name, the second column lists the type of change that was reported, while the remaining columns indicate what changed for the relevant property or value.

Type Comparer Report of Problems

Extension	Type of Change	Property 1	Value 1	Property 2	Valu

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Attribute not found because the extension was de-released	type	User	qualifier	rule
btg	Attribute not found because the extension was de-released	type	User	qualifier	segr
btg	Attribute not found because the extension was de-released	type	User	qualifier	conc
btg	Attribute not found because the extension was de-released	type	UserGroup	qualifier	BTG
commercesearch	Attribute not found because the extension was de-released	type	Product	qualifier	solr
btg	Attribute not found because the extension was de-released	type	AbstractAction	qualifier	segr
btg	Attribute not found because the extension was de-released	type	AbstractAction	qualifier	segr
apiregistryservices	Attribute not found	type	OAuthClientDetails	qualifier	oAut

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
integrationservices	Attribute not found	type	IntegrationObjectItemAttribute	qualifier	part
commercesearch	Attribute not found because the extension was de-released	type	SolrIndexedProperty	qualifier	solrI
commercesearch	Attribute not found because the extension was de-released	type	SolrIndexedProperty	qualifier	boos
commercesearch	Attribute not found because the extension was de-released	type	SolrIndexedProperty	qualifier	face
commercesearch	Attribute not found because the extension was de-released	type	SolrSort	qualifier	visib
liveeditaddon	Attribute not found because the extension was de-released	type	ContentSlot	qualifier	lock
btg	Attribute not found because the extension was de-released	type	CMSSite	qualifier	segr
cmscockpit	Attribute not found because the extension was de-released	type	PreviewData	qualifier	liveE

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
liveeditaddon	Attribute not found because the extension was de-released	type	PreviewData	qualifier	devi
liveeditaddon	Attribute not found because the extension was de-released	type	PreviewData	qualifier	view
liveeditaddon	Attribute not found because the extension was de-released	type	PreviewData	qualifier	devi
secaddon	Attribute not found	type	SecChatComponent	qualifier	chat
secaddon	Attribute not found	type	SecChatComponent	qualifier	chat
secaddon	Attribute not found	type	SecChatComponent	qualifier	vide
secaddon	Attribute not found	type	SecChatComponent	qualifier	chat
cissapdigitalpayment	Attribute not found	type	SAPDigitalPaymentConfiguration	qualifier	bypa
apiregistryservices	Type not found	code	Endpoint	–	–
apiregistryservices	Type not found	code	AbstractDestination	–	–
apiregistryservices	Type not found	code	ExposedDestination	–	–
apiregistryservices	Type not found	code	ConsumedDestination	–	–
apiregistryservices	Type not found	code	AbstractCredential	–	–
apiregistryservices	Type not found	code	BasicCredential	–	–
apiregistryservices	Type not found	code	ExposedOAuthCredential	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
apiregistryservices	Type not found	code	ConsumedCertificateCredential	–	–
apiregistryservices	Type not found	code	EventConfiguration	–	–
apiregistryservices	Type not found	code	EventPropertyConfiguration	–	–
apiregistryservices	Type not found	code	EventMappingConstraint	–	–
sapcpconfiguration	Type not found	code	CecService	–	–
sapcpconfiguration	Type not found	code	CecTechnicalUser	–	–
sapcpconfiguration	Type not found	code	AbstractCecServiceMapping	–	–
sapcpconfiguration	Type not found	code	BaseSiteCecServiceMapping	–	–
commercesearch	Type not found because the extension was de-released	code	AbstractSolrFacetVisibilityRule	–	–
commercesearch	Type not found because the extension was de-released	code	FacetValueCountSolrFacetVisibilityRule	–	–
commercesearch	Type not found because the extension was de-released	code	OtherFacetValueCountSolrFacetVisibilityRule	–	–
commercesearch	Type not found because the extension was de-released	code	CategorySelectedSolrFacetVisibilityRule	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
commercesearch	Type not found because the extension was de-released	code	ValueCoverageSolrFacetVisibilityRule	–	–
commercesearch	Type not found because the extension was de-released	code	ConditionalSolrSort	–	–
commercesearch	Type not found because the extension was de-released	code	AbstractSolrSortCondition	–	–
commercesearch	Type not found because the extension was de-released	code	SelectedCategoryHierarchySolrSortCondition	–	–
commercesearch	Type not found because the extension was de-released	code	SolrHeroProductDefinition	–	–
commercesearch	Type not found because the extension was de-released	code	AbstractSolrSearchProfile	–	–
commercesearch	Type not found because the extension was de-released	code	GlobalSolrSearchProfile	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
commercesearch	Type not found because the extension was de-released	code	CategorySolrSearchProfile	–	–
commercesearch	Type not found because the extension was de-released	code	SolrFacetReconfiguration	–	–
commercesearch	Type not found because the extension was de-released	code	SolrBoostRule	–	–
b2badmincockpit	Type not found because the extension was de-released	code	B2BUnitActiveConstraint	–	–
btg	Type not found because the extension was de-released	code	BTGItem	–	–
btg	Type not found because the extension was de-released	code	BTGSegment	–	–
btg	Type not found because the extension was de-released	code	BTGOutputActionDefinition	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGAssignToGroupDefinition	–	–
btg	Type not found because the extension was de-released	code	CmsRestrictionActionDefinition	–	–
btg	Type not found because the extension was de-released	code	BtgSegmentRestriction	–	–
btg	Type not found because the extension was de-released	code	BTGRule	–	–
btg	Type not found because the extension was de-released	code	BTGAbstractResult	–	–
btg	Type not found because the extension was de-released	code	BTGSegmentResult	–	–
btg	Type not found because the extension was de-released	code	BTGRuleResult	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGCondition	–	–
btg	Type not found because the extension was de-released	code	BTGConditionResult	–	–
btg	Type not found because the extension was de-released	code	BTGExpression	–	–
btg	Type not found because the extension was de-released	code	BTGOperator	–	–
btg	Type not found because the extension was de-released	code	BTGOperand	–	–
btg	Type not found because the extension was de-released	code	BTGGenericOperand	–	–
btg	Type not found because the extension was de-released	code	ScriptMedia	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGAbstractScriptOperand	–	–
btg	Type not found because the extension was de-released	code	BTGMediaScriptOperand	–	–
btg	Type not found because the extension was de-released	code	BTGStringScriptOperand	–	–
btg	Type not found because the extension was de-released	code	BTGAbstractOrderOperand	–	–
btg	Type not found because the extension was de-released	code	BTGLastOrdersOperand	–	–
btg	Type not found because the extension was de-released	code	BTGLastOrdersPriceOperand	–	–
btg	Type not found because the extension was de-released	code	BTGNumberOfOrdersOperand	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGNumberOfOrdersRelativeDateOperand	–	–
btg	Type not found because the extension was de-released	code	BTGLastOrderDateOperand	–	–
btg	Type not found because the extension was de-released	code	BTGOrderTotalSumOperand	–	–
btg	Type not found because the extension was de-released	code	BTGEachOrderTotalSumOperand	–	–
btg	Type not found because the extension was de-released	code	BTGProductsInOrdersOperand	–	–
btg	Type not found because the extension was de-released	code	BTGCategoriesInOrdersOperand	–	–
btg	Type not found because the extension was de-released	code	BTGAbstractCartOperand	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGCartTotalSumOperand	–	–
btg	Type not found because the extension was de-released	code	BTGProductsInCartOperand	–	–
btg	Type not found because the extension was de-released	code	BTGCategoriesInCartOperand	–	–
btg	Type not found because the extension was de-released	code	BTGQuantityOfProductInCartOperand	–	–
btg	Type not found because the extension was de-released	code	BTGCartIsEmptyOperand	–	–
btg	Type not found because the extension was de-released	code	BTGAbstractCustomerOperand	–	–
btg	Type not found because the extension was de-released	code	BTGUserMemberOfGroupsOperand	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGUserGenderOperand	–	–
btg	Type not found because the extension was de-released	code	BTGUserAddressPostalCodeOperand	–	–
btg	Type not found because the extension was de-released	code	BTGUserCountryOperand	–	–
btg	Type not found because the extension was de-released	code	BTGAbstractWCMSOperand	–	–
btg	Type not found because the extension was de-released	code	BTGVisitedContentpagesOperand	–	–
btg	Type not found because the extension was de-released	code	BTGViewedProductsOperand	–	–
btg	Type not found because the extension was de-released	code	BTGViewedCategoriesOperand	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGReferalUrlOperand	–	–
btg	Type not found because the extension was de-released	code	BTGUrlParameterOperand	–	–
btg	Type not found because the extension was de-released	code	BTGAbstractReferenceOperand	–	–
btg	Type not found because the extension was de-released	code	BTGAbstractLiteralOperand	–	–
btg	Type not found because the extension was de-released	code	BTGIntegerLiteralOperand	–	–
btg	Type not found because the extension was de-released	code	BTGDoubleLiteralOperand	–	–
btg	Type not found because the extension was de-released	code	BTGStringLiteralOperand	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGBooleanLiteralOperand	–	–
btg	Type not found because the extension was de-released	code	BTGGenderEnumLiteralOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferencePriceOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceProductsOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceCategoriesOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceDateOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceExactDateOperand	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGReferenceContentpagesOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceRegExpListOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceKeyValuePairListOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceStringListOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferenceCountriesOperand	–	–
btg	Type not found because the extension was de-released	code	BTGReferencePrincipalGroupsOperand	–	–
btg	Type not found because the extension was de-released	code	BTGConfig	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Type not found because the extension was de-released	code	BTGResultCleaningCronJob	–	–
btgcockpit	Type not found because the extension was de-released	code	BtgJasperWidgetPreferences	–	–
liveeditaddon	Type not found because the extension was de-released	code	DeviceSupport	–	–
b2bbtgaddon	Type not found because the extension was de-released	code	OrganizationOrderStatistics	–	–
b2bbtgaddon	Type not found because the extension was de-released	code	OrganizationOrdersReportingCronJob	–	–
b2bbtgaddon	Type not found because the extension was de-released	code	BTGOrganizationTotalSpentInCurrencyLastYearOperand	–	–
b2bbtgaddon	Type not found because the extension was de-released	code	BTGOrganizationTotalSpentInCurrencyRelativeDatesOperand	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
b2bbtgaddon	Type not found because the extension was de-released	code	BTGAssignOrganizationToGroupDefinition	–	–
b2bbtgaddon	Type not found because the extension was de-released	code	BTGNumberOfOrdersAboveAmountRelativeDateOperand	–	–
b2bbtgaddon	Type not found because the extension was de-released	code	BTGOrganizationUIDOfUserOperand	–	–
b2bbtgaddon	Type not found because the extension was de-released	code	BTGOrganizationUIDsOperand	–	–
personalizationaddon	Type not found	code	PersonalizationScriptComponent	–	–
apiregistryservices	EnumType not found	code	DestinationChannel	–	–
apiregistryservices	EnumType not found	code	EventPriority	–	–
apiregistryservices	EnumType not found	code	EventMappingType	–	–
sapcpconfiguration	EnumType not found	code	ServiceClient	–	–
commercesearch	EnumType not found because the extension was de-released	code	FacetValueCountOperator	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
commercesearch	EnumType not found because the extension was de-released	code	FacetVisibilityRuleCondition	–	–
commercesearch	EnumType not found because the extension was de-released	code	FacetSelectedState	–	–
commercesearch	EnumType not found because the extension was de-released	code	SolrBoostConditionOperator	–	–
btg	EnumType not found because the extension was de-released	code	BTGConditionEvaluationScope	–	–
btg	EnumType not found because the extension was de-released	code	BTGUserAddressOperandMode	–	–
btg	EnumType not found because the extension was de-released	code	BTGEvaluationMethod	–	–
btg	EnumType not found because the extension was de-released	code	BTGRuleType	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	EnumType not found because the extension was de-released	code	BTGResultScope	–	–
cmscockpit	EnumType not found because the extension was de-released	code	LiveEditVariant	–	–
liveeditaddon	EnumType not found because the extension was de-released	code	DeviceOrientation	–	–
liveeditaddon	EnumType not found because the extension was de-released	code	CMSComponentAdminAction	–	–
liveeditaddon	EnumType not found because the extension was de-released	code	CMSComponentAdminActionGroup	–	–
liveeditaddon	EnumType not found because the extension was de-released	code	CMSMenuItemType	–	–
liveeditaddon	EnumType not found because the extension was de-released	code	MenuBannerSize	–	–
odata2webservicesfeaturetests	EnumType not found	code	OData2webservicesFeatureTestPropertiesTypes	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
sapmodel	EnumType Properties Change	code	SAPOrderStatus::((sapmodel))::YEnumType[sapmodel-items	changed properties	[dyn
processing	EnumType Value Change	type	ProcessState	values	SAP
integrationservices	Forbidden type-location change	code	InboundRequest	–	–
integrationservices	Forbidden type-location change	code	InboundRequestError	–	–
configurablebundleservices	Forbidden type-location change	code	AutoPickBundleSelectionCriteria	–	–
apiregistryservices	Relation not found	type	Endpoint2AbstractDestination	–	–
apiregistryservices	Relation not found	type	EventConfiguration2EventPropertyConfiguration	–	–
sapcpconfiguration	Relation not found	type	ServiceCecMappingRelation	–	–
sapcpconfiguration	Relation not found	type	CecTechnicalUserSiteMapping	–	–
commercesearch	Relation not found because the extension was de-released	type	SolrIndexedProperty2FacetVisibilityRules	–	–
commercesearch	Relation not found because the extension was de-released	type	SolrHeroProductDef2ProductRelation	–	–
commercesearch	Relation not found because the extension was de-released	type	SolrIndexedProperty2SolrFacetReconfiguration	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
commercesearch	Relation not found because the extension was de-released	type	SolrFacetReconfiguration2SolrSearchProfile	–	–
commercesearch	Relation not found because the extension was de-released	type	SolrIndexedProperty2SolrBoostRuleRelation	–	–
commercesearch	Relation not found because the extension was de-released	type	SolrBoostRule2SolrSearchProfile	–	–
btg	Relation not found because the extension was de-released	type	BTGSegmentToCMSSite	–	–
btg	Relation not found because the extension was de-released	type	BTGSegmentToBTGRulesRelation	–	–
btg	Relation not found because the extension was de-released	type	BTGSegmentToAbstractActionsRelation	–	–
btg	Relation not found because the extension was de-released	type	BTGSegmentBTGSegmentResultRelation	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Relation not found because the extension was de-released	type	UserBTGSegmentResultRelation	–	–
btg	Relation not found because the extension was de-released	type	BTGRuleToBTGConditionsRelation	–	–
btg	Relation not found because the extension was de-released	type	BTGConditionBTGConditionResultRelation	–	–
btg	Relation not found because the extension was de-released	type	UserBTGConditionResultRelation	–	–
btg	Relation not found because the extension was de-released	type	BTGRuleBTGRuleResultRelation	–	–
btg	Relation not found because the extension was de-released	type	UserBTGRuleResultRelation	–	–
btg	Relation not found because the extension was de-released	type	BTGAssignToGroupDefinitionGroup	–	–

Extension	Type of Change	Property 1	Value 1	Property 2	Valu
btg	Relation not found because the extension was de-released	type	BTGReferenceOperandToContentPages	–	–
btg	Relation not found because the extension was de-released	type	BTGReferenceOperandToProducts	–	–
btg	Relation not found because the extension was de-released	type	BTGReferenceOperandToCategories	–	–
btg	Relation not found because the extension was de-released	type	BTGReferenceOperandToCountries	–	–
btg	Relation not found because the extension was de-released	type	BTGReferenceOperandToPrincipalGroups	–	–
btg	Relation not found because the extension was de-released	type	RestrictionActionDefinitionToSegmentRestriction	–	–

Type Comparer Report of Modifications

The following table lists results from the Type Comparer Report. The first column lists the extensions that have been modified, while the remaining columns list the details of those modifications.

Extension Name	Severity: FATAL, ERROR, WARNING, INFO, DEBUG	Modification: ADDED, CHANGED, UPDATED, DEPRECATED,REMOVED	Type: ATOMICTYPE, COLLECTIONTYPE, ENUMTYPE, ENUMTYPE_VALUE, MAPTYPE, RELATION, RELATION_CUSTOMPROPERTY, ITEMTYPE, ITEMTYPE_CUSTOMPROPERTY, ITEMTYPE_ATTRIBUTE, ITEMTYPE_ATTRIBUTE_CUSTOMPROPERTY, ITEMTYPE_DEPLOYMENT	Detail 1 itemtyp
((scimservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	User
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((gigyaservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Custom
((scimservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	UserGrc
((sapcpiadapter))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Address
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Abstrac
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote

Extension Name	Severity: FATAL, ERROR, WARNING, INFO, DEBUG	Modification: ADDED, CHANGED, UPDATED, DEPRECATED,REMOVED	Type: ATOMICTYPE, COLLECTIONTYPE, ENUMTYPE, ENUMTYPE_VALUE, MAPTYPE, RELATION, RELATION_CUSTOMPROPERTY, ITEMTYPE, ITEMTYPE_CUSTOMPROPERTY, ITEMTYPE_ATTRIBUTE, ITEMTYPE_ATTRIBUTE_CUSTOMPROPERTY, ITEMTYPE_DEPLOYMENT	Detail 1 itemtyp
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Quote
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	QuoteEr
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	QuoteEr
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	QuoteEr
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	QuoteEr
((validation))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Decimal
((validation))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Decimal
((merchandisingservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Catalog
((sapcpiproductexchange))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	ClassAt
((sapproductconfigservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Classific
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Comme
((integrationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Integrat
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi

Extension Name	Severity: FATAL, ERROR, WARNING, INFO, DEBUG	Modification: ADDED, CHANGED, UPDATED, DEPRECATED,REMOVED	Type: ATOMICTYPE, COLLECTIONTYPE, ENUMTYPE, ENUMTYPE_VALUE, MAPTYPE, RELATION, RELATION_CUSTOMPROPERTY, ITEMTYPE, ITEMTYPE_CUSTOMPROPERTY, ITEMTYPE_ATTRIBUTE, ITEMTYPE_ATTRIBUTE_CUSTOMPROPERTY, ITEMTYPE_DEPLOYMENT	Detail 1 itemtyp
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((sapcpiadapter))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((saprevenuecloudcustomer))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCpi
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Consign
((basecommerce))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	PointOf
((gigyaloginaddon))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	BaseSit
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCor
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPCor
((saprevenuecloudpaddon))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	Paymen
((sapquoteintegration))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	QuotePr
((sapproductconfigservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	ERPVari
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPOrd
((saprevenuecloudorder))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPOrd
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxCusto
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxSegm
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxUserT
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxResul
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxConfi
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxConfi

Extension Name	Severity: FATAL, ERROR, WARNING, INFO, DEBUG	Modification: ADDED, CHANGED, UPDATED, DEPRECATED,REMOVED	Type: ATOMICTYPE, COLLECTIONTYPE, ENUMTYPE, ENUMTYPE_VALUE, MAPTYPE, RELATION, RELATION_CUSTOMPROPERTY, ITEMTYPE, ITEMTYPE_CUSTOMPROPERTY, ITEMTYPE_ATTRIBUTE, ITEMTYPE_ATTRIBUTE_CUSTOMPROPERTY, ITEMTYPE_DEPLOYMENT	Detail 1 itemtyp
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxConfi
((personalizationservices))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	CxConfi
((saprevenuecloudproduct))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	UsageC
((saprevenuecloudproduct))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	UsageC
((saprevenuecloudproduct))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPRev
((saprevenuecloudcustomer))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	B2BUni
((secaddon))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SecCha
((cissapdigitalpayment))	INFO	ADDED	ITEMTYPE_ATTRIBUTE	SAPDigi
((integrationservices))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	Inbounc
((integrationservices))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	Inbounc
((integrationservices))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	Inbounc
((integrationservices))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	Inbounc
((integrationservices))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	Inbounc
((integrationservices))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	Inbounc
((integrationservices))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	Inbounc
((saprevenuecloudproduct))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	PerUnit

Extension Name	Severity: FATAL, ERROR, WARNING, INFO, DEBUG	Modification: ADDED, CHANGED, UPDATED, DEPRECATED,REMOVED	Type: ATOMICTYPE, COLLECTIONTYPE, ENUMTYPE, ENUMTYPE_VALUE, MAPTYPE, RELATION, RELATION_CUSTOMPROPERTY, ITEMTYPE, ITEMTYPE_CUSTOMPROPERTY, ITEMTYPE_ATTRIBUTE, ITEMTYPE_ATTRIBUTE_CUSTOMPROPERTY, ITEMTYPE_DEPLOYMENT	Detail 1 itemtyp
((saprevenuecloudproduct))	INFO	CHANGED	ITEMTYPE_ATTRIBUTE	PerUnit
((integrationservices))	INFO	ADDED	ITEMTYPE	Monitor
((integrationservices))	INFO	ADDED	ITEMTYPE	Monitor
((scimservices))	INFO	ADDED	ITEMTYPE	ScimUs
((inboundservices))	INFO	ADDED	ITEMTYPE	Inbound
((outboundservices))	INFO	ADDED	ITEMTYPE	Outbou
((outboundservices))	INFO	ADDED	ITEMTYPE	Outbou
((sapcpiadapter))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapcpiadapter))	INFO	ADDED	ITEMTYPE	SAPCpi
((ruleengine))	INFO	ADDED	ITEMTYPE	DroolsK
((gigyaservices))	INFO	ADDED	ITEMTYPE	GigyaCo
((gigyaservices))	INFO	ADDED	ITEMTYPE	GigyaFi
((personalizationservices))	INFO	ADDED	ITEMTYPE	CxDefau
((saprevenuecloudproduct))	INFO	ADDED	ITEMTYPE	SapReve
((sapcpireturnsexchange))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapcpireturnsexchange))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapcpireturnsexchange))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapproductconfigcpiorderexchange))	INFO	ADDED	ITEMTYPE	SAPCpi

Extension Name	Severity: FATAL, ERROR, WARNING, INFO, DEBUG	Modification: ADDED, CHANGED, UPDATED, DEPRECATED,REMOVED	Type: ATOMICTYPE, COLLECTIONTYPE, ENUMTYPE, ENUMTYPE_VALUE, MAPTYPE, RELATION, RELATION_CUSTOMPROPERTY, ITEMTYPE, ITEMTYPE_CUSTOMPROPERTY, ITEMTYPE_ATTRIBUTE, ITEMTYPE_ATTRIBUTE_CUSTOMPROPERTY, ITEMTYPE_DEPLOYMENT	Detail 1 itemtyp
((merchandisingservices))	INFO	ADDED	ITEMTYPE	MerchIn
((merchandisingservices))	INFO	ADDED	ITEMTYPE	Abstrac
((merchandisingservices))	INFO	ADDED	ITEMTYPE	MerchP
((merchandisingservices))	INFO	ADDED	ITEMTYPE	MerchIn
((merchandisingservices))	INFO	ADDED	ITEMTYPE	MerchS
((sapquoteintegration))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapquoteintegration))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapquoteintegration))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapquoteintegration))	INFO	ADDED	ITEMTYPE	SAPCpi
((sapquoteintegration))	INFO	ADDED	ITEMTYPE	SAPCpi
((merchandisingaddon))	INFO	ADDED	ITEMTYPE	Merchar
((gigyaloginaddon))	INFO	ADDED	ITEMTYPE	Abstrac
((gigyaloginaddon))	INFO	ADDED	ITEMTYPE	GigyaRa
((backoffice))	INFO	ADDED	ITEMTYPE	EnumCo
((platformbackoffice))	INFO	ADDED	ITEMTYPE	HybrisE
((acceleratorcms))	INFO	DEPRECATED	ITEMTYPE	CMSUiE
((acceleratorcms))	INFO	DEPRECATED	ITEMTYPE	CMSAct

Spring Framework Changes

The following table lists Spring framework changes for SAP Commerce between 1811 and 1905.

No.	Bean ID	Previou
1	org.springframework.context.annotation.internalRequiredAnnotationProcessor	<bean b beanId=
2	de.hybris.platform.test.MyUnit	<bean b spring.>

No.	Bean ID	Previou
3	auditableSaver	<bean b resourc name='
4	testCronJobFactory	<bean b resourc
5	testSimpleDistributedProcessHandler	<bean b beanRe xmlns=
6	multiValueTestSimpleDistributedProcessHandler	<bean b beanId= parentN
7	completelyFailingTestSimpleDistributedProcessHandler	<bean b beanId= parentN
8	sometimesFailingTestSimpleDistributedProcessHandler	<bean b beanId= parentN
9	testImportCsvUtil	<bean b resourc
10	localizedConstraintsRegistry	<bean b beanRe name='
11	workflowDao:beanResourceDescription	<bean b [cockpi name='
12	workflowDao	<bean b [cockpi name='
13	:name	<proper
14	linkEntryDao	<bean b [mcc-sp name='
15	defaultLinkEntryService	<bean b path res name='
16	classificationImportJobPreparer	<bean b beanRe
17	classificationImportJobPrepareMapping	<bean b beanRe name='
18	org.springframework.integration.expression.IntegrationEvaluationContextAwareBeanPostProcessor#0	<bean b beanId= xmlns=

No.	Bean ID	Previou
19	listMergeDirective$child#0:beanResourceDescription	<bean b scope=
20	consumedDestinationRemoveInterceptor	<bean b beanId= xmlns=
21	consumedDestinationValidateInterceptor	<bean b beanId= xmlns=
22	consumedOAuthCredentialRemoveInterceptor	<bean b beanId= xmlns=
23	consumedOAuthCredentialValidateInterceptor	<bean b beanId= xmlns=
24	defaultIntegrationObjectService	<bean b beanRe name='
25	defaultAbstractIntegrationKeyGenerator	<bean b beanId= xmlns=
26	defaultMapToIntegrationKeyGenerator:beanClassName	<bean b beanId= parentN name='
27	defaultMapToIntegrationKeyGenerator:parentName	<bean b beanId= parentN name='
28	defaultMapToIntegrationKeyGenerator	<bean b beanId= parentN name='
29	:name	<proper
30	:name	<proper
31	:name	<proper
32	defaultAbstractRestTemplateCreator	<bean b beanRe name='

No.	Bean ID	Previou
33	defaultOutboundServiceFacade	<bean b beanRe name=" name=" name="
34	:name	<proper
35	:name	<proper
36	:name	<proper
37	:name	<proper
38	defaultOutboundServiceFacade	<proper
39	defaultOutboundHttpClient	<bean b [outbou
40	defaultOutboundHttpClient	<bean b [outbou
41	defaultOutboundHttpClient:beanId	<bean b [outbou
42	defaultOutboundHttpClient:beanResourceDescription	<bean b [outbou
43	defaultOutboundHttpClient:scope	<bean b [outbou
44	defaultCertificateService	<bean b beanRe <proper
45	kymaEventRestTemplateWrapper	<bean b beanRe <proper
46	:name	<proper
47	kymaDestinationRestTemplateWrapper	<bean b beanRe <proper
48	:name	<proper

No.	Bean ID	Previou
49	:name	<proper
50	defaultStorageRequestFactory	<bean b beanRe <proper
51	:name	<proper
52	:name	<proper
53	defaultStorageRequestFactory	<proper
54	defaultItemLookupRequestFactory	<bean b beanRe <proper name='
55	defaultODataEntryToIntegrationKeyGenerator	<bean b beanId= parentN
56	sapCpiProperties	<bean b xmlns= name='
57	sapCpiOAuthDefaultClient	<bean b [sapcpi name='
58	sapCpiCustomerDefaultClient	<bean b [sapcpi name='
59	sapCpiOrderDefaultClient	<bean b [sapcpi name='
60	sapCpiDefaultPropertiesResolver	<bean b resourc
61	sapCpiOAuthDefaultService	<bean b beanRe name='
62	sapCpiDefaultOutboundRequestDecorator	<bean b beanId= xmlns=
63	sapCpiDefaultOutboundServiceFacade	<bean b parentN </bean

No.	Bean ID	Previou
64	sapCpiProductFeaturePersistenceHook	<bean b beanId= xmlns= name=' </bean
65	:name	<prope
66	:name	<prope
67	sapCpiProductFeaturePersistenceHook	<prope
68	sapCpiProductFeaturePersistenceHook	<prope
69	defaultImageMagickService	<bean b beanRe name='
70	defaultreportcockpitSystemSetup	<bean b beanRe name='
71	initiateYaasConfigurationY2ySyncExecutionService	<bean b beanId= parentN </bean
72	defaultConfigurationSyncServicee	<bean b beanId= xmlns= name='
73	defaultExcelJavaTypeTranslator	<bean b beanRe <prope name='
74	defaultExcelJavaTypeTranslator:scope	<bean b beanRe <prope name='
75	defaultExcelJavaTypeTranslator	<bean b beanRe <prope name='
76	defaultAbstractExcelMediaImportTranslator	<bean b beanRe <prope name='

No.	Bean ID	Previou
77	defaultAbstractExcelMediaImportTranslator:scope	<bean b beanRe <proper name='
78	defaultAbstractExcelMediaImportTranslator	<bean b beanRe <proper name='
79	:name	<proper
80	:name	<proper
81	:name	<proper
82	:name	<proper
83	defaultExcelEnumTypeTranslator	<bean b beanRe <proper
84	defaultExcelEnumTypeTranslator:scope	<bean b beanRe <proper
85	defaultExcelEnumTypeTranslator	<bean b beanRe <proper
86	defaultAbstractCatalogVersionAwareTranslator	<bean b beanId= scope=
87	defaultAbstractCatalogVersionAwareTranslator:scope	<bean b beanId= scope=
88	defaultAbstractCatalogVersionAwareTranslator	<bean b beanId= scope=
89	defaultExcelCatalogVersionTypeTranslator	<bean b beanRe scope= name='
90	:name	<proper

No.	Bean ID	Previou
91	defaultExcelCatalogVersionTypeTranslator	<proper
92	defaultExcelProductSupercategoriesTypeTranslator	<bean b beanId= parentN <proper
93	:name	<proper
94	:name	<proper
95	defaultExcelProductSupercategoriesTypeTranslator	<proper
96	defaultExcelEurope1PricesTypeTranslator	<bean b beanRe <proper
97	defaultExcelEurope1PricesTypeTranslator:scope	<bean b beanRe <proper
98	defaultExcelEurope1PricesTypeTranslator	<bean b beanRe <proper
99	defaultExcelPlatformValidationEngineAwareLocalizedStrategy	<bean b beanId= spring.x name='
100	defaultExcelClassificationReferenceValidator	<bean b beanId= xmlns= </bean
101	defaultExcelGenericReferenceValidator	<bean b beanRe <proper
102	defaultBackofficeConfigurationSystemSetup	<bean b beanRe name='
103	catalogVersionCompareService	<bean b beanId= xmlns= name='
104	abstractTaxDiscountValueParser	<bean b beanRe name=' name='

No.	Bean ID	Previou
105	defaultAsConfigurationPartOfCleanupInterceptor	<bean b beanId= parentN
106	asConfigurationPartOfCleanupInterceptorMapping	<bean b beanRe name='
107	defaultPromotionsService	<bean b path res
108	defaultCmsRelatedItemsCollector:beanResourceDescription	<bean b beanRe xmlns=
109	defaultCmsItemVisitorRegistry:beanResourceDescription	<bean b beanRe xmlns= </bean
110	defaultCmsVersionService	<bean b beanRe name=' name=' name=' name=' name=' name='
111	cmsVersionGCProcessDefinitionResource:beanResourceDescription	<bean b beanRe <proper
112	defaultAbstractCMSVersionGCAction:beanResourceDescription	<bean b beanId= parentN name='
113	defaultCollectRelatedCMSVersionsGCProcessAction:beanResourceDescription	<bean b beanId= parentN
114	defaultCollectRetainableCMSVersionsGCProcessAction:beanResourceDescription	<bean b beanId= parentN
115	defaultRemoveCMSVersionsGCProcessAction:beanResourceDescription	<bean b beanId= parentN </bean
116	defaultCmsVersionGCProcessService:beanResourceDescription	<bean b beanId= xmlns= name='

No.	Bean ID	Previou
117	defaultCmsVersionGCProcessPredicate:beanResourceDescription	<bean b beanId= xmlns=
118	cmsVersionGCProcessThreadPoolTaskExecutor:beanResourceDescription	<bean b beanRe <prope
119	cmsVersionGCProcessThreadFactory:beanResourceDescription	<bean b path res
120	defaultCMSItemAttributeToDataConverter	<bean b beanId= xmlns= <prope
121	cmsComponentTypePredicate:beanClassName	<bean b beanRe <prope
122	cmsComponentTypePredicate:beanResourceDescription	<bean b beanRe <prope
123	abstractTypePredicate:beanClassName	<bean b beanRe <prope
124	abstractTypePredicate:beanResourceDescription	<bean b beanRe <prope
125	defaultCMSPageService:parentName	<bean b beanRe xmlns= name=' name=' name=' name=' name='
126	defaultCMSPageService	<bean b beanRe xmlns= name=' name=' name=' name=' name='
127	:name	<prope
128	:name	<prope
129	:name	<prope
130	:name	<prope

No.	Bean ID	Previou
131	:name	<proper
132	:name	<proper
133	:name	<proper
134	:name	<proper
135	:name	<proper
136	defaultCMSPageService	<proper
137	defaultCMSPageService	<proper
138	defaultCMSPageService	<proper
139	defaultCMSPageService	<proper
140	defaultCMSPageService	<proper
141	defaultCMSPageService	<proper
142	defaultCMSAdminPageService	<bean b beanRe xmlns= name=' name='
143	baseStoreTimeRestrictionDescriptionHandler:beanClassName	<bean b beanRe
144	textService	<bean b [mobile name='
145	messageHelper	<bean b [mobile
146	actionHelper	<bean b path res <proper name='

No.	Bean ID	Previou
147	velocityTemplateHelper	<bean b path res
148	loggingFilter	<bean b path res
149	smsService	<bean b resourc name='
150	phoneListService	<bean b beanRe name='
151	phoneNumberService	<bean b beanRe name='
152	phoneNumberValidator	<bean b path res </bean
153	phoneNumberInterceptorMapping	<bean b beanRe name='
154	thumbnailGenerator	<bean b path res
155	code2dService	<bean b resourc
156	barcodeMediaService	<bean b beanRe name='
157	detectionService	<bean b path res
158	mobileSendMessagePerformable	<bean b beanRe name='
159	mobileservices.manager	<bean b resourc name='
160	mobileServicesCustomerCleanupHookMergeDirective	<bean b parentN
161	mobileServicesCustomerCleanupHook	<bean b beanId= xmlns=

No.	Bean ID	Previou
162	incomingSMSMessageGateway	<bean b beanRe name='
163	messageScheduler	<bean b beanRe name='
164	smsEngineRegistry	<bean b beanRe name='
165	incomingMessageProcessor	<bean b beanRe xmlns= name='
166	incomingMessageRoutingStrategy	<bean b beanId= xmlns=
167	receiveTask	<bean b resourc name='
168	outgoingMessageProcessor	<bean b beanRe xmlns= name='
169	sendTask	<bean b [mobile name='
170	outgoingMessageRoutingStrategy	<bean b beanId= xmlns=
171	messagePricingStrategy	<bean b beanRe name='
172	abstractProcessor	<bean b beanRe name='
173	abstractProcessingTask	<bean b beanRe name='
174	httpEngine	<bean b resourc
175	smppEngine	<bean b resourc name='
176	mBloxSmppEngine	<bean b path res <proper

No.	Bean ID	Previou
177	mBloxProviderStrategy	<bean b beanRe
178	abstractSMSEngine	<bean b beanRe <proper
179	mobileSendItemLinkAction	<bean b beanRe xmlns=
180	mobileSendLinkAction	<bean b beanRe xmlns=
181	mobileSendTextAction	<bean b beanRe xmlns=
182	mobileReceiveAndSendTextAction	<bean b beanId= parentN
183	mobileReceiveAndSendItemAction	<bean b beanId= parentN
184	mobileReceiveAndSubscribeAction	<bean b beanId= parentN
185	abstractMobileAction	<bean b beanRe name='
186	testSendSMSEngine	<bean b resourc <proper
187	brokenTestSendSMSEngine	<bean b beanRe xmlns=
188	unavailableTestSendSMSEngine	<bean b beanRe xmlns=
189	dataModelConsitencyChecker	<bean b beanRe <proper
190	helloWorldText	<bean b resourc <proper
191	brokenHelloWorldText	<bean b beanRe xmlns=

No.	Bean ID	Previou
192	mobileValidationService	<bean b beanRe <proper
193	mobileActionAssignmentRemoveInterceptor	<bean b beanId= xmlns=
194	mobileReceivingActionRemoveInterceptor	<bean b beanId= xmlns=
195	mobileActionKeywordRemoveInterceptor	<bean b beanId= xmlns=
196	phoneNumberListRemoveInterceptor	<bean b beanId= xmlns=
197	mobileShortcodeRemoveInterceptor	<bean b beanId= xmlns=
198	mobileAggregatorRemoveInterceptor	<bean b beanId= xmlns=
199	phoneNumberRemoveInterceptor	<bean b beanRe <proper
200	mobileSendMessageCronJobValidator	<bean b beanId= xmlns=
201	mobileSendMessageCronJobInterceptorMapping	<bean b beanRe <proper
202	assignmentRemoveInterceptorMapping	<bean b beanRe <proper
203	receivingActionRemoveInterceptorMapping	<bean b beanRe <proper
204	actionKeywordRemoveInterceptorMapping	<bean b beanRe <proper
205	phoneNumberListRemoveInterceptorMapping	<bean b beanRe <proper
206	mobileShortcodeRemoveInterceptorMapping	<bean b beanRe <proper

No.	Bean ID	Previou
207	mobileAggregatorRemoveInterceptorMapping	<bean b beanRe <proper
208	phoneNumberRemoveInterceptorMapping	<bean b beanRe <proper
209	mobileReceiveAndSendVoucherAction	<bean b beanId= parentN
210	defaultYaasConfigurationService	<bean b beanRe name='
211	charonFactory	<bean b [yaascc
212	yaasInterceptor	<bean b path res name='
213	yaasServiceInterceptorMapping	<bean b beanRe name='
214	yaasClientCredentialInterceptorMapping	<bean b beanRe name='
215	defaultYaasServiceFactory	<bean b beanRe name='
216	defaultBaseSiteClientCredentialLocator	<bean b beanId= xmlns= name='
217	defaultBaiduMapsServiceWrapper	<bean b beanRe xmlns=
218	defaultBaiduMapTools	<bean b path res name='
219	defaultFirstPlacemarkWinsBaiduResponseParser	<bean b beanId= scope=
220	defaultGeoReponseParserProviders	<bean b xmlns=
221	chineseTaxInvoiceService	<bean b beanRe comme scope= name='

No.	Bean ID	Previou
222	chineseTaxInvoiceDao	<bean b beanRe xmlns=
223	paymentInfoCreator	<bean b beanRe name='
224	:name	<proper
225	paymentInfoCreator	<proper
226	defaultBackofficeFacetSearchConfigCache	<bean b beanId= xmlns=
227	listMergeDirective$child#1:beanResourceDescription	<bean b xmlns=
228	defaultRuleEngineCalculationService	<bean b beanId= xmlns= <proper name='
229	abstractRuleExecutableSupport	<bean b beanId= xmlns= <proper name='
230	defaultRuleOrderEntryGroupFixedDiscountRAOAction	<bean b beanId= parentN
231	defaultRuleOrderThresholdPerfectPartnerConditionTranslator	<bean b beanId= xmlns=
232	defaultRuleOrderThresholdPerfectPartnerConditionTranslator:scope	<bean b beanId= xmlns=
233	defaultRuleOrderThresholdPerfectPartnerConditionTranslator	<bean b beanId= xmlns=
234	defaultRuleCartTotalConditionTranslator	<bean b beanRe
235	defaultRuleCartTotalConditionTranslator:scope	<bean b beanRe

No.	Bean ID	Previou
236	defaultRuleCartTotalConditionTranslator	<bean b beanRe
237	defaultRuleQualifyingProductsConditionTranslator	<bean b beanId= xmlns=
238	defaultRuleQualifyingProductsConditionTranslator:scope	<bean b beanId= xmlns=
239	defaultRuleQualifyingProductsConditionTranslator	<bean b beanId= xmlns=
240	defaultQualifyingCategoriesConditionTranslator	<bean b beanId= xmlns=
241	defaultQualifyingCategoriesConditionTranslator:scope	<bean b beanId= xmlns=
242	defaultQualifyingCategoriesConditionTranslator	<bean b beanId= xmlns=
243	defaultRuleTargetCustomersConditionTranslator	<bean b beanId= xmlns=
244	defaultRuleTargetCustomersConditionTranslator:scope	<bean b beanId= xmlns=
245	defaultRuleTargetCustomersConditionTranslator	<bean b beanId= xmlns=
246	defaultRuleProductPriceConditionTranslator	<bean b beanId= xmlns=
247	defaultRuleProductPriceConditionTranslator:scope	<bean b beanId= xmlns=
248	defaultRuleProductPriceConditionTranslator	<bean b beanId= xmlns=
249	defaultRuleCustomerSupportConditionTranslator	<bean b beanId= xmlns=
250	defaultRuleCustomerSupportConditionTranslator:scope	<bean b beanId= xmlns=

No.	Bean ID	Previou
251	defaultRuleCustomerSupportConditionTranslator	<bean b beanId= xmlns=
252	defaultRuleEntryGroupTypeConditionTranslator	<bean b beanId= xmlns=
253	defaultRuleEntryGroupTypeConditionTranslator:scope	<bean b beanId= xmlns=
254	defaultRuleEntryGroupTypeConditionTranslator	<bean b beanId= xmlns=
255	ruleOrderEntryGroupFixedDiscountAction	<bean b beanRe name='
256	defaultProductsConditionResolutionStrategy	<bean b beanId= scope=
257	defaultCatConditionResolutionStrategy	<bean b beanId= xmlns=
258	defaultQualifyingCategoryPotentialPromotionMessageActionSupplementStrategy	<bean beanCl beanId= [promo
259	defaultQualifyingProductPotentialPromotionMessageActionSupplementStrategy	<bean beanCl beanId= [promo
260	defaultProductModelUrlResolver	<bean b beanRe name=' name='
261	de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping#16:beanResourceDescription	<bean b beanId= spring.x name='
262	de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping#16	<bean b beanId= spring.x name='
263	de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping#16	<prope

No.	Bean ID	Previou
264	defaultSetupSyncJobService	<bean b beanRe name=" name=" </bean
265	cxConsentWithdrawnEventListener	<bean b beanId= parentN name="
266	:name	<proper
267	cxAnonymousConsentChangeEventListener	<bean b beanId= parentN name="
268	defaultCxActionResultService	<bean b beanRe <proper <proper name="
269	defaultCxSegmentService	<bean b beanRe <proper name=" name="
270	:name	<proper
271	:name	<proper
272	:name	<proper
273	:name	<proper
274	:name	<proper
275	:name	<proper
276	:name	<proper
277	:name	<proper
278	defaultCxSegmentService	<proper

No.	Bean ID	Previou
279	defaultCxIntegrationMappingService	<bean b beanId= xmlns= name=' name='
280	mapMergeDirective$child#1:beanResourceDescription	<bean b parentN </bean
281	defaultCxCatalogConsumptionLayerMapper	<bean b beanId= xmlns= name='
282	defaultCxProductConsumptionLayerMapper	<bean b beanId= xmlns= name='
283	defaultCxLocationConsumptionLayerMapper	<bean b beanId= xmlns= name='
284	cxScriptMapperInterceptor	<bean b beanRe name='
285	defaultCxProcessParameterConsentReferenceStrategy	<bean b beanId= parentN <proper
286	consentReferenceEventDataHandler	<bean b beanRe name='
287	consentReferenceEventDataProviderListMergeDirective	<bean b parentN
288	defaultCxProfileIdentifierStrategy	<bean b beanId= xmlns= name='
289	:name	<proper
290	cdsCxOccInterceptor	<bean b path res <proper
291	:name	<proper
292	:name	<proper

No.	Bean ID	Previou
293	listMergeDirective$child#13:beanResourceDescription	<bean b parentN name='
294	listMergeDirective$child#13	<bean b parentN name='
295	:name	<proper
296	listMergeDirective$child#13	<proper
297	modifyPopulatorList$child#0:beanResourceDescription	<bean b parentN </bean
298	modifyPopulatorList$child#1:beanResourceDescription	<bean b parentN </bean
299	kymaIntegrationSampleDataSystemSetup	<bean b beanId= xmlns=
300	secCustomerValidateInterceptorMapping:beanResourceDescription	<bean b beanRe name='
301	defaultGigyaLoginFacade	<bean b beanRe name=' <proper name='
302	sapProductConfigDefaultCharonFacade	<bean b beanRe <proper <proper <proper
303	:name	<proper
304	sapProductConfigDefaultServiceVersionProvider	<bean b beanId= xmlns=
305	sapProductConfigDefaultCPSConfigurationPopulator	<bean b beanId= scope= <proper
306	sapProductConfigDefaultCPSConfigurationPopulator	<proper
307	:name	<proper

No.	Bean ID	Previou
308	:name	<proper
309	:name	<proper
310	sapProductConfigDefaultCPSKnowledgeBaseContainerCacheValueLoader	<bean b beanId= [sappro <proper
311	:name	<proper
312	sapProductConfigDefaultCPSKnowledgeBaseHeadersCacheValueLoader	<bean b beanId= [sappro
313	:name	<proper
314	sapProductConfigDefaultCharonPricingFacade	<bean b beanId= xmlns=
315	:name	<proper
316	:name	<proper
317	defaultBundleCommerceRuleService	<bean b beanId= xmlns= name=" name=" <proper
318	:name	<proper
319	:name	<proper
320	:name	<proper
321	:name	<proper
322	:name	<proper
323	defaultBundleCommerceRuleService	<proper

No.	Bean ID	Previou
324	de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping#17:beanResourceDescription	<bean b beanId= spring.x
325	de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping#17	<bean b beanId= spring.x
326	de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping#18:beanResourceDescription	<bean b beanId= spring.x
327	defaultBundleOrderEntryModifiableChecker	<bean b beanId= xmlns=
328	defaultAbstractBundleComponentEditableChecker	<bean b beanId= xmlns=
329	defaultCartBundleComponentEditableChecker	<bean b beanId= parentN name='
330	:name	<proper
331	defaultCartBundleComponentEditableChecker	<proper
332	org.springframework.beans.factory.annotation.RequiredAnnotationBeanPostProcessor#0	<bean b beanId= [cockpi
333	loginService	<bean b [cockpi name='
334	systemService	<bean b [cockpi
335	scopedTarget.userRightsCache	<bean b path res name='
336	userRightsCache	<bean b defined name='
337	newItemService	<bean b path res <proper name='

No.	Bean ID	Previou
338	savedValuesService	<bean b resourc
339	extensibleCockpitTypeService	<bean b beanRe <proper <proper
340	defaultCockpitPropertyService	<bean b beanRe
341	defaultCockpitTypeLoaderChain	<bean b beanRe <proper
342	defaultClassificationPropertyPathResolver	<bean b beanId= xmlns=
343	baseTypeCache	<bean b [cockpi
344	baseTypePropertyCache	<bean b path res
345	scopedTarget.extendedTypeCache	<bean b beanRe <attribu
346	extendedTypeCache	<bean b defined name='
347	scopedTarget.objectTemplateCache	<bean b beanRe <attribu
348	objectTemplateCache	<bean b defined name='
349	genericItemValueHandler	<bean b beanRe <proper <proper
350	classificationValueHandler	<bean b beanRe <proper <proper
351	widgetparamValueHandler	<bean b beanRe <proper
352	valueService	<bean b resourc <proper

No.	Bean ID	Previou
353	cockpitModelHelper	<bean b resourc <proper <proper
354	BrowserModelFactory	<bean b beanRe
355	AbstractPageableBrowserModel	<bean b beanRe <proper
356	valueHandlerRegistry	<bean b beanRe <proper
357	genericItemValueHandlerMapping	<bean b beanRe <proper
358	classificationValueHandlerMapping	<bean b beanRe <proper
359	widgetparamValueHandlerMapping	<bean b beanRe <proper
360	defaultSearchService	<bean b beanRe <proper
361	genericSearchProvider	<bean b beanRe <proper
362	conditionTranslatorRegistry	<bean b beanRe
363	genericQueryConditionTranslator	<bean b beanRe
364	referenceGenericQueryConditionTranslator	<bean b beanId= xmlns=
365	featureGenericQueryConditionTranslator	<bean b beanId= xmlns=
366	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#0	<bean b beanId= resourc <proper
367	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#1	<bean b beanId= resourc <proper

No.	Bean ID	Previou
368	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#2	<bean b beanId= resourc <proper
369	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#3	<bean b beanId= resourc <proper
370	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#4	<bean b beanId= resourc <proper
371	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#5	<bean b beanId= resourc <proper
372	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#6	<bean b beanId= resourc <proper
373	specialDateConditionTranslator	<bean b beanRe
374	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#7	<bean b beanId= resourc <proper
375	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#8	<bean b beanId= resourc <proper
376	de.hybris.platform.cockpit.services.search.impl.ConditionTranslatorRegistry.ConditionTranslatorMapping#9	<bean b beanId= resourc <proper
377	orderEntryLabelProvider	<bean b path res
378	categoryLabelProvider	<bean b beanRe
379	currencyLabelProvider	<bean b beanRe
380	countryLabelProvider	<bean b path res

No.	Bean ID	Previou
381	enumerationValueLabelProvider	<bean b beanRe
382	mediaLabelProvider	<bean b path res
383	productLabelProvider	<bean b path res
384	productReferenceLabelProvider	<bean b beanRe
385	unitLabelProvider	<bean b resourc
386	priceRowLabelProvider	<bean b beanRe
387	languageLabelProvider	<bean b beanRe
388	catalogVersionLabelProvider	<bean b beanRe
389	workflowLabelProvider	<bean b beanRe
390	workflowTemplateLabelProvider	<bean b beanRe
391	workflowActionTemplateLabelProvider	<bean b beanRe
392	commentLabelProvider	<bean b beanRe
393	commentAttachmentLabelProvider	<bean b beanRe
394	principalLabelProvider	<bean b beanRe
395	catalogAwareModelLabelProvider	<bean b beanRe xmlns=

No.	Bean ID	Previou
396	catalogVersionModelLabelProvider	<bean b beanRe xmlns=
397	categoryModelLabelProvider	<bean b beanRe xmlns=
398	commentAttachmentModelLabelProvider	<bean b beanId= parentN
399	commentModelLabelProvider	<bean b beanRe xmlns=
400	countryModelLabelProvider	<bean b beanRe xmlns=
401	currencyModelLabelProvider	<bean b beanRe xmlns=
402	enumerationValueModelLabelProvider	<bean b beanRe xmlns=
403	languageModelLabelProvider	<bean b beanRe xmlns=
404	mediaModelLabelProvider	<bean b beanRe xmlns=
405	orderModelLabelProvider	<bean b beanRe xmlns=
406	priceRowModelLabelProvider	<bean b beanRe xmlns=
407	principalModelLabelProvider	<bean b beanRe xmlns=
408	productModelLabelProvider	<bean b beanRe xmlns=
409	productReferenceModelLabelProvider	<bean b beanRe xmlns=
410	unitModelLabelProvider	<bean b beanRe xmlns=

No.	Bean ID	Previou
411	workflowActionTemplateModelLabelProvider	<bean b beanId= parentN
412	workflowModelLabelProvider	<bean b beanRe xmlns=
413	workflowTemplateModelLabelProvider	<bean b beanRe xmlns=
414	labelService	<bean b resourc <proper
415	mediaUpdateService	<bean b path res
416	objectCompareService	<bean b beanRe
417	uiConfigurationService	<bean b beanRe <proper <proper name='
418	defaultUIConfigurationPersistingStrategy	<bean b beanId= xmlns=
419	dashboardPersistingStrategy	<bean b beanRe xmlns=
420	editorPersistingStrategy	<bean b beanRe xmlns=
421	advancedSearchPersistingStrategy	<bean b beanId= parentN <proper
422	listViewPersistingStrategy	<bean b beanRe xmlns=
423	scopedTarget.uiComponentCache	<bean b beanRe <attribu
424	uiComponentCache	<bean b defined name='
425	cockpitUIComponentConfigurationDao	<bean b beanRe

No.	Bean ID	Previou
426	jaxbBasedUIComponentConfigurationFactory	<bean b beanId= xmlns= name='
427	listViewConfigurationFactory	<bean b beanRe xmlns= name='
428	editorConfigurationFactory	<bean b beanRe xmlns= name='
429	advancedSearchConfigurationFactory	<bean b beanRe xmlns= name='
430	baseConfigurationFactory	<bean b beanRe xmlns= name='
431	gridViewConfigurationFactory	<bean b beanRe xmlns=
432	wizardConfigurationFactory	<bean b beanRe xmlns= name='
433	widgetDashboardConfigurationFactory	<bean b beanId= parentN <prope
434	uiAccessRightService	<bean b beanRe <prope
435	cockpitUIComponentAccessRightDao	<bean b beanRe <prope
436	defaultCockpitJaxbContextCache	<bean b beanRe
437	dragAndDropWrapperService	<bean b beanRe <prope
438	deprecatedWorkflowService	<bean b beanRe <prope </bean

No.	Bean ID	Previou
439	defaultSavedQueryDao	<bean b beanRe <proper
440	savedQueryUserRightsDao	<bean b beanRe xmlns=
441	objectCollectionService	<bean b beanRe <proper
442	savedQueryService	<bean b beanRe <proper <proper
443	dynamicQueryService	<bean b beanRe
444	scopedTarget.undoManager	<bean b beanRe <attribu
445	undoManager	<bean b in class name='
446	mediaInfoService	<bean b path res <proper
447	modelSaveDataLanguageInterceptor	<bean b beanRe
448	valueHandlerDataLanguagePointcut	<bean b xmlns=
449	modelHelperDataLanguagePointcut	<bean b xmlns=
450	newItemDataLanguagePointcut	<bean b xmlns=
451	org.springframework.aop.aspectj.AspectJPointcutAdvisor#4	<bean b scope=
452	validationServiceDataLanguagePointcut	<bean b xmlns=
453	cockpitValidationService	<bean b beanRe <proper

No.	Bean ID	Previou
454	validationUIHelper	<bean b resourc
455	cockpitCommentService	<bean b beanRe
456	xmlDataProvider	<bean b beanRe <proper
457	customXmlDataProvider	<bean b beanRe
458	localizedXmlDataProvider	<bean b beanRe <proper
459	reportsService	<bean b path res <proper
460	reportsDAO	<bean b [cockpi name='
461	defaultCommentLayerService	<bean b beanRe xmlns= <proper <proper name='
462	defaultCommentsCommandsRegistry	<bean b xmlns=
463	defaultCreateNewCommentExecutor	<bean b beanId= xmlns=
464	defaultSelectCommentExecutor	<bean b beanRe
465	defaultEditCommentExecutor	<bean b beanRe
466	abstractUserCommentsPermissionsCheckStrategy	<bean b beanId= xmlns= name='
467	defaultUserCanEditCommentStrategy	<bean b beanId= parentN

No.	Bean ID	Previou
468	defaultUserCanReplyCommentStrategy	<bean b beanId= parentN
469	defaultUserCanDeleteCommentStrategy	<bean b beanId= parentN
470	defaultUserCanMoveCommentStrategy	<bean b beanId= parentN
471	defaultUserCanCreateCommentStrategy	<bean b beanId= parentN
472	defaultCockpitCelumDelegate	<bean b beanRe
473	workflowFacade	<bean b beanRe <proper name=' name='
474	defaultTableValueRequestCache	<bean b resourc
475	defaultUiAccessRightServiceWriteRequestCache	<bean b beanRe
476	defaultUiAccessRightServiceReadRequestCache	<bean b beanRe
477	defaultValueHandlerChainRequestCache	<bean b path res
478	defaultMatchingDescriptorsRequestCache	<bean b beanRe
479	defaultCoverageInfoRequestCache	<bean b resourc
480	classAttributeRequestCache	<bean b resourc
481	AbstractSimplifiedModelLabelProvider	<bean b beanId= parentN
482	entitlementLabelProvider	<bean b beanRe xmlns=

No.	Bean ID	Previou
483	b2bWorkflowMigrationFor4_4Release:beanResourceDescription	<bean b beanRe comme xmlns= name='
484	b2bUnitServiceProxy	<bean b resourc </bean
485	defaultB2BApprovalProcessService	<bean b beanId= xmlns=
486	defaultUserFacade	<bean b path res name=' name=' <proper <proper
487	:name	<proper
488	:name	<proper
489	:name	<proper
490	:name	<proper
491	:name	<proper
492	:name	<proper
493	:name	<proper
494	:name	<proper
495	:name	<proper
496	:name	<proper
497	:name	<proper
498	:name	<proper

No.	Bean ID	Previou
499	defaultCustomerFacade	<bean b beanRe name=" <proper name=" name=" name=" name=" <proper
500	defaultStoreFinderFacade	<bean b beanRe name=" name="
501	:name	<proper
502	:name	<proper
503	:name	<proper
504	defaultCustomerGroupFacade	<bean b beanRe name=" name="
505	:name	<proper
506	:name	<proper
507	:name	<proper
508	:name	<proper
509	defaultComponentTypeStructureRegistry:beanResourceDescription	<bean b beanId= spring.x name="
510	isProductAssignableFromPredicate:beanResourceDescription	<bean b spring.x name="
511	isCategoryAssignableFromPredicate:beanResourceDescription	<bean b spring.x name="
512	basicComponentTypePopulator:beanResourceDescription	<bean b beanRe xmlns=
513	i18nComponentTypePopulator:beanResourceDescription	<bean b beanRe xmlns=

No.	Bean ID	Previou
514	previewCategoryComponentTypePopulator:beanResourceDescription	<bean b beanId= spring.x
515	pageCategoryComponentTypePopulator:beanResourceDescription	<bean b beanRe xmlns=
516	restrictionCategoryComponentTypePopulator:beanResourceDescription	<bean b beanId= spring.x
517	basicComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
518	i18nComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
519	cmsStructureEnumTypeComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= registry <proper
520	cmsStructureEnumTypeComponentTypeAttributePopulator	<bean b beanId= registry <proper
521	:name	<proper
522	:name	<proper
523	nonEditableComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
524	mediaFormatsComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= service- name="
525	mediaContainerContainedTypesComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= structu
526	requiredComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
527	localizedComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
528	numberComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x

No.	Bean ID	Previou
529	floatComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
530	shortStringComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
531	longStringComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
532	richTextComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
533	booleanComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
534	mediaComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
535	mediaContainerComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= service
536	productTypeAttributePopulator:beanResourceDescription	<bean b beanId= scope=
537	categoryTypeAttributePopulator:beanResourceDescription	<bean b beanId= scope=
538	restrictionTypeAttributePopulator:beanResourceDescription	<bean b beanId= scope=
539	itemLinkToggleTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
540	navigationNodeSelectorComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= registry
541	navigationEntryItemPopulator:beanResourceDescription	<bean b beanRe parentN name='
542	dropdownComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
543	cmsItemDropdownComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= service name='

No.	Bean ID	Previou
544	navigationEntryDropdownComponentTypeAttributePopulator:beanResourceDescription	<bean b structu xmlns=
545	pageDropdownComponentTypeAttributePopulator:beanResourceDescription	<bean b service <prope
546	componentRestrictionComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= registry
547	pageRestrictionComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= service
548	dateComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
549	multiProductSelectorPopulator:beanResourceDescription	<bean b beanId= scope=
550	multiCategorySelectorPopulator:beanResourceDescription	<bean b beanId= scope=
551	displayConditionEditorPopulator:beanResourceDescription	<bean b beanId= scope=
552	userGroupIdComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
553	userGroupLabelComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= service
554	cmsLinkToSelectTypeAttributePopulator:beanResourceDescription	<bean b beanId= spring.x
555	cmsLinkToTypeAttributePopulator:beanResourceDescription	<bean b beanRe xmlns=
556	cmsLinkComponentLinkToIdTypeAttributePopulator:beanResourceDescription	<bean b beanId= service
557	previewCatalogI18nComponentTypeAttributePopulator:beanResourceDescription	<bean b beanId= service
558	cmsLinkComponentLinkToLabelTypeAttributePopulator:beanResourceDescription	<bean b beanId= service

No.	Bean ID	Previou
559	cmsLinkProductAndCategoryIdTypeAttributePopulator:beanResourceDescription	<bean b beanId= service-
560	cmsLinkProductAndCategoryLabelTypeAttributePopulator:beanResourceDescription	<bean b beanId= registry
561	cmsLinkContentPageLabelTypeAttributePopulator:beanResourceDescription	<bean b beanId= service-
562	productCatalogVersionsSelectorPopulator:beanResourceDescription	<bean b beanId= spring.>
563	defaultPreviewDataForVersionModeEditableAttributesPopulator:beanResourceDescription	<bean b beanId= registry
564	defaultComponentTypeFacade	<bean b beanRe <proper name='
565	defaultCmsComponentTypeStructureModelConverter	<bean b beanId= spring.> name=' name='
566	:name	<proper
567	defaultCmsPageFacade	<bean b path res <proper <proper name=' name='
568	mapMergeDirective$child#9:beanResourceDescription	<bean b parentN </bean
569	defaultAbstractCMSComponentContainerModelVisitor	<bean b beanId= spring.>
570	defaultAbstractCMSComponentContainerModelVisitor:scope	<bean b beanId= spring.>
571	defaultAbstractCMSComponentContainerModelVisitor	<bean b beanId= spring.>
572	defaultAbstractCMSComponentModelVisitor	<bean b beanId= spring.>

No.	Bean ID	Previou
573	defaultItemSynchronizationFacade	<bean b beanRe xmlns= <proper name=" name="
574	defaultCmsEnumContentConverter	<bean b beanId= xmlns= </bean
575	:name	<proper
576	defaultCmsPrimaryPageWithLabelExistsPredicate	<bean b beanId= spring.x name="
577	defaultCmsUpdateAbstractPageValidator	<bean b beanId= parentN <proper name=" </bean
578	defaultCmsAbstractCMSComponentValidator	<bean b beanId= parentN
579	defaultCmsUpdateAbstractRestrictionValidator	<bean b beanId= parentN
580	:name	<proper
581	:name	<proper
582	:name	<proper
583	:name	<proper
584	:name	<proper
585	:name	<proper
586	:name	<proper
587	:name	<proper

No.	Bean ID	Previou
588	:name	<proper
589	:name	<proper
590	:name	<proper
591	:name	<proper
592	:name	<proper
593	:name	<proper
594	:name	<proper
595	:name	<proper
596	defaultPageModelToDataRenderingPopulator	<bean b beanId= xmlns= name='
597	defaultSlotModelToDataRenderingPopulator	<bean b beanId= xmlns= name='
598	defaultCMSNavigationNodeModelToDataRenderingPopulator	<bean b beanId= spring.> name='
599	defaultPageRenderingService	<bean b beanRe <proper <proper
600	renderingCategoryPageModelSupplier	<bean b beanId= xmlns= name='
601	renderingContentPageModelSupplier	<bean b beanId= xmlns= name='
602	:name	<proper

No.	Bean ID	Previou
603	renderingProductPageModelSupplier	<bean b beanId= xmlns= name='
604	defaultSynchronizationFacade	<bean b beanRe name=' name=' name=' <proper
605	mapMergeDirective$child#13:beanResourceDescription	<bean b scope=
606	de.hybris.platform.cmsfacades.types.service.impl.DefaultComponentTypeAttributeStructure#29:beanResourceDescription	<bean b beanId= [sapym name='
607	de.hybris.platform.cmsfacades.types.service.impl.DefaultComponentTypeAttributeStructure#31:beanResourceDescription	<bean b beanId= [merch <proper
608	modifyPopulatorList$child#2:beanResourceDescription	<bean b parentN </bean
609	sapProductConfigDefaultBaseFacade	<bean b beanRe <proper name=' <proper name='
610	sapProductConfigDefaultMessageMapper	<bean b beanId= xmlns=
611	modifyPopulatorList$child#3:beanResourceDescription	<bean b parentN </bean
612	modifyPopulatorList$child#4:beanResourceDescription	<bean b parentN </bean
613	modifyPopulatorList$child#5:beanResourceDescription	<bean b parentN </bean
614	modifyPopulatorList$child#6:beanResourceDescription	<bean b parentN </bean

No.	Bean ID	Previou
615	sapProductConfigDefaultConfigOverviewFacade	<bean b beanId= xmlns= name='
616	modifyPopulatorList$child#7:beanResourceDescription	<bean b parentN </bean
617	modifyPopulatorList$child#8:beanResourceDescription	<bean b parentN </bean
618	sapCpiOrderDefaultService	<bean b beanRe <proper
619	sapCpiSendOmmOrderToScpiAction	<bean b beanId= parentN name='
620	sapCpiSendOmmOrderCancellationToScpiAction	<bean b beanId= parentN name='
621	defaultSapCpiCustomerService	<bean b beanId= xmlns=
622	modifyPopulatorList$child#9:beanResourceDescription	<bean b parentN </bean
623	modifyPopulatorList$child#10:beanResourceDescription	<bean b parentN </bean
624	modifyPopulatorList$child#11:beanResourceDescription	<bean b parentN </bean
625	modifyPopulatorList$child#12:beanResourceDescription	<bean b parentN </bean
626	modifyPopulatorList$child#13:beanResourceDescription	<bean b parentN </bean
627	modifyPopulatorList$child#14:beanResourceDescription	<bean b parentN </bean
628	modifyPopulatorList$child#15:beanResourceDescription	<bean b parentN </bean

No.	Bean ID	Previou
629	modifyPopulatorList$child#16:beanResourceDescription	<bean b parentN </bean
630	modifyPopulatorList$child#17:beanResourceDescription	<bean b parentN </bean
631	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#0:beanResourceDescription	<bean b beanId= [configu <proper
632	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#1:beanResourceDescription	<bean b beanId= [configu <proper
633	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#2:beanResourceDescription	<bean b beanId= [configu <proper
634	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#3:beanResourceDescription	<bean b beanId= [configu <proper
635	defaultBundleOrderPopulator:parentName	<bean b beanId= parentN
636	configurablePopulatorModification$child#0:beanResourceDescription	<bean b parentN name="
637	modifyPopulatorList$child#20:beanResourceDescription	<bean b parentN </bean
638	:name	<proper
639	defaultBundleCommerceOrderEntryPopulator	<bean b beanId= parentN <proper
640	modifyPopulatorList$child#21:beanResourceDescription	<bean b parentN </bean
641	:name	<proper
642	modifyPopulatorList$child#22:beanResourceDescription	<bean b parentN </bean

No.	Bean ID	Previou
643	defaultBundleCommerceCartPopulator:parentName	<bean b beanId= parentN
644	defaultBundleCommerceCartPopulator	<bean b beanId= parentN
645	modifyPopulatorList$child#23:beanResourceDescription	<bean b parentN </bean
646	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#4:beanResourceDescription	<bean b beanId= [entitle name='
647	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#5:beanResourceDescription	<bean b beanId= [entitle name='
648	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#6:beanResourceDescription	<bean b beanId= [entitle name='
649	modifyPopulatorList$child#24:beanResourceDescription	<bean b parentN </bean
650	configurablePopulatorModification$child#1:beanResourceDescription	<bean b parentN name='
651	modifyPopulatorList$child#25:beanResourceDescription	<bean b parentN </bean
652	modifyPopulatorList$child#26:beanResourceDescription	<bean b parentN </bean
653	org.springframework.integration.config.ServiceActivatorFactoryBean#5	<bean b beanId= name='
654	org.springframework.integration.config.ServiceActivatorFactoryBean#5	<proper
655	org.springframework.integration.config.ServiceActivatorFactoryBean#5	<proper
656	org.springframework.integration.config.ServiceActivatorFactoryBean#6	<bean b beanId= name='

No.	Bean ID	Previou
657	org.springframework.integration.config.ServiceActivatorFactoryBean#6	<proper
658	org.springframework.integration.config.ServiceActivatorFactoryBean#7	<bean b beanId= name='
659	org.springframework.integration.config.ServiceActivatorFactoryBean#9	<bean b beanId= name='
660	org.springframework.integration.config.ServiceActivatorFactoryBean#11	<bean b beanId= name='
661	org.springframework.integration.config.ServiceActivatorFactoryBean#11	<proper
662	org.springframework.integration.config.ServiceActivatorFactoryBean#13	<bean b beanId= name='
663	org.springframework.aop.aspectj.AspectJPointcutAdvisor#5	<bean b scope=
664	org.springframework.aop.aspectj.AspectJPointcutAdvisor#6	<bean b scope=
665	org.springframework.aop.aspectj.AspectJPointcutAdvisor#7	<bean b scope=
666	defaultAbstractHybrisVelocityContextFactory	<bean b beanId= xmlns= name=' <proper
667	:name	<proper
668	:name	<proper
669	:name	<proper
670	:name	<proper
671	:name	<proper
672	:name	<proper
673	:name	<proper

No.	Bean ID	Previou
674	:name	<prope
675	mapMergeDirective$child#14:beanResourceDescription	<bean b parentN </bean
676	mapMergeDirective$child#15:beanResourceDescription	<bean b scope=
677	cxCmsComponentContainerStrategy	<bean b beanRe name='
678	mapMergeDirective$child#16:beanResourceDescription	<bean b parentN </bean
679	mapMergeDirective$child#17:beanResourceDescription	<bean b parentN </bean
680	listMergeDirective$child#14:beanResourceDescription	<bean b parentN
681	listMergeDirective$child#15:beanResourceDescription	<bean b parentN
682	mapMergeDirective$child#18:beanResourceDescription	<bean b parentN </bean
683	mapMergeDirective$child#19:beanResourceDescription	<bean b parentN </bean
684	defaultAbstractMediaContainerComponentModelVisitor	<bean beanCla beanId= spring.x
685	defaultAbstractMediaContainerComponentModelVisitor:scope	<bean beanCla beanId= spring.x
686	defaultAbstractMediaContainerComponentModelVisitor	<bean beanCla beanId= spring.x
687	defaultNavigationBarCollectionComponentModelVisitor	<bean b beanId= spring.x
688	defaultNavigationBarCollectionComponentModelVisitor:scope	<bean b beanId= spring.x

No.	Bean ID	Previou
689	defaultNavigationBarCollectionComponentModelVisitor	<bean b beanId= spring.x
690	defaultNavigationBarComponentModelVisitor	<bean b beanId= scope=
691	defaultNavigationBarComponentModelVisitor:scope	<bean b beanId= scope=
692	defaultNavigationBarComponentModelVisitor	<bean b beanId= scope=
693	defaultNavigationComponentModelVisitor	<bean b beanId= xmlns=
694	defaultNavigationComponentModelVisitor:scope	<bean b beanId= xmlns=
695	defaultNavigationComponentModelVisitor	<bean b beanId= xmlns=
696	mapMergeDirective$child#21:beanResourceDescription	<bean b parentN </bean
697	modifyPopulatorList$child#27:beanResourceDescription	<bean b parentN </bean
698	configurablePopulatorModification$child#2:beanResourceDescription	<bean b parentN name='
699	modifyPopulatorList$child#28:beanResourceDescription	<bean b parentN </bean
700	modifyPopulatorList$child#29:beanResourceDescription	<bean b parentN </bean
701	modifyPopulatorList$child#30:beanResourceDescription	<bean b parentN </bean
702	mapMergeDirective$child#25:beanResourceDescription	<bean b parentN </bean
703	cartTaxInvoicePopulator	<bean b beanRe comme xmlns= name='

No.	Bean ID	Previou
704	orderTaxInvoicePopulator	<bean b beanRe comme xmlns= name='
705	taxInvoiceReversePopulator	<bean b beanRe comme xmlns= name='
706	chineseTaxInvoiceCheckoutFacade	<bean b beanId= parentN <proper
707	defaultTaxInvoiceReverseConverter	<bean b parentN name='
708	modifyPopulatorList$child#31:beanResourceDescription	<bean b parentN </bean
709	modifyPopulatorList$child#32:beanResourceDescription	<bean b parentN </bean
710	modifyPopulatorList$child#33:beanResourceDescription	<bean b parentN </bean
711	modifyPopulatorList$child#34:beanResourceDescription	<bean b parentN </bean
712	modifyPopulatorList$child#35:beanResourceDescription	<bean b parentN </bean
713	modifyPopulatorList$child#36:beanResourceDescription	<bean b parentN </bean
714	acceleratorImageFormatMapping:beanResourceDescription	<bean b parentN
715	acceleratorProductPrimaryImagePopulator:beanResourceDescription	<bean b parentN </bean
716	acceleratorProductGalleryImagesPopulator:beanResourceDescription	<bean b parentN </bean
717	defaultCommerceWebServicesPaymentFacade	<bean b beanId= parentN name='

No.	Bean ID	Previou
718	wsSiteBaseUrlResolutionService	<bean b beanId= parentN
719	defaultPaymentSubscriptionResultDao	<bean b beanId= parentN
720	defaultPaymentSubscriptionResultService	<bean b beanId= parentN </bean
721	webServicesPlaceOrderHook	<bean b beanRe <proper
722	wsPlaceOrderMethodHooksMergeDirective	<bean b parentN
723	oldPaymentSubscriptionResultRemovalJob	<bean b beanId= parentN <proper
724	defaultMappingLabelsPopulator	<bean b beanRe <proper
725	paymentDataConverterListMergeDirective	<bean b parentN name='
726	defaultGigyaSwitchUIComponentAspect	<bean b beanRe
727	defaultProductPriceAndStockConverter:beanResourceDescription	<bean b parentN name='
728	modifyPopulatorList$child#37:beanResourceDescription	<bean b parentN </bean
729	batchFilesElectronics.adapter.source	<bean b scope= name='
730	:name	<proper
731	batchFilesApparel.adapter.source	<bean b scope= name='
732	:name	<proper

No.	Bean ID	Previou
733	org.springframework.integration.config.ServiceActivatorFactoryBean#15	<bean b beanId= name='
734	org.springframework.integration.config.ServiceActivatorFactoryBean#15	<proper
735	mccSiteUrlHelper	<bean b [yaccel <proper
736	modifyPopulatorList$child#38:beanResourceDescription	<bean b parentN </bean
737	de.hybris.platform.cmsfacades.types.service.impl.DefaultComponentTypeAttributeStructure#32:beanResourceDescription	<bean b beanId= [config/ name='
738	cisAddress:beanClassName	<bean b spring.x
739	cisAddress:beanResourceDescription	<bean b spring.x
740	cisLineItem:beanClassName	<bean b spring.x
741	cisLineItem:beanResourceDescription	<bean b spring.x
742	defaultCisLineItemPopulator:beanClassName	<bean b beanRe
743	defaultCisLineItemPopulator:beanResourceDescription	<bean b beanRe
744	defaultCisLineItemPopulator	<bean b beanRe
745	defaultCisAddressConverter:beanResourceDescription	<bean b parentN
746	defaultCisLineItemConverter:beanResourceDescription	<bean b parentN
747	modifyPopulatorList$child#39:beanResourceDescription	<bean b parentN </bean
748	xyformscockpitsLabelLocator	<bean b beanRe <proper

No.	Bean ID	Previou
749	modifyPopulatorList$child#40:beanResourceDescription	<bean b parentN </bean
750	marketplaceCartValidationHook:beanClassName	<bean b beanRe name='
751	customerVendorReviewsCleanupHook:beanClassName	<bean b beanId= xmlns=
752	batchFilesMarketplace.adapter.source	<bean b scope= name='
753	mapMergeDirective$child#26:beanResourceDescription	<bean b scope=
754	sapCpiSendOmsOrderToScpiAction	<bean b beanId= parentN <proper
755	sapCpiSendOmsOrderCancellationToScpiAction	<bean b beanId= parentN
756	sapcpiorderexchangeomsb2bSystemSetup	<bean b beanId= xmlns=
757	sapcpiorderexchangeomsb2bService	<bean b beanId= xmlns= name='
758	chineseAddressDao	<bean b beanRe xmlns=
759	chineseAddressService	<bean b beanRe xmlns=
760	chinesePostcodeValidateStrategy	<bean b beanRe <proper
761	nameWithTitleFormatStrategy	<bean b beanRe <proper
762	defaultCityConverter	<bean b parentN name='
763	defaultDistrictConverter	<bean b parentN name='

No.	Bean ID	Previou
764	cityPopulator	<bean b [chines
765	districtPopulator	<bean b resourc
766	chineseAddressFacade	<bean b beanRe <proper name='
767	chineseAddressPopulator	<bean b beanRe <proper
768	modifyPopulatorList$child#41:beanResourceDescription	<bean b parentN </bean
769	chineseAddressReversePopulator	<bean b beanRe xmlns=
770	chineseUserFacade	<bean b path res
771	defaultChineseAbstractEmailContext	<bean b beanId= parentN name='
772	defaultChineseUserDao	<bean b beanRe
773	chineseUserDetailsService	<bean b beanRe name='
774	chineseCustomerAccountService	<bean b beanId= parentN name='
775	chineseVerificationCodeGenerationStrategy	<bean b beanId= xmlns=
776	chineseVerificationCodeSendingStrategy	<bean b beanId= xmlns=
777	defaultChineseSmsService	<bean b beanRe
778	chineseProfileSMSChannelStrategy	<bean b beanId= parentN

No.	Bean ID	Previou
779	chineseB2BUnitFacade	<bean b beanId= parentN
780	chineseC2LItemZoneDeliveryModeDao	<bean b beanId= parentN
781	chineseDeliveryTimeSlotDao	<bean b beanId= xmlns=
782	chineseC2LItemZoneDeliveryModeValueDao	<bean b beanId= xmlns=
783	chineseDeliveryModeLookupStrategy	<bean b beanId= parentN name='
784	chineseDeliveryTimeSlotService	<bean b beanId= xmlns=
785	chineseDeliveryService	<bean b beanRe xmlns=
786	chineseFindDeliveryCostStrategy	<bean b beanId= parentN name='
787	chineseSLFindDeliveryCostStrategy	<bean b beanId= parentN name='
788	defaultDeliveryTimeSlotConverter	<bean b parentN name='
789	chineseDeliveryTimeSlotFacade	<bean b beanId= parentN <proper
790	deliveryTimeSlotPopulator	<bean b beanRe
791	cartDeliveryTimeSlotPopulator	<bean b beanRe
792	modifyPopulatorList$child#42:beanResourceDescription	<bean b parentN </bean

No.	Bean ID	Previou
793	modifyPopulatorList$child#43:beanResourceDescription	<bean b parentN </bean
794	orderDeliveryTimeSlotPopulator	<bean b beanId= parentN
795	modifyPopulatorList$child#44:beanResourceDescription	<bean b parentN </bean
796	productInterestDao	<bean b beanRe <proper
797	productInterestPopulator	<bean b beanId= xmlns= <proper
798	abstractStockLevelStatusJob	<bean b beanRe xmlns= name=" name="
799	defaultStockNotificationSmsProcessor	<bean b beanId= xmlns=
800	mapMergeDirective$child#27:beanResourceDescription	<bean b parentN </bean
801	:name	<proper
802	:name	<proper
803	mapMergeDirective$child#28:beanResourceDescription	<bean b parentN </bean
804	modifyPopulatorList$child#45:beanResourceDescription	<bean b parentN </bean
805	modifyPopulatorList$child#46:beanResourceDescription	<bean b parentN </bean
806	modifyPopulatorList$child#47:beanResourceDescription	<bean b parentN </bean

No.	Bean ID	Previou
807	currentFactoryFindBundlePricingStrategy	<bean b beanId= parentN name='
808	defaultBundleCommercePlaceOrderMethodHook	<bean b beanId= xmlns=
809	:name	<proper
810	defaultBundleCommercePlaceOrderMethodHook	<proper
811	defaultBundleCommerceCartService	<bean b beanId= parentN name=' name=' name='
812	autoPickBundleComponentEditableChecker	<bean b beanId= parentN name='
813	autoPickCartBundleComponentEditableChecker	<bean b beanId= parentN name='
814	:name	<proper
815	autoPickCartBundleComponentEditableChecker	<proper
816	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#14:beanResourceDescription	<bean b beanId= [subscr name='
817	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#15:beanResourceDescription	<bean b beanId= [subscr name='
818	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#16:beanResourceDescription	<bean b beanId= [subscr name='
819	de.hybris.platform.commercefacades.xstream.alias.TypeAliasMapping#17:beanResourceDescription	<bean b beanId= [subscr name='

No.	Bean ID	Previou
820	modifyPopulatorList$child#48:beanResourceDescription	<bean b parentN </bean
821	modifyPopulatorList$child#49:beanResourceDescription	<bean b parentN </bean
822	modifyPopulatorList$child#50:beanResourceDescription	<bean b parentN </bean
823	modifyPopulatorList$child#51:beanResourceDescription	<bean b parentN </bean
824	modifyPopulatorList$child#52:beanResourceDescription	<bean b parentN </bean
825	modifyPopulatorList$child#53:beanResourceDescription	<bean b parentN </bean
826	modifyPopulatorList$child#54:beanResourceDescription	<bean b parentN </bean
827	modifyPopulatorList$child#55:beanResourceDescription	<bean b parentN </bean
828	modifyPopulatorList$child#56:beanResourceDescription	<bean b parentN </bean
829	defaultSubscriptionCartFacade	<bean b beanRe xmlns= <prope
830	defaultOauth2Client:beanResourceDescription	<bean b beanRe 18.08.0 xmlns= name='
831	defaultSubscriptionClient:beanResourceDescription	<bean b beanRe 18.08.0 xmlns= name='
832	subscriptionConfiguration:beanClassName	<bean b beanId= addon/ RC2.jar xmlns= name=' name='

No.	Bean ID	Previou
833	subscriptionConfiguration:beanResourceDescription	<bean b beanId= addon/ RC2.jar xmlns= name=" name="
834	modifyPopulatorList$child#57:beanResourceDescription	<bean b parentN </bean
835	modifyPopulatorList$child#58:beanResourceDescription	<bean b parentN </bean
836	modifyPopulatorList$child#59:beanResourceDescription	<bean b parentN </bean
837	defaultSapSubscriptionConfigurationService	<bean b beanId= xmlns= name="
838	:name	<proper
839	defaultSapSubscriptionConfigurationService	<proper
840	defaultSapSubscriptionConfigurationService	<proper
841	modifyPopulatorList$child#60:beanResourceDescription	<bean b parentN </bean
842	modifyPopulatorList$child#61:beanResourceDescription	<bean b parentN </bean
843	defaultBundleCartFacade	<bean b beanRe xmlns= <proper
844	modifyPopulatorList$child#62:beanResourceDescription	<bean b parentN </bean
845	:name	<proper
846	modifyPopulatorList$child#63:beanResourceDescription	<bean b parentN </bean

No.	Bean ID	Previou
847	defaultBundleOrderEntryPopulator	<bean b beanId= parentN name='
848	modifyPopulatorList$child#64:beanResourceDescription	<bean b parentN </bean
849	:name	<proper
850	modifyPopulatorList$child#65:beanResourceDescription	<bean b parentN </bean
851	:name	<proper
852	modifyPopulatorList$child#66	<bean b parentN </bean
853	modifyPopulatorList$child#67	<bean b parentN </bean
854	defaultBundleCartPopulator	<bean b beanRe xmlns= name='
855	modifyPopulatorList$child#68	<bean b parentN </bean
856	defaultBundleMiniCartPopulator	<bean b beanId= parentN </bean
857	defaultProductBundlePopulator	<bean b beanId= parentN <proper
858	defaultBundleCartPotentialProductsPopulator	<bean b beanId= parentN name=' name='
859	defaultBundleCartPotentialProductDisableRulePopulator	<bean b beanId= parentN name='
860	chineseStockLevelReservationHistoryEntryDao	<bean b beanId= xmlns=

No.	Bean ID	Previou
861	chineseStockLevelReservationHistoryEntryService	<bean b beanId= xmlns=
862	chinesePaymentServicesStrategy	<bean b beanId= xmlns=
863	chineseCheckoutService	<bean b beanRe <proper <proper name='
864	chineseCommercePlaceOrderMethodHook	<bean b beanId= xmlns=
865	chinesepaymentCommercePlaceOrderMethodHooksListMergeDirective	<bean b [chines name='
866	chineseOrderDao	<bean b beanRe xmlns=
867	chineseOrderService	<bean b beanRe xmlns= name='
868	chineseCheckAuthorizeOrderPaymentAction	<bean b beanId= xmlns=
869	chineseTakePaymentAction	<bean b beanRe
870	chineseFraudCheckOrderInternalAction	<bean b beanId= xmlns=
871	weChatPayPaymentInfoStrategy	<bean b beanId= xmlns=
872	wechatpayPaymentService	<bean b beanId= parentN <proper name=' name='
873	weChatPayPaymentTransactionStrategy	<bean b beanId= xmlns= <proper

No.	Bean ID	Previou
874	weChatPayNotificationService	<bean b beanId= xmlns= name='
875	weChatPayPaymentTransactionDao	<bean b beanId= xmlns=
876	weChatPayPaymentTransactionEntryDao	<bean b beanId= xmlns=
877	weChatPayOrderService	<bean b beanRe xmlns=
878	weChatPayOrderDao	<bean b beanRe
879	weChatPayConfiguration	<bean b beanRe <proper name=' name='
880	weChatPayHttpClient	<bean b beanRe
881	weChatPayPayTransKeyGenerator	<bean b beanId= xmlns= name='
882	defaultAlipayConfiguration	<bean b beanRe <proper name=' name=' name=' name=' name='
883	alipayPaymentService	<bean b beanRe xmlns= <proper <proper <proper name=' name='
884	alipayPaymentInfoStrategy	<bean b beanRe <proper
885	alipayHandleResponseStrategy	<bean b beanId= xmlns=

No.	Bean ID	Previou
886	alipayPaymentTransactionStrategy	<bean b beanId= xmlns= <proper
887	alipayOrderService	<bean b beanRe xmlns=
888	alipayOrderDao	<bean b beanRe
889	alipayPayTransKeyGenerator	<bean b beanRe <proper
890	alipayPaymentTransactionEntryDao	<bean b beanId= xmlns=
891	alipayPaymentTransactionDao	<bean b beanRe
892	defaultAlipayCreateRequestStrategy	<bean b beanId= xmlns=
893	defaultAlipayResponseValidationStrategy	<bean b beanId= xmlns=
894	alipaySubmitOrderStrategy	<bean b beanRe <proper
895	defaultAlipayService	<bean b beanRe <proper
896	orderPayImmediatelyPopulator	<bean b beanRe comme scope= name='
897	chineseCheckoutFacade	<bean b beanRe xmlns= name=' <proper
898	defaultCartChinesePaymentInfoConverter	<bean b parentN name='
899	cartChinesePaymentInfoPopulator	<bean b beanId= xmlns= </bean

No.	Bean ID	Previou
900	modifyPopulatorList$child#69	<bean b parentN </bean
901	chineseOrderPopulator	<bean b beanRe xmlns= <proper
902	modifyPopulatorList$child#70	<bean b parentN </bean
903	modifyPopulatorList$child#71	<bean b parentN </bean
904	submitOrderEventTask	<bean b beanRe <proper name='
905	chineseCustomerFacade	<bean b beanRe xmlns= <proper
906	chineseCustomerPopulator	<bean b beanRe
907	modifyPopulatorList$child#72	<bean b parentN </bean
908	chineseEmailContextFactory	<bean b beanId= parentN
909	extOrderNotificationEmailContext	<bean b beanId= parentN <proper
910	extCustomerEmailContext	<bean b beanRe xmlns=
911	extForgottenPasswordEmailContext	<bean b beanId= parentN </bean
912	extDeliverySentEmailContext	<bean b beanRe xmlns=
913	extReadyForPickupEmailContext	<bean b beanId= parentN name='

No.	Bean ID	Previou
914	extOrderCancelledEmailContext	<bean b beanId= parentN </bean
915	extOrderRefundEmailContext	<bean b beanRe xmlns=
916	extConsignmentCollectionReminderEmailContext	<bean b beanId= parentN name=' name='
917	extOrderPartiallyModifiedEmailContext	<bean b beanId= parentN <proper
918	extOrderPartiallyCanceledEmailContext	<bean b beanId= parentN
919	extOrderPartiallyRefundedEmailContext	<bean b beanId= parentN
920	extNotPickedUpConsignmentCanceledEmailContext	<bean b beanId= parentN name='
921	mockAlipayCreateRequestStrategy	<bean b beanRe xmlns=
922	mockAlipayResponseValidationStrategy	<bean b beanId= parentN
923	mockService	<bean b resourc <proper <proper name='
924	modifyPopulatorList$child#73	<bean b parentN </bean
925	modifyPopulatorList$child#74	<bean b parentN </bean
926	modifyPopulatorList$child#75	<bean b parentN </bean
927	modifyPopulatorList$child#76	<bean b parentN </bean

No.	Bean ID	Previou
928	modifyPopulatorList$child#77	<bean b parentN </bean
929	modifyPopulatorList$child#78	<bean b parentN </bean
930	customerCouponCodeValueProvider	<bean b beanId= parentN <proper name='
931	customerCouponCodeValueResolver	<bean b beanId= parentN name='
932	customerCouponFacetDisplayNameProvider	<bean b beanId= scope= name='
933	abstractCouponNotificationJob	<bean b beanRe xmlns= name=' name='
934	generateCouponNotificationEmail	<bean b parentN </bean
935	mapMergeDirective$child#29:beanResourceDescription	<bean b parentN </bean
936	CouponEffectiveNotificationEmailRegistrar	<bean b beanRe <proper
937	CouponExpireNotificationEmailRegistrar	<bean b beanRe <proper
938	CouponEffectiveNotificationSmsRegistrar	<bean b beanRe <proper
939	CouponExpireNotificationSmsRegistrar	<bean b beanRe <proper
940	couponNotificationSiteMessageProcessor	<bean b beanId= xmlns=
941	couponNotificationEmailProcessDefinitionResource	<bean b beanRe <proper

No.	Bean ID	Previou
942	mapMergeDirective$child#30:beanResourceDescription	<bean b parentN </bean
943	mapMergeDirective$child#31:beanResourceDescription	<bean b parentN </bean
944	customerCouponCommercePlaceOrderMethodHook	<bean b beanId= xmlns=
945	customerCouponCommercePlaceOrderMethodHooksListMergeDirective	<bean b [custon name='
946	couponNotificationCleanupHook	<bean b beanId= xmlns=
947	couponNotificationCleanupHookListMergeDirective	<bean b parentN
948	mapMergeDirective$child#32:beanResourceDescription	<bean b parentN </bean
949	listMergeDirective$child#17:beanResourceDescription	<bean b parentN
950	defeultFlashBuyFacade	<bean b beanRe xmlns= name='
951	defaultCustomerCouponFacade	<bean b beanRe <proper name=' name='
952	mapMergeDirective$child#33:beanResourceDescription	<bean b parentN </bean
953	mapMergeDirective$child#34:beanResourceDescription	<bean b parentN </bean