Installing and Initializing SAPhybris 5.7 Commerce Using the Installer

- Install JdK1.8 and set Classpath.
- Extract the Hybris Software zip file and Goto the cmd prompt with the path
  Drive\Hybris\Installer
  Ex:C:\hybris5.7\installer.

- Install the hybris Commerceflavor using the required recipe by entering the following command:

  On Windows : **install.bat -r<*recipe_name*>**
  Ex: ***install.bat -r* b2c_telco_full**

  On Linux or Mac : **./install.sh -r<*recipe_name*>**
  Ex: **./install.sh -r *b2c_telco_full***
- Initialize hybris Commerce by entering the following command:

  On Windows**:** **install.bat -r<*recipe_name*>initialize**
  Ex: ***install.bat -r b2c_telco_full* initialize**

  On Linux or Mac: **./install.sh -r<*recipe_name*>initialize**
  Ex:- **./install.sh -r *b2c_telco_full* initialize**

- Start hybris Commerce by entering the following command:
  On Windows**:** **install.bat -r<*recipe_name*>start**
  Ex: ***install.bat -r b2c_telco_full* start**

  On Linux or Mac: **./install.sh -r<*recipe_name*>start**
  Ex:- **./install.sh -r *b2c_telco_full* start**

Environment Setup

1.Install Java

Download the latest JDK package (Java 8 is fully supported and required since Hybris 5.5.1)

2. Downloading Hybris

Download the Hybris Commerce suite and extract it.

3. Install version control tool

Install the version control tool of choice for your particular project e.g.

GIT

- sourceTree
- eGit

SVN

- subclipse
- TortoiseSVN (Windows only)
- Syncro SVN

4. Clone or Checkout project source code

- Depending on the version control system, clone or checkout the source code into the correct path according to the paths used in localextensions.xml (usually <HYBRIS_DIR>/bin/custom)
- Ensure the project specific local.properties and localextensions.xml are on the config directory (usually <HYBRIS_DIR/config> although could be another location if an HYBRIS_CONFIG_DIR environment variable is defined).

5. Compile Platform

Run "ant clean all" to do a clean build of the platform

6. Setup Database Server

- Install the database server locally. It is recommended to use the same database

server that will be used in production to identify incompatibilities as early as possible.
- Create a schema and database user using the schema, username and password that is defined in local.properties

7. Initialize Database

Run "ant initialize" to initialize the database and import the essential and project data impex files. If the project has been setup correctly we should now have a working system after this step.

8. Setup IDE

- Download and install your preferred IDE. The most common ones are Eclipse and STS.

8.1 Eclipse Workspace Setup

- Create a new workspace in Eclipse
- Import projects into Eclipse
    1. Click "File > Import" from the top menu
    2. Select"General > Existing Projects into Workspace" then browse to the Hybris Commerce home directory
    3. Select projects that are referenced in localextensions.xml
    4. Make sure that the Copy projects into workspace checkbox is**not**checked, and click Finish
    5. If the custom projects are not in the Hybris Commerce home directory follow from step 2 and select the directory of the custom projects.
- Configure Ant
    - Go to Windows->Preferences->Ant->Runtime->Ant Home and select the ant installation shipped with the platform(e.g. <HYBRIS_DIR>\bin\platform\apache-ant-1.9.X - version may be different depending on the hybris platform version)
    - Show the Ant View (Windows->Show view->Other)
    - Drag and drop the <HYBRIS_DIR>/bin/platform/build.xml file into the Ant view
    - Double click on the**all**target to run it.

How to configure your Database to use a non-default Database(Optional)

hybris comes bundled with hSQL which is not recommended for production environments. If you want to configure a different database use the steps below which use MySQL as an example.

1. Make sure that there is an installation of MySQL available for you to use.

2.Add the following entries to your **config/local.properties** file and save.

- Copy the mysql configuration code from **project.properties** file which is in platform folder and paste it in local.properties file .

- Enable mysql.mysql.allow.fractional.seconds=false by uncommiting it .(for mysql-connector-java-5.1.18 jar ,we need to enable mysql.mysql.allow.fractio-nal.seconds=false).

- Write your database schema name in place of <dbname> and write your db and give
username and password in the places of <username> and <password> Mysql

local.properties

```
 #A type of MySQL database tables, for example InnoDB for transactional or
MyISAM for non-transactional tables
mysql.tabletype=InnoDB

 Set this property to true if MySQL jar 5.6.4 or later is used. It allows to create
datetime columns which support
 fractional seconds.
mysql.allow.fractional.seconds=false

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

- Put mysql connector jar in dbdriver folder of lib which is in platform folder.. **(/opt/hybris-commerce-suite-5.7.0.0merchandisetril/hybris/bin/platform/lib/dbdriver).**

- Rebuild hybris by invoking (double-clicking) the **Ant all** task from within Eclipse.

- Go to **etc** and in that folder and go to **mysql** folder and find for **my.cnf** file and in that file
find for **max_allowed_packet** property and assign **1024** value to that property. (we can find two times the max_allowed_packet property in my.cnf file and assign 1024 to both).But this file is modified with commandprompt with admin permissions because we can't modify directly because we don't have Permissons to write

- Start the hybris Server.
- Check whether the start-up log now mentions **mysql**:

```
INFO  [hybrisserver]
INFO  [hybrisserver]
****************************************************************
INFO  [hybrisserver]
INFO  [hybrisserver] Starting up hybris server
INFO  [hybrisserver]
INFO  [hybrisserver] Configuration:
INFO  [hybrisserver]
INFO  [hybrisserver] Cluster:     disabled
INFO  [hybrisserver] Tenant:      master
INFO  [hybrisserver] OS:          Windows 7 6.1, x86
INFO  [hybrisserver] Database:    Pool: JNDI=null - mysql, table prefix:
INFO  [hybrisserver]              <username>@localhost -
jdbc:mysql://localhost/<dbname>   <-----------
INFO  [hybrisserver] Platform:    4.2.1
INFO  [hybrisserver] Java:        Sun Microsystems Inc.
INFO  [hybrisserver]              Java(TM) SE Runtime Environment, 1.6.0_14-b08
INFO  [hybrisserver]              C:/Program Files (x86)/Java/jdk1.6.0_14/jre
INFO  [hybrisserver] VM Locale:   language=en,country=US,region=
INFO  [hybrisserver] Cache:       20000 entries.
INFO  [hybrisserver] Server type: tomcat
INFO  [hybrisserver]
INFO  [hybrisserver]
****************************************************************
```

Note that if switching to a new database, the system will need to be initialized

Hybris Commerce Accelerator

The hybris Commerce Accelerator is a ready-to-use web framework that enables you to jumpstart your implementation and easily build and maintain a feature-rich omni-channel commerce solution.

Overview

Built on the SAP hybris Commerce, the hybris Commerce Accelerator allows organizations to extend functionality according to their needs.

Core Accelerator

The hybris Commerce Accelerator is a ready-to-use web framework that delivers a feature-rich omni-channel commerce solution quickly and easily. The hybris Commerce Accelerator takes full advantage of the unique flexibility and capability of the hybris Commerce Suite.

Overview of Modules used by the hybris Accelerator

The hybris Commerce Accelerator is an out-of-the-box, best-practice, e-Commerce implementation powered by the hybris Commerce Suite. Accelerator gives enterprises a faster, lower-cost implementation path resulting in a working website with a library of key components. The end-user documentation collected here guides you through the hybris Commerce Accelerator configuration

and can help you achieve your day-to-day tasks and objectives.

ACCELERATORS

There are a variety of hybris Commerce Accelerators that target specific industries, regions, and types of commerce. Select the Accelerator that best suits your organization:

B2C

The hybris Commerce Accelerator is a ready-to-use omni-channel solution that you can use to speed implementation, boost sales, and increase growth across all of your channels.

B2B

The hybris Commerce B2B Accelerator is a ready-to-use Web framework that enables you to jump-start your B2B implementation and easily build and maintain a feature-rich, omni-channel commerce solution.

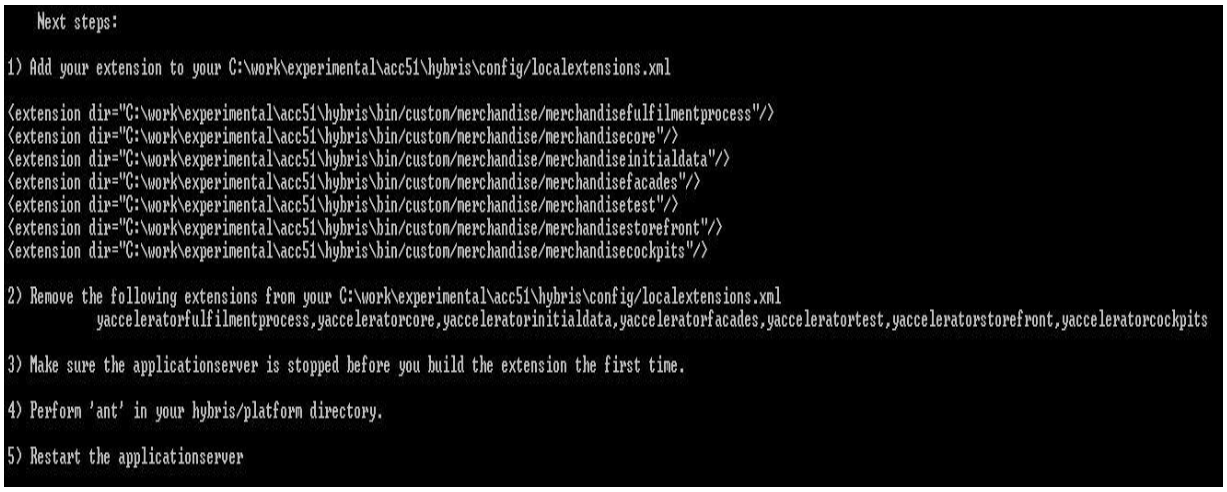
Mechandise Set Up

- Make sure Hybris 5.7.0.2 zip file is extracted
- Open cmd from platform(/opt/hybris-commerce-suite-5.5.1.0/hybris/bin/platform$ for linux)
- Set ant environment by passing the following commad

  setantenv.bat   for **Windows**

  . ./setantenv.sh for **linux**
- Run ant command and then press enter.
- Run ant clean all command.
- Run ant modulegen.
- Press enter(for B2C)
- Create an Extension eg:"pillowtalk"
- Create Pakage eg:"com.techouts.pillowtalk"
- After this one "custom" folder will be created in **bin** with the following extensions.
  - /custom/merchandise/ pillowtalkcockpits
  - /custom/merchandise/ pillowtalk core
  - /custom/merchandise/ pillowtalk facades
  - /custom/merchandise/ pillowtalk fulfilmentprocess

➢ /custom/merchandise/ pillowtalk initialdata



```
   Next steps:

1) Add your extension to your C:\work\experimental\acc51\hybris\config/localextensions.xml

<extension dir="C:\work\experimental\acc51\hybris\bin/custom/merchandise/merchandisefulfilmentprocess"/>
<extension dir="C:\work\experimental\acc51\hybris\bin/custom/merchandise/merchandisecore"/>
<extension dir="C:\work\experimental\acc51\hybris\bin/custom/merchandise/merchandiseinitialdata"/>
<extension dir="C:\work\experimental\acc51\hybris\bin/custom/merchandise/merchandisefacades"/>
<extension dir="C:\work\experimental\acc51\hybris\bin/custom/merchandise/merchandisetest"/>
<extension dir="C:\work\experimental\acc51\hybris\bin/custom/merchandise/merchandisestorefront"/>
<extension dir="C:\work\experimental\acc51\hybris\bin/custom/merchandise/merchandisecockpits"/>

2) Remove the following extensions from your C:\work\experimental\acc51\hybris\config/localextensions.xml
        yacceleratorfulfilmentprocess,yacceleratorcore,yacceleratorinitialdata,yacceleratorfacades,yacceleratortest,yacceleratorstorefront,yacceleratorcockpits

3) Make sure the applicationserver is stopped before you build the extension the first time.

4) Perform 'ant' in your hybris/platform directory.

5) Restart the applicationserver
```

➢ Also Comment the B2b extensions from localextensions.xml except B2badmincockpit .

➢ Local.properties from config must be changed to

local.properties
pillowtalkstorefront.additionalWebSpringConfigs.b2ccheckoutaddon=classpath:/b2ccheckoutaddon/web/spring/b2ccheckoutaddon-web-spring.xml

- In extensionsinfo.xml from pillowtalkstorefront copy and Replace /store in local.propertice file

extensionsinfo.xml
<webmodule jspcompile="false"webroot="/pillowtalkstorefront"/>

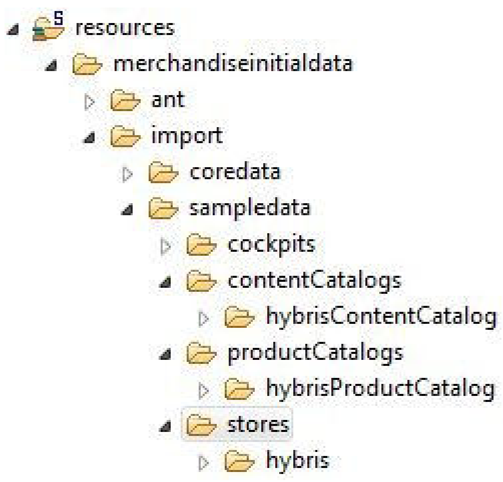
- In local.properties from config

local.properties

**merchandisestorefront.webroot=/pillowtalkstorefront
storefrontContextRoot=/pillowtalkstorefront
website.hybris.http=http://hybris.local:9001/pillowtalkstorefront
website.hybris.https=https://hybris.local:9002/pillowtalkstorefront**

- Rebuild by ant clean all ,Restart server go to HAC and perform Intilize

- In Eclipse change the Directorystructure like this



Note : And also change the coredata also.Don't fotgot to Replace the hybris with pillowtalk

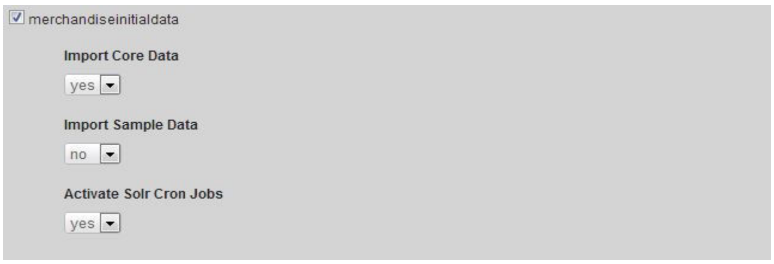
- Import the impex files in the following order

- **From core data i.e.** pillowtalkinitialdata/resources/pillowtalkinitialdata/import/coredata/stores/hybris/
- Store.impex
- Store_en.impex
- Site.impex
- Site_en.impex
- Solr.impex
- Solr_en.impex
- Solrtrigger.impex

- **From ContentCatalogData i.e.** /import/coredata/contentCatalogs/pillowtalkContentCatalog/
  - Catalog.impex
  - Catalog_en.impex
  - Cms_content.impex
  - Cms_content_en.impex
  - Email_content.impex
  - Email_content_en_impex

- **From ProductCatalogData i.e.** **/import/coredata/productCatalogs/**pillowtalk**ProductCatalog/**
  1. Catalog.impex
  2. Catalog_en.impex

- **Mobile Files i.e.** **../import/coredata/contentCatalogs/**pillowtalk**ContentCatalog/**
- Cms_mobile_content.impex
- Cms_mobile_content_en.impex
  - Run**ant all**,start server.
  - Update the project data of the pillowtalkcore extension in hac.



☑ merchandiseinitialdata

Import Core Data: yes

Import Sample Data: no

Activate Solr Cron Jobs: yes

- Go to InitialDataSystemsetup.java in eclipse from pillowtalkinitialdata and add this code

InitialDataSystemsetup.java

packagede.hybris.merchandise.initialdata.setup;
 	
...
importjava.util.Arrays;
...
 	
@SystemSetup(extension = MerchandiseInitialDataConstants.EXTENSIONNAME)
publicclassInitialDataSystemSetupextendsAbstractSystemSetup
{
...
publicstaticfinalString MERCHANDISE ="pillowtalk";
...
 	
@SystemSetup(type = Type.PROJECT, process = Process.ALL)
publicvoidcreateProjectData(finalSystemSetupContext context)
{
 	
finalList<ImportData> importData =newArrayList<ImportData>();
 	
finalImportData hybrisImportData =newImportData();
 	
hybrisImportData.setProductCatalogName(MERCHANDISE);
hybrisImportData.setContentCatalogNames(Arrays.asList(MERCHANDISE));
hybrisImportData.setStoreNames(Arrays.asList(MERCHAN

```
DISE));
importData.add(hybrisImportData);

getCoreDataImportService().execute(this, context,
importData);
getEventService().publishEvent(newCoreDataImportedEvent
(context, importData));

getSampleDataImportService().execute(this, context,
importData);
getEventService().publishEvent(newSampleDataImportedEv
ent(context, importData));
}
...
}
```

- **From SampleData i.e.**

  import/sampledata/productCatalogs/pillowtalkProductCatalog/
  - Categories.impex
  - Categories_en.impex
  - Suppliers.impex
  - **Copy Products images in Specified folder**
  - Product.impex
  - Product_media.impex
  - Products_en.impex
  - Productprices.impex
  - Product_relations.impex
  - Product_stocklevels.impex
  - Warehouse.impex

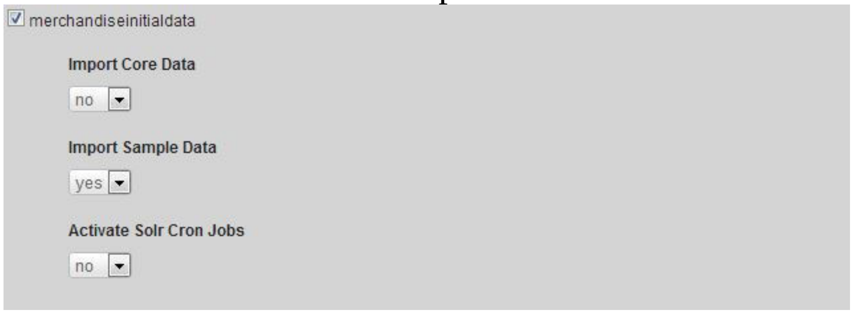
- Add the following bean definition in pillowtalk**core/resources/**pillowtalk**core-spring.xml**

pillowtalk**core-spring.xml**
<beanid="pillowtalkCategorySource"parent="abstractCategorySource"> <propertyname="rootCategory"value="1"/> </bean> <beanid="pillowtalksBrandCategorySource"parent="abstractCategorySource"

```
>
<propertyname="rootCategory"value="brands"/>
</bean>

<beanid="pillowtalkCategoryCodeValueProvider"parent="abstractCategoryC
odeValueProvider">
<propertyname="categorySource"ref="pillowtalkCategorySource"/>
</bean>
<beanid="pillowtalkBrandCategoryCodeValueProvider"parent="abstractCate
goryCodeValueProvider">
<propertyname="categorySource"ref="pillowtalkBrandCategorySource"/>
</bean>
<beanid="pillowtalkCategoryNameValueProvider"parent="abstractCategoryN
ameValueProvider">
<propertyname="categorySource"ref="pillowtalkCategorySource"/>
</bean>
<beanid="pillowtalkBrandCategoryNameValueProvider"parent="abstractCate
goryNameValueProvider">
<propertyname="categorySource"ref="pillowtalkBrandCategorySource"/>
</bean>
```

- Import productcockpit.user from **.../import/sampledata/cockpits/productcockpit/productcockpit-users.impex**
- Restart server
- Go to hac and update initial data



☑ merchandiseinitialdata

Import Core Data: no

Import Sample Data: yes

Activate Solr Cron Jobs: no

❖ **Copy Content images in Specified folder**

- **Import CMS site content i.e. from**
  /import/sampledata/contentCatalogs/pillowtalk**ContentCatalog/**
  - Images
  - Cms_content.impex
  - Cms_content_en.impex
  - Email.content.impex

- Email_content_en.impex
- Import Cmscockpit_users.impex from

  /import/sampledata/cockpits/cmscockpit/cmscockpit-users.impex

- Add **site-hybris.properties in merchandisestorefront/web/webroot/WEB-INF/messages/site-hybris.properties.**
- Ant clean all
- Start server
- Run your merchandise store.

Note for detailed impex files follow this link
https://wiki.hybris.com/display/tr52/hybris+5+Developer+Training+Trails+-+Part+II+-+Commerce
https://wiki.hybris.com/display/accdoc50to56/b2ccheckout+AddOn

Directory Structure of SAP Hybris Commerce :

All extensions developed by hybris are grouped in specific directories under the /bindirectory to distinguish between the different kinds of extensions that hybris offers.

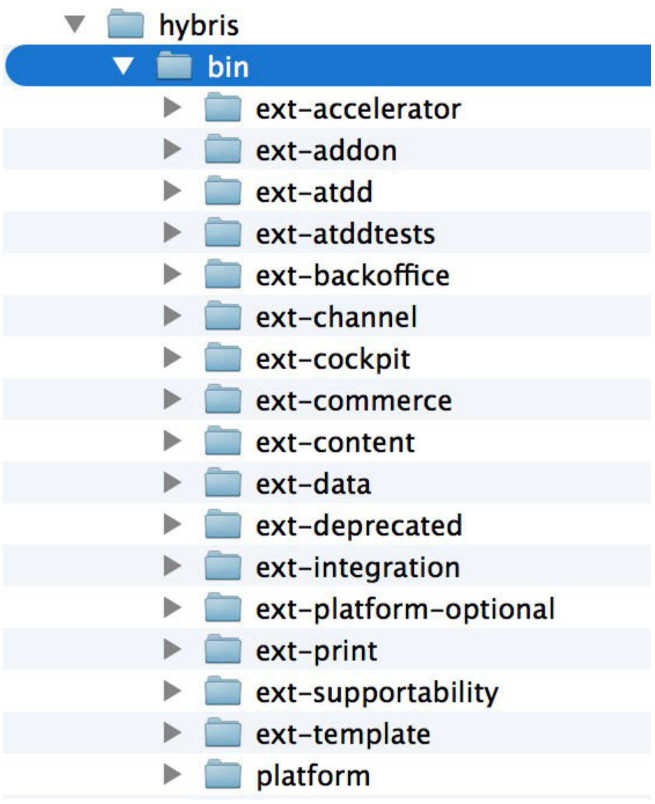
Additionally, distinguishing the /bindirectory from the other directories is beneficial for both partner developers and system administrators when updating SAP hybris Commerce.

Directory Structure

The directory structure of hybris Commerce has a clear distinction between binary files stored in the **/bin**directory and the other files like:

- Configuration files stored in the **config** folder
- Data files stored in the **data** folder
- Log files stored in the **log** folder
- Temporary files stored in the **temp** folder.

The advantage of having the binaries directory separated from the rest of the directories is that you can easily integrate updates of hybris Commerce. Your customized configuration files will not be modified by the update process.



```
▼ hybris
  ▼ bin
    ▶ ext-accelerator
    ▶ ext-addon
    ▶ ext-atdd
    ▶ ext-atddtests
    ▶ ext-backoffice
    ▶ ext-channel
    ▶ ext-cockpit
    ▶ ext-commerce
    ▶ ext-content
    ▶ ext-data
    ▶ ext-deprecated
    ▶ ext-integration
    ▶ ext-platform-optional
    ▶ ext-print
    ▶ ext-supportability
    ▶ ext-template
    ▶ platform
```

Extension Packaging

An extension can be a part of multiple modules. To ensure that each extension exists only once, but may be used by multiple modules, we group the extensions developed by hybris in the directories under the **/bin** directory as presented in the Figure on the right.

Easier Development

One of the benefits of this horizontal directory structure is the fact that it is easier for developers to work with the hybris Commerce system. Each extension is a separate Eclipse project with correctly configured dependencies, and can be opened as needed in your IDE.

Technical Aspects

- The**/bin**directory and its subdirectories do not contain any customizable configuration data. Never change anything within this directory. During the update process, the**/bin**directory and its subdirectories may be completely replaced with newer version of hybris Commerce.
- Custom configuration data such as the license,**local.properties**,and**localextensions.xml**files, must reside in

the/**config**directory.

- If noconfigdirectory is available, you will be asked on the first**ant**call about which configuration template should be used;developorproduction. Refer to theConfiguration Templatesdocument for details.
- Theconfigdirectory for development is an Eclipse project and should be added as a separate project.

Directory Structure Overview

Select a tab to view the Directory Structure for a particular Release version.

Directory	Description	EnivironmentVaraible
/bin	Contains the hybris Platform directories, the template directory, and the hybris extensions directory. It may also contain the directory for partner extensions or custom extensions made by customers for their own use.	*${HYBRIS_BIN_DIR}*
• /bin/custom	This directory is created during the process of creating the custom extensions. It should contain your own project extensions. For details about creating extensions see the Creating a New Extension document.	
• /bin/ext-accelerator	This directory contains **acceleratorcms, acceleratorfacades, acceleratorservices, acceleratorstorefrontcommons, alipay, b2bacceleratorfacades, b2bacceleratorservices, b2bpunchout, chinaacceleratorfacades, chinaacceleratorservices, savedorderforms, timedaccesspromotionsfacades,**	

	timedaccesspromotionsservices, timedaccesspromotionsserviceshmc extensions.
• /bin/ext-addon	This directory contains addon-related extensions.
• /bin/ext-atdd	This directory contains the ATDD engine.
• /bin/ext-atddtests	This directory contains the ATDD tests for particular modules.
• /bin/ext-backoffice	This directory contains, **alipaymentbackoffice, b2bcommercebackoffice, backoffice, basecommercebackoffice, chinaacceleratorbackoffice, commercesearchbackoffice, commerceservicesbackoffice, ordermanagementbackoffice, promotionsbackoffice, solrfacetsearchbackoffice, timedaccesspromotionsbackoffice, voucherbackoffice, warehousingbackoffice, xyformsbackoffice, ybackoffice** extensions.
• /bin/ext-channel	This directory contains **cscockpit, instore, mobileoptionals, mobileservices** extensions.
• **/bin/ext-cockpit**	This directory contains the following cockpit extensions: **admincockpit**, **cockpit**, **mcc**, **reportcockpit, ycockpit**.
• **/bin/ext-commerce**	This directory contains commerce-related extensions.
• **/bin/ext-content**	This directory contains **bmecat, bmecathmc, classificationsystems, classificationsystemshmc, cms2,**

	cms2lib, cmscockpit, importcockpit, liveeditaddon, productcockpit, productcockpitsampledata.
• **/bin/ext-data**	This directory contains sample data extensions.
• **/bin/ext-deprecated**	This directory contains extensions which are to be deprecated along with the next release.
• /bin/ext-eventtracking	This directory contains **eventtrackingmodel, eventtrackingpublisher, eventtrackingservices**.
• /bin/ext-incubator	This directory contains **deltadetection, y2ysync, y2ysync-datahub-ext, y2ysyncdemoelectronics, y2ysynchmc, yaasconnect.**
• **/bin/ext-integration**	This directory contains extensions used for integration of the hybris Commerce Suite with SAP systems. For the full list of extensions, see Extensions and AddOns of Hybris-SAP Solution Integration.
• **/bin/ext-platform-backoffice**	This directory contains **mediaconversionbackoffice** and **platformbackoffice**.
• **/bin/ext-platform-optional**	This directory contains optional platform extensions.
• **/bin/ext-print**	This directory contains **print**, **printcockpit** and **printhmc** extensions
• **/bin/ext-supportability**	This directory contains **hybrisdatasupplier, hybrisrootcauseanalysis** and **hybristransportandchange** extensions.
• **/bin/ext-template**	This directory contains all **extgen**

	templates.	
• **/bin/platform**	This directory contains the actual Platform functionalities. It includes core extensions, the build framework, custom extension templates in **/extgen**, and the application server directories.	
/config	The directory contains your custom configuration files for the hybris Commerce Suite, such as: **local.properties**, **localextensions.xml**, and **hybrislicence.jar**. This directory also contains the files for the customization mechanism of the hybris Commerce Suite.	*${HYBRIS_CONFIG_DIR}*
/data	This directory contains runtime data, such as: • Media files, such as product pictures. See also Media folders. • LuceneSearch indexes • HSQLDB files	*${HYBRIS_DATA_DIR}*
/log	This directory contains log files from the hybris Server, JDBC logging, and so on.	*${HYBRIS_LOG_DIR}*
/roles	This directory is empty but after you create a role, this role's directories will be kept here.	*${HYBRIS_ROLES_DIR}*
/temp	This directory contains temporary files.	*${HYBRIS_TEMP_DIR}*

Ant Commands

ant build:

It is used to build the xml and .java files.
It doesn’t erase the .class and build files, it only update the changes in xml.
It can be always used when we want xml and java files.
If any changes in xml and java build the server.

ant clean:
It is used to remove all the generated .class files.

ant clean all:
Cleans all the generated class files and build the entire hybris suite.

ant all:
It builds the entire commerce suite but didn’t erase any .class file.

ant initialize:
It is used to initialize the server, means it will delete all the item type(tables) and create the item types from bean name.

ant update system:
It is used to update the table, where required(when any attribute(column) is changed, added or deleted) then it will effect the updated table.
ant modulegen :
To create our own project extentions (7 extentions)
ant extgen:
To create extention or addon it is depend on which template using
ant addoninstall:
The addoninstall tool allows you to configure an AddOn for a storefront. It also adds the AddOn to the **extensioninfo.xml** file of the storefront, and generates the relevant project.propertiesfile for the AddOn.

The following is an example of the **ant addoninstall** command

```
ant addoninstall -Daddonnames="randomimageaddon"-
DaddonStorefront.yacceleratorstorefront="merchandisestorefront"
```

addonuninstall

The addonuninstall **tool can be used to remove an AddOn from the storefront**

```
ant addonuinstall -Daddonnames="randomimageaddon"-
DaddonStorefront.yacceleratorstorefront="merchandisestorefront"
```

ant runcronjob
hybris Commerce Suite allows running cron jobs using an Ant target.

ant yunitinit
To Initializes JUnit tenant
ant yunitupdate
To Update JUnit tenant