Class Setup

Exercise

SAP Customer Experience

TABLE OF CONTENTS

GOAL

In this exercise, you will set up the SAP Commerce Cloud platform, which will serve as the basis for the remaining exercises in this class.

Once you've built the site with the recipe provided, you will import its extensions into an IDE.

INSTRUCTIONS

Every exercise in this class comes with a verification script you can run to test your work. As these scripts can't possibly test everything, they're mostly useful for determining if you forgot something. That is, if they fail, you did something wrong, and if they pass, you *probably* completed the exercise correctly.

These verification scripts rely on you following the naming conventions in the exercises *to the letter*. That means you need to name your classes and components **exactly as given in the exercises**, including the same capitalization. Otherwise, the scripts can't determine if your work is correct. Naturally, you wouldn't have to follow these conventions in a real-life project.

If you are currently working on an SAP Commerce project on your computer, check your environment variables. If any of them start with `y_`, then see **note 1** in the Troubleshooting Guide.

While each module in this course is independent of all the other modules, we highly recommend that you perform the exercises in the order listed in the Overview slides so that there are not any accidental conflicts. If you do encounter a conflict, try running the Class Setup solution before completing your exercise.

Preparation - Install Required Software

If you're using an **SAP VM**, please skip this section and **start working on Step 1** (the required software has already been installed for you).

If you're working on your own computer, you need to make sure you have the following software installed:

JDK

The SAP Commerce Cloud platform currently runs on (64-bit) Java 17.

Check that you're running the Java 17 SDK

Verify that you're using the JRE from the Java 17 SDK when *running* and *compiling* Java from the command line. Open a *Terminal* or *cmd* window and enter the following commands:

- `java -version`
- `javac -version`

The version numbers of both commands need to match exactly.

E.g. You may get the following responses (assuming version 17.0.3):

`openjdk version "17.0.3"` or `java version "17.0.3"`

If either fails to give the correct result, you're not using the Java 17 SDK. If the `javac` command can't be found, you probably have only the run-time environment installed (JRE), and not the software development kit (SDK). In either case, you need to configure your command-line window to use the correct SDK…

P1 If you don't already have one on-hand, download the SAPMachine 17 SDK from the following location:

- https://sap.github.io/SapMachine/

P2 Install the unzipped file to a location appropriate for your Operating System. For example:

- Windows: `C:\Program Files\Java\sapmachine-jdk-17.0.3`
- MacOS: `/Library/Java/JavaVirtualMachines/sapmachine-jdk-17.0.3.jdk`
- Linux: `/opt/sapmachine-jdk-17.0.3`

P3 In Windows and Linux, set `JAVA_HOME` to point to the directory above. In MacOS, set it to `/Library/Java/JavaVirtualMachines/sapmachine-jdk-17.0.3.jdk/Contents/Home`.

P4 Prepend the location of the bin directory to your `PATH` (`$JAVA_HOME/bin` on MacOS/Linux, `%JAVA_HOME%\bin` on Windows)

Step 1 - Installing the SAP Commerce Cloud Accelerator

1A On SAP-Provided VMs (running Linux)

1A.1 If you're using either of the available VMs:

- SAP Hybris InstructorLed Training portal (via ReadyTech)
- SAP Classroom VM / Training Landscape (via https://class.learning.sap.com)

The course .zip file was extracted for you in the following location:

`~/c4h340_2211` (for C4H340 course) or `~/c4h341_2211` (for C4H341 course)

From this point forward, we will refer to this directory as `MYPATH`.
(If you like, you can set an environment variable to point to it.)

Within `MYPATH`, SAP Commerce Cloud Accelerator was placed into the directory `$MYPATH/workspace.`

The directory `$MYPATH/workspace/TrainingLabTools` contains automated setup scripts and starter files necessary to prepare exercises, which you will then complete.

It also contains an automated SAP Commerce installation recipe, which you will use next to configure SAP Commerce's settings, and specify which extensions to include when building your server.

1A.2 Copy this recipe folder, which the training team has created for you:
`$MYPATH/workspace/TrainingLabTools/resources/commerce_developer` into
`$MYPATH/workspace/installer/recipes/`.

1A.3 Once copied to the proper location, use this recipe to install the SAP Commerce Cloud as follows:

- Open a **Terminal** window (found under *Applications* > *Utilities*),
- change directory to `$MYPATH/workspace/installer,`
- in this directory, execute the following command:

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

1B On MacOS or Linux (running on your own system)

1B.1 For C4H340, unzip **sapCom-dev1-2211.zip**; For C4H341, unzip **sapCom-dev2-2211.zip,** the zip file content will be unzipped into a new directory under your home directory, which we will refer to as `MYPATH`. If you like, you can set an environment variable (`$MYPATH`) to point to it.

The directory `$MYPATH/workspace/TrainingLabTools` contains scripts and files which prepare your workspace for exercises, including a recipe defining which SAP Commerce extensions to include when building your server.

1B.2 Copy this recipe (`$MYPATH/workspace/TrainingLabTools/resources/commerce_developer`) to `$MYPATH/workspace/installer/recipes/commerce_developer`.

1B.3 Install the SAP Commerce Cloud using the copied recipe.

Open a **Terminal** window, navigate to `$MYPATH/workspace/installer` and execute the following command:

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

1C On Windows

1C.1 For C4H340, unzip **sapCom-dev1-2211.zip**; For C4H341, unzip **sapCom-dev2-2211.zip**; the zip file content will be unzipped into a new directory, which we will refer to as `MYPATH`. If you like, you can set an environment variable (`%MYPATH%`) to point to it.

Make sure you unzip the Commerce application in a directory close to the root of your filesystem and ensure there are no spaces in the folder name. For example, a good unzip location is `C:\training`. (Windows has a maximum path length of 256 characters and the SAP Commerce Cloud platform contain files whose total path lengths are close to that value so unzip to a directory as close to the drive root as possible)

The directory `%MYPATH%\workspace\TrainingLabTools` contains scripts and files necessary to complete your exercises, including a recipe defining which SAP Commerce extensions to include when building your server.

1C.2 Copy this recipe (`%MYPATH%\workspace\TrainingLabTools\resources\commerce_developer`) to `%MYPATH%\workspace\installer\recipes\commerce_developer`.

1C.3 Install the SAP Commerce Cloud using the copied recipe.

Open a **cmd** window (please avoid using **PowerShell**), navigate to `%MYPATH%\workspace\installer`, and execute the following command:

```
install -r commerce_developer initialize
```

The recipe contains all the information needed to install and set up our environment appropriately, such as necessary plugins, required extensions, and the properties we use in our configuration. For further information about recipes and guides on how to create your own recipes, please refer to Creating Installer Recipes.

This copied recipe is not part of the OOTB SAP Commerce Cloud application. It was created specifically for this training class.

Step 2 - Configure the hosts file

Note: If you are using an SAP Virtual Machine environment, the following step was performed for you. If you like, you can view the hosts file to verify the necessary entry.

Modifying the **`hosts`** file allows SAP Commerce Cloud to use the URL to determine which Commerce-defined "WCMS Site" should receive your HTTP request.

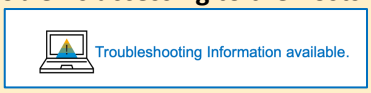
If you are using Linux or MacOS, the file is located here: `/etc/hosts`

If you are on Windows, the file is located here: `C:\Windows\System32\drivers\etc\hosts`

Add the following alias to the hosts file (As the last line of the file):

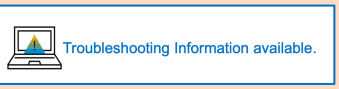
```
127.0.0.1 electronics.local
```

Problems accessing to the hosts file



See note **4**

No Administrator rights on your machine?



See note **5**

Step 3 - Verification and Archiving

Preparation: Set Up Ant

In future lab exercises, if you were to make changes to any Java class files or to **`*-items.xml`** files you will need to invoke the **ant** build tool. Open a *Terminal* or *cmd* window and keep it open for whenever you want to compile your code. In this window:

3.1 Any time you start with a NEW cmd or terminal window, you will need to once again set the **ant** and SAP Commerce Cloud platform environment variables in the current terminal window by navigating to the directory ***MYPATH*`/workspace/hybris/bin/platform`** and executing:
`. ./setantenv.sh` (for Linux or MacOS) or `setantenv.bat` (for Windows).

You will need to run this command every time you open a new terminal or command prompt window in which you want to run Ant commands for SAP Commerce Cloud.

3.2 Once your cmd or terminal's environment has been initialized for the session, you would then build your environment by invoking `ant all`. (This is for future reference—you do not need to do a build right now.)

> **The Linux / MacOS 'dot' Operator**
>
> On a Unix system like Linux or MacOS, the period operator is shorthand for the built-in 'source' command. It indicates the script should run in the current shell's context (as opposed to the script's own local sub-context), thereby allowing the `setantenv.sh` script to modify your shell's environment variables and set the Ant environment variables to match the SAP Commerce Cloud "installation" for which you ran the script.

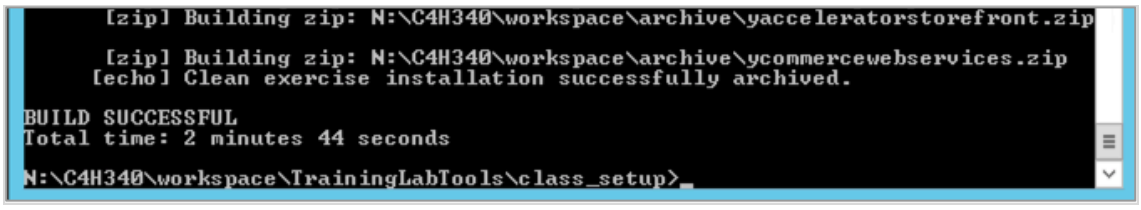
3C Load Verification Scripts

We have provided Groovy "exercise verification scripts" for most exercises. They need to be loaded into the Commerce Platform as "Script" items (thus, persisted in the database) because they must run from ***within*** the platform (via HAC). These will help you verify that you've completed your exercise correctly.

3C.1 In the same *Terminal* or *cmd* window you ran commands in during Step 2, navigate to `MYPATH/workspace/TrainingLabTools/class_setup` and execute the following command:

```
ant -f class_setup_tasks.xml verification
```

When this completes (it will take a few minutes), you should see a Build Successful message in your terminal or command prompt. If not, ask your instructor for help.



3D Verify Your Work

You just loaded verification scripts that can be invoked from HAC. However, the present exercise is one that doesn't have a verification script. You can manually verify that you've completed the exercise correctly by starting the server and logging into HAC:

3D.1 Start the SAP Commerce Cloud server by opening your *Terminal* or *cmd* window, navigating to `MYPATH/workspace/hybris/bin/platform`, and invoking `./hybrisserver.sh` (on Linux or MacOS) or `hybrisserver` (on Windows).

Be aware that immediately after an "ant initialize", starting up your server and accessing the Backoffice login screen will both take a significant amount of extra time. Likewise, the Backoffice home page will take much longer to first appear.

3D.2 Open a browser tab and navigate to `http://localhost:9001/backoffice` or `https://localhost:9002/backoffice`.

Once the login screen appears, enter **admin** as the username and **nimda** as the password to login. After a successful login to Backoffice, you may log out.

(While you wait for the login page to appear, you may perform the next step in parallel.)

3D.3 Open a browser tab and navigate to `http://localhost:9001/` or `https://localhost:9002/`. This is the URL for the SAP Hybris Administration Console (HAC). Enter **admin** as the username and **nimda** as the password to login.

If you can login to HAC, your system is installed. You can navigate to the **Extensions** tab (under the **Platform** tab) to see the list of extensions that were used to build your system (based on the list of extensions specified in the `commerce_developer` recipe).

Check that the verification scripts were properly loaded by going to *Console* > *Scripting Languages*. Click on the **Browse** tab. Either double-click on the **groovy** folder or click once on [+] to expand the node. If you performed all the steps successfully, you should see a list of all of the verification scripts for the exercises; if not, double-check your work or ask the instructor for help. If you *DO* see the verification scripts in the HAC, then you may log out of HAC.

If you performed step 3B.3 while waiting for Backoffice to come up in 3B.2, go back to your browser tab for Backoffice and complete step 3B.2. Check that your initial login screen has finally rendered. Login using **admin** as the username and **nimda** as the password to login. After a few minutes, you should be successfully logged into Backoffice and see the dashboard view. You may now log out of Backoffice.

3D.4 In the *Terminal* or *cmd* window where the server is running, and initiate a shutdown by entering `CTRL-C`. At this moment, please wait for the shutdown process to complete cleanly. Interrupting the shutdown process with additional `CTRL-C` commands can cause problems. If you accidentally do that, see the details in the box below:

Incomplete shutdown issues:

Among other important operations, the shutdown script issues a shutdown command to the SOLR standalone server. If you accidentally hit `CTRL-C` twice or if you close the Commerce server window before the script finishes, the SOLR server process may keep running in the background. If this is the case, you will probably not be able to restart the server until you forcibly kill the SOLR server process in the *Task Manager* (Windows) or in the *Activity Monitor* (Mac). If that happens, please talk to your instructor.

3E Archive Your Setup

You are now ready to create an "archive" (set of five ZIP files) of your initial setup, which our "exercise setup scripts" will use as a "clean slate" while preparing your workspace for each exercise…

Shut down the commerce server by going to the cmd or Terminal window and issuing a "control-C" (ctrl-C). Also shut down Eclipse to release any file locks it may have obtained.

3E.1 In the same *Terminal* or *cmd* window you ran your Commerce server (after the server shuts down), navigate back to `MYPATH/workspace/TrainingLabTools/class_setup` and execute the following command:

```
ant -f class_setup_tasks.xml archive
```

When this script returns a "build successful", then congratulations: **you have successfully completed this step**. If ***not*** successful, ask your instructor for help.

Step 4 - Just for Windows users

If you're using a Windows System

In the same *cmd* window you just used to run your archive ant task, execute `ant clean` before you proceed to the next step.

Step 5 - Development Environment Setup

Following are some steps that you'll need to take to prepare your development environment for doing the exercises. For this training:

- if you are using the SAP VM, we have included a pre-configured Eclipse IDE that is located under ~/c4h340_2211/eclipse-bundle (for C4H340 course) or ~/c4h341_2211/eclipse-bundle (for C4H341 course).
- If you are using your own computer, please make sure to download and install the Eclipse IDE (please use the Eclipse IDE for Enterprise Java and Web Developers). Also make sure to install the SAP Commerce plug in as stated in the information box below.

In our pre-configured Eclipse, we have initialized and pre-arranged a development workspace for you which already includes the TrainingLabTools project (contains the tools for setting up and verifying your exercises) and the **SAP Hybris Commerce Development Tools** plugin.

If you would like to install the **SAP Hybris Commerce Development Tools** plugin in your own Eclipse IDE, you can do so in the Eclipse Marketplace, just search for "SAP Commerce" and you should find it. (You can also find it here: https://developers.sap.com/trials-downloads.html)

5A Launch Eclipse

5A.1 Begin by launching Eclipse IDE (either pre-configured on the SAP VM, or installed by yourself). Double-click on the **Eclipse** icon in the eclipse-bundle directory.

Eclipse in MacOS issues:

If you're using MacOS and you can't start eclipse, it might be that the eclipse has been placed in quarantine by the operating system.

See note **3**

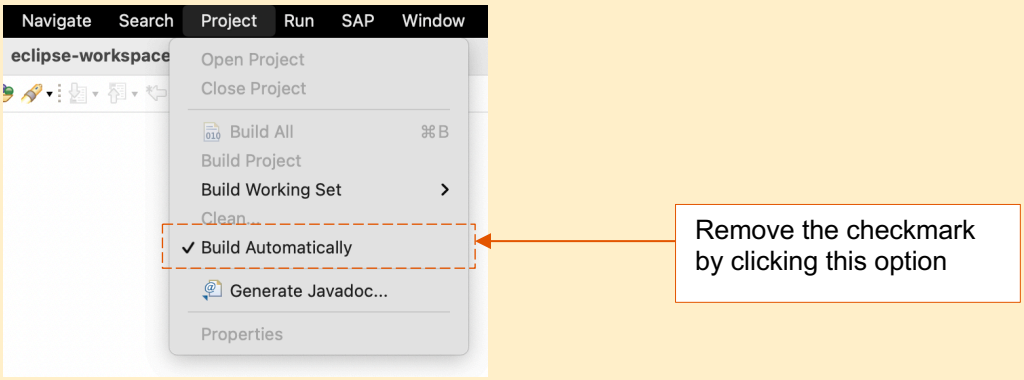
5A.2 When you first start Eclipse, it may ask for a directory to use as its workspace. If so, browse to the workspace directory we have provided at `MYPATH/workspace` and select it, then click on the **Launch** button.

5A.3 After you see the workspace open, please turn off "automatic build" according to the following step in the following technical info note:

Turn Off Automatic Build

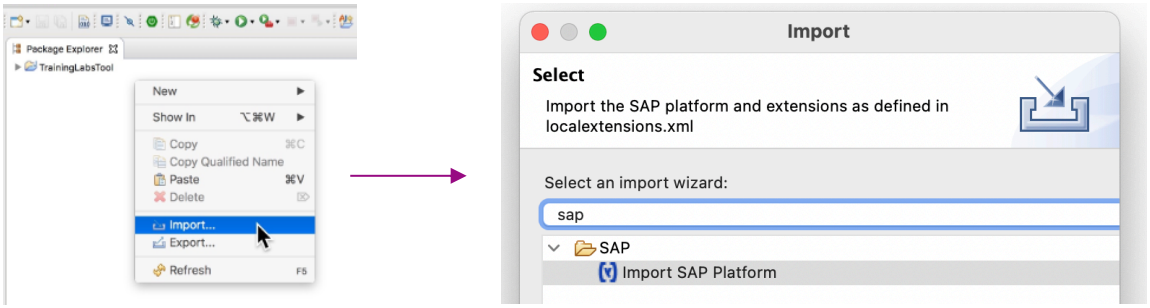
Because the SAP Commerce Cloud platform is always built using Ant, there is no point in your IDE trying to compile automatically on its own. With OOB Eclipse, though, that feature is turned on by default. We've turned it off for you in the pre-configured Eclipse. If you do need to turn it off, you

can find it in the **Project** menu (if the **Build Automatically** menu selection is checked, select it to uncheck it). See below:



5B Importing using SAP Commerce Cloud Development Tools Plug-in

5B.1 Go to the import dialog, either from the **File** menu, or by right-clicking in the *Package Explorer*. In the dialog box, type `sap` in the search box, and select the **Import SAP Platform** wizard:



5B.2 On the Import dialog, click on **Next**. Then you'll be asked to locate the *Platform Home*. Click on **Browse...**, navigate to (and select) your 'platform' folder (i.e., `MYPATH/workspace/hybris/bin/platform`).

5B.3 Click on **Finish** and watch all your projects appear in the Package Explorer. That's it!

Please note, maybe you will have to turn of the "Automatic build" again, as it could be re-enabled by the importing of the specified commerce projects.

Set the text file encoding in Eclipse to `UTF-8`. To do that, go to *Window* > *Preferences* > *General* > *Workspace* (on MacOS: *Eclipse* > *Preferences* > *General* > *Workspace*). Within the **Text file encoding** section, via the radio button and pulldown menu, check whether or not the **Default** is `UTF-8`. If it isn't, select `Other: UTF-8`

5C After Importing

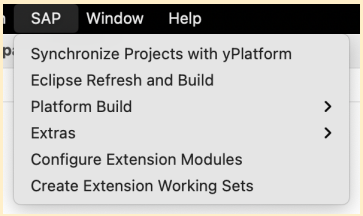
Extensions displayed in your IDE

Notice that there are far more projects (extensions) here than are explicitly listed in localextensions.xml. That's because these additional extensions are referenced as dependencies of those listed, and thus loaded dynamically. If you're importing projects manually, you need at least those in `localextensions.xml`, but it won't hurt to import the extra ones. To get a complete list of all extensions used by your system, run `ant extensionsxml` to generate the file `localextensions-generated.xml` in your config folder.

Refreshing the IDE

When adding/removing files from one of your extensions, you need to refresh the extension in Eclipse before you can see the changes.

After creating a new extension and adding it to your `localextensions.xml` file, you can import it into Eclipse by selecting **Synchronize Projects with yPlatform** in the *SAP* dropdown menu.



Clearing up compilation errors in Eclipse

Remember that when *ant* runs without returning errors, you're good to go. But Eclipse may report errors after running *SAP* > **Eclipse Refresh and Build** that aren't errors for *ant*. These errors are a result of test source folders being included in the compilation when they shouldn't be. You don't have to clear them for *ant* to work, but if they annoy you, here's how to make them go away:

The first few entries Eclipse displays inside an extension are directories included in the build path. These directories are identified by the following icon: . Right-click on the following folders in Eclipse and select *Build Path* > *Remove from Build Path:*

1. `backoffice/testsrc`
2. `basecommerce/testsrc`
3. `platform/ext/core/testsrc`
4. `voucher/testsrc`

Note that the folders in platform will show up as a single entry (for example: `ext/core/testsrc`) – you don't need to drill down to find them.

When you're done, invoke **Eclipse Refresh and Build** from the *SAP* dropdown menu.

Automated Solution Script

If you ***did NOT*** successfully complete this exercise and ***do NOT*** wish to complete this exercise manually, you can run an automated class-setup script instead.

Warning: ***DO NOT*** run the following automated setup script if you have already successfully completed this exercise:

S1 Navigate to the *Terminal* or *cmd* window where the server is running, and if it is running, stop it by entering `CTRL-C`.

S2 Navigate to *`MYPATH`*`/workspace/TrainingLabTools/class_setup` and execute:

```
ant -f class_setup_tasks.xml solution
```

Unfortunately, this won't set up your IDE. You'll still have to do that manually.

RECAP

In this exercise, you configured SAP Commerce Cloud using a recipe we provided. You imported the extensions into your site using an IDE and loaded Groovy verification scripts into the SAP Commerce platform to be persisted. You are now prepared to do the rest of the exercises.

GENERAL COLOR-CODE INFORMATION

Throughout the exercises, colored textboxes hold useful information for you to read. Any text within a red box is critical information and will have severe effects on your system or exercise success if you ignore it. Have a look at some examples below:

- Critical information/instructions:

  Please use the recommended names exactly as listed …

- Technical information about IDE details, file systems, or operational systems:

  Do not forget to configure your class path in Eclipse:

  Right click your trainingservices project and …

- Exercise notes, elaborations, or deep dives:

  The recipe contains all the information needed to…

- General notes, best/better practices, overviews, or sub-exercise summaries (for example, halfway recapitulation):

  You managed to create a new product, congratulations ☺

www.sap.com

SAP Customer Experience