



Accelerators

Generated on: 2021-10-21 10:01:55 GMT+0000

SAP Commerce | 1905

PUBLIC

Original content: <https://help.sap.com/viewer/4c33bf189ab9409e84e589295c36d96e/1905/en-US>

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the <https://help.sap.com/viewer/disclaimer>.

B2C Accelerator Implementation

The B2C Accelerator module allows you to remove apparel store functionality from your installation of SAP Commerce Accelerator.

[Customizing the B2C Accelerator](#)

This procedure describes how to create a customized B2C Accelerator using the B2C Accelerator installation recipe.

[B2C Store Extensions Related Data](#)

The other extensions providing data for the project are the `apparelstore` and `electronicsstore` AddOn extensions and `yacceleratorinitialdata` extension. They come with a set of only Project data.

[Removing Apparel and the Apparel Stores](#)

You can remove apparel support from your SAP Commerce Accelerator.

Customizing the B2C Accelerator

This procedure describes how to create a customized B2C Accelerator using the B2C Accelerator installation recipe.

Prerequisites

Before running the `modulegen` tool, download the latest version of the SAP Commerce solution. For more details, see [Download](#).

Once you have downloaded SAP Commerce, extract the SAP Commerce ZIP file in `<BASE_DIR>`.

→ Tip

If you are in a training class, the SAP Commerce ZIP file will be provided by your trainer.

Procedure

1. Open a command line and go to the installer directory of your SAP Commerce instance.
 - Windows: `cd <HYBRIS_HOME_DIR>\installer`
 - Unix/MacOS: `cd <HYBRIS_HOME_DIR>/installer`
2. Build SAP Commerce using the `b2c_acc_plus` recipe.
 - Windows: `install.bat -r b2c_acc_plus -A initAdminPassword=<your_password>`
 - Unix/MacOS: `./install.sh -r b2c_acc_plus -A initAdminPassword=<your_password>`
3. Navigate to the `<HYBRIS_BIN_DIR>/platform` directory and set your ant environment by entering the following command.
 - Windows: `setantenv.bat`
 - Unix/MacOS: `./setantenv.sh`
4. Run the `ant modulegen` command and modify the `input.name` and `input.package` parameters as required.


```
ant modulegen -Dinput.module=accelerator -Dinput.name=training -Dinput.package=de.hybris.train
```

Where:

- `input.module` parameter: Configures the **modulegen** task to use the `accelerator` module.
- `input.name` value: Indicates the prefix that is added to the new extensions generated by **modulegen**.

- `input.package` value: Defines the default Java package prefix.
- `input.template` parameter: Defines how to use the default SAP Commerce configuration.

i Note

The training variable can be substituted for any string value.

5. Remove all the `yaccelerator*` extensions from the `<HYBRIS_CONFIG_DIR>/localextensions.xml` file.
6. Add the new extensions to the `<HYBRIS_CONFIG_DIR>/localextensions.xml` file.

```
<extension dir="${HYBRIS_BIN_DIR}/custom/training/trainingbackoffice"/>
  <extension dir="${HYBRIS_BIN_DIR}/custom/training/trainingfulfilmentprocess"/>
  <extension dir="${HYBRIS_BIN_DIR}/custom/training/traininginitialdata"/>
  <extension dir="${HYBRIS_BIN_DIR}/custom/training/trainingstorefront"/>
  <extension dir="${HYBRIS_BIN_DIR}/custom/training/trainingtest"/>
```

i Note

If you include the `trainingtest` extension, it imports test data, which can be inaccurate in a real-world scenario. Only include the `trainingtest` extension if you intend to perform tests with this Accelerator.

7. Re-install the B2C Accelerator AddOns on the training storefront by running the following ant command from the `<HYBRIS_BIN_DIR>/platform` folder.

```
ant reinstall_addons -Dtarget.storefront=trainingstorefront
```

8. Add the following properties to `<HYBRIS_CONFIG_DIR>/local.properties` file.

```
website.electronics.http=http://electronics.local:9001/trainingstorefront
website.electronics.https=https://electronics.local:9002/trainingstorefront
website.apparel-uk.http=http://apparel-uk.local:9001/trainingstorefront
website.apparel-uk.https=https://apparel-uk.local:9002/trainingstorefront
website.apparel-de.http=http://apparel-de.local:9001/trainingstorefront
website.apparel-de.https=https://apparel-de.local:9002/trainingstorefront
piwik.tracker.url.electronics=http\://electronics.local\9002/trainingstorefront/ev
piwik.tracker.https.url.electronics=https\://electronics.local\9002/trainingstorefro
piwik.tracker.url.apparel-uk=http\://apparel-uk.local\9002/trainingstorefront/events
piwik.tracker.https.url.apparel-uk=https\://apparel-uk.local\9002/trainingstorefront.
piwik.tracker.url.apparel-de=http\://apparel-de.local\9002/trainingstorefront/events
piwik.tracker.https.url.apparel-de=https\://apparel-de.local\9002/trainingstorefront.
```

9. Run `ant clean all` and `ant initialize` from the `<HYBRIS_BIN_DIR>/platform` directory.

Windows:

```
cd %HYBRIS_HOME_DIR%\hybris\bin\platform & ant clean all
ant initialize
```

Unix/MacOS

```
cd $HYBRIS_HOME_DIR/hybris/bin/platform; ant clean all
ant initialize
```

10. Start the SAP Commerce server by running the following command from the `<HYBRIS_BIN_DIR>/platform` folder.

- Windows: `hybrisserver.bat`
- Unix/MacOS: `./hybrisserver.sh`

11. If necessary, allow blocked libraries to run in MacOS Catalina. For further information, see [Allowing Apps on MacOS](#).

12. Add the following to your hosts file.

```
127.0.0.1 localhost electronics.local apparel-uk.local apparel-de.local
```

You can access the B2C Accelerator electronics and apparel storefronts with the following URLs:

- <https://electronics.local:9002/trainingstorefront/>
- <https://apparel-uk.local:9002/trainingstorefront/>
- <https://apparel-de.local:9002/trainingstorefront/>

Related Information

[Modulegen](#)

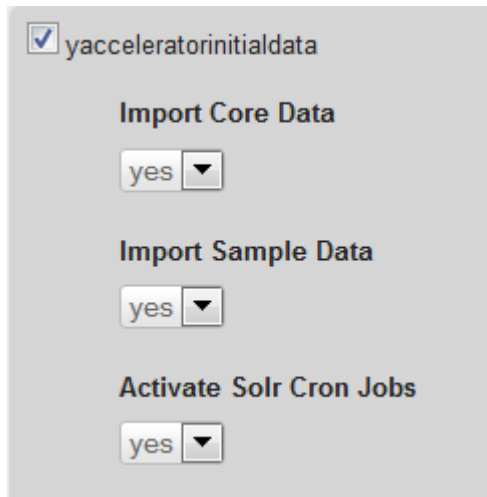
[Environment Variables](#)

B2C Store Extensions Related Data

The other extensions providing data for the project are the apparelstore and electronicsstore AddOn extensions and yacceleratorinitialdata extension. They come with a set of only Project data.

B2C Store Extensions Project Data

Project data setup for the yacceleratorinitialdata extension provides the following options:



☒ yacceleratorinitialdata

Import Core Data

yes ▼

Import Sample Data

yes ▼

Activate Solr Cron Jobs

yes ▼

Import Core Data loads:

- Essential data: warehouses, delivery modes, and currencies
- CMS templates
- Empty catalogs
- Email templates

Import Sample Data loads:

- Products
- Cockpits users
- CMS content
- Promotions
- Points of service

Activate Solr Cron Jobs runs the Solr cron jobs.

Project data setup for the sample data extensions contain:

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

10/21/2021

- Site and store configuration: catalogs, classifications, countries, languages, currencies, delivery options, themes, media formats, and tax groups.
- Base CMS content: page templates, page types, pages, components, and cockpit configurations.
- Solr search configuration: index configurations, indexed types and properties, ranges, queries, and sorts.
- Products
- Product categories
- Promotions
- Product and category media
- Product stock levels
- CMS content, including sample components, pages, images, and paragraphs.
- Email content
- Points of Service (POS): stores and associated media.
- Cockpit users
- BTG rules
- Sample product reviews

Resource Folder Structure

The resource folders of the `apparelstore` and `electronicsstore` AddOn extensions are divided into two parts :

1. Core data: create empty sites, empty catalogs, and essential data
2. Sample data: populate catalogs with products, import users, and user rights

Products

The sample data provided with the SAP Commerce Accelerator demonstrates two approaches to defining a product catalog in the system:

- Using [Classifications](#) in **electronics** shop
- Using [Variants](#) in **apparel** shop

CMS Content

The sample CMS content for both sites is almost identical but using different media and themes in the two sites makes them look totally different.

Points of Sale

Points of Sale are loaded into the Electronics shop, Apparel UK and Apparel DE separately, so searching in one Apparel site, for example, does not return results from the other.

Promotions

Sample promotions are included that apply for specific products, specific categories and across entire orders.

Related Information

[yacceleratorcore Extension](#)

[acceleratorservices Extension](#)

[apparelstore Extension](#)

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

Removing Apparel and the Apparel Stores

You can remove apparel support from your SAP Commerce Accelerator.

The SAP Commerce Accelerator offers a reference storefront with an Apparel functionality that supports product variants out of the box. If your project does not require apparel support, you can remove this functionality. This topic explains how to remove Apparel from the Commerce Accelerator source code, which includes the Apparel data model, any Apparel services and facades, as well as the Apparel sample storefronts configuration.

Before starting the workflow described here, see [Customizing the Accelerator with extgen and modulegen](#) to get an overview of working with template extensions.

The procedure for removing the Apparel features depends on the extensions they are related to. The following sections provide the procedure for each extension.

yacceleratorcore Extension

Removing Apparel features related to the yacceleratorcore extension is executed in several steps and in a specific order. Remove Apparel features in the following order:

1. Remove the data model.
2. Hot folder batch import
3. Remove beans related to Solr Search

Remove the Data Model

To remove the data model, you start by removing the data model changes and any associated configurations for the cockpits as follows.

1. In the yacceleratorcore/resources/yacceleratorcore-items.xml file, remove everything inside the **Apparel** typegroup :

```
<typegroup name="Apparel">
```

```
<itemtype code="ApparelProduct" extends="Product"
autocreate="true" generate="true"
jaloclass="de.hybris.platform.yacceleratorcore.model.ApparelProduct"
<description>Base apparel product extension type
<attributes>
<attribute qualifier="genders" type="GenderList"
<description>List of genders that the ApparelProduct has
<modifiers />
<persistence type="property" />
</attribute>
</attributes>
</itemtype>
```

```
<itemtype code="ApparelStyleVariantProduct" extends="ApparelProduct"
autocreate="true" generate="true"
jaloclass="de.hybris.platform.yacceleratorcore.model.ApparelStyleVariantProduct"
<description>Apparel style variant type that extends ApparelProduct
<attributes>
<attribute qualifier="style" type="LocalizedString"
metatype="VariantAttributeDescriptor"
<description>Colour/Pattern of the product.
<modifiers />
<persistence type="property" />
</attribute>
```

```
<attribute qualifier="swatchColour" type="java.lang.String" />
```

```

<description>A normalised colour mapping to a
<modifiers />
<persistence type="property" />
</attribute>
</attributes>

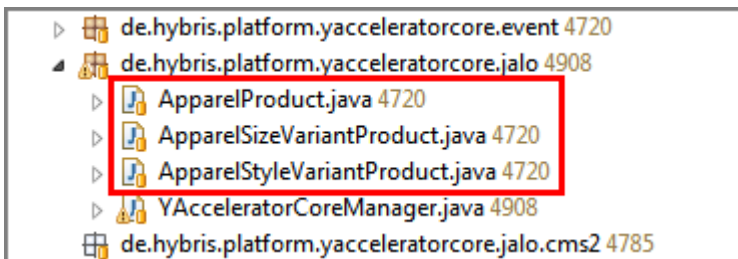
</itemtype>

<itemtype code="ApparelSizeVariantProduct" ex
autocreate="true" generate="true"
jaloclass="de.hybris.platform.yacceleratorcore
<description>Apparel size variant type that c
<attributes>
<attribute qualifier="size" type="localized:j
metatype="VariantAttributeDescriptor">
<description>Size of the product.</description
<modifiers />
<persistence type="property" />
</attribute>
</attributes>

</itemtype>
</typegroup>

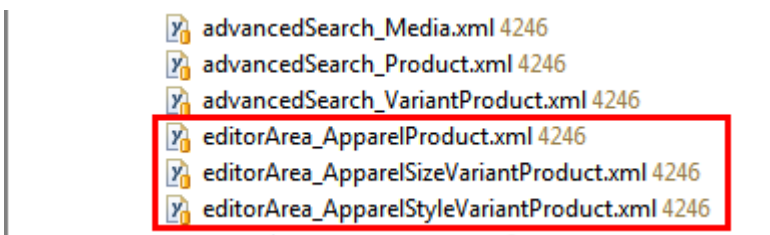
```

2. Remove the **ApparelProduct**, **ApparelSizeVariantProduct**, and **ApparelStyleVariantProduct** classes from `yacceleratorcore/src/de/hybris/platform/yacceleratorcore/jalo`.



3. Remove the following Apparel-specific Product Cockpit editor configuration XML files from the `yacceleratorcockpits/resources/yacceleratorcockpits-config/cockpitgroup` folder:

- `editorArea_ApparelProduct.xml`
- `editorArea_ApparelSizeVariantProduct.xml`
- `editorArea_ApparelStyleVariantProduct.xml`



4. Remove the **BatchIntegrationTest.testVariant** test method from the `yacceleratorcore` extension.
5. Execute the `ant clean all` command in the `yacceleratorcore` extension to ensure the extension still compiles.

Hot Folder Batch Import

You must remove the back import instance for the Apparel store. To do so, remove the `yacceleratorcore/resources/yacceleratorcore/integration/hot-folder-store-apparel-spring.xml` file.

Solr Search

When you delete the store content, the scripts that create the Solr index are removed. You must also remove specific value-provider Spring beans from the yacceleratorcore extension application context.

1. Remove the **CategorySource** beans from the yacceleratorcore/resources/yacceleratorcore-spring.xml file:

```
<bean id="apparelCategorySource" parent="abstractCategorySource">
    <property name="rootCategory" value="categories"/>
</bean>
<bean id="apparelBrandCategorySource" parent="abstractCategorySource">
    <property name="rootCategory" value="brands"/>
</bean>
```

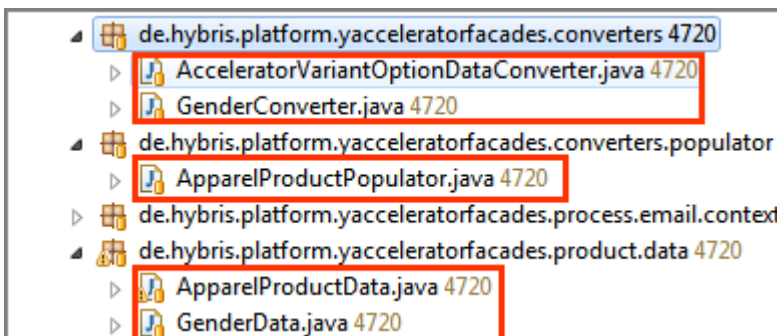
2. Remove the **ValueProvider** beans from the yacceleratorcore/resources/yacceleratorcore-spring.xml:

```
<bean id="apparelCategoryCodeValueProvider" parent="abstractCategoryCodeValueProvider">
    <property name="categorySource" ref="apparelCategorySource"/>
</bean>
<bean id="apparelBrandCategoryCodeValueProvider" parent="abstractCategoryCodeValueProvider">
    <property name="categorySource" ref="apparelBrandCategorySource"/>
</bean>
<bean id="apparelCategoryNameValueProvider" parent="abstractCategoryNameValueProvider">
    <property name="categorySource" ref="apparelCategorySource"/>
</bean>
<bean id="apparelBrandCategoryNameValueProvider" parent="abstractCategoryNameValueProvider">
    <property name="categorySource" ref="apparelBrandCategorySource"/>
</bean>
```

yacceleratorfacades Extension

Most of the code in the yacceleratorfacades extension template is provided to support Apparel functionality. This procedure assumes that you have not adapted any of the code you plan to delete for purposes other than Apparel.

1. Remove the **AcceleratorVariantOptionDataConverter** and **GenderConverter** classes from the yacceleratorfacades/src/de/hybris/platform/yacceleratorfacades/converters folder of the yacceleratorfacades extension.
2. Remove the **ApparelProductPopulator** class from the yacceleratorfacades/src/de/hybris/platform/yacceleratorfacades/converters/populator folder of the yacceleratorfacades extension.
3. Remove the **ApparelProductData** and **GenderData** classes from the yacceleratorfacades/src/de/hybris/platform/yacceleratorfacades/product/data folder of the yacceleratorfacades extension:



4. Remove the Spring bean configuration for all of these classes in the yacceleratorfacades/resources/yacceleratorfacades-spring.xml file.

```
<alias name="acceleratorVariantOptionDataConverter" alias="variantOptionDataConverter"/>
<bean id="acceleratorVariantOptionDataConverter" parent="abstractVariantOptionDataConverter">
    <property name="mediaService" ref="mediaService"/>
</bean>
```



```

<property name="mediaContainerService" ref="mediaContainerService"/>
<property name="typeService" ref="typeService"/>
<property name="imageFormatMapping" ref="imageFormatMapping"/>
<property name="variantAttributeMapping">
  <map>
    <entry key="ApparelStyleVariantProduct.style">
      </map>
    </property>
  </bean>

  <alias name="acceleratorGenderConverter" alias="acceleratorGenderConverter"/>
  <bean id="acceleratorGenderConverter" class="de.hybris.platform.accelerator.converter.GenderConverter"/>
  <property name="typeService" ref="typeService"/>
</bean>

  <bean id="apparelProductPopulator" class="de.hybris.platform.accelerator.converter.ApparelProductPopulator"/>
  <property name="genderConverter" ref="genderConverter"/>
</bean>

  <alias name="acceleratorProductConverter" alias="acceleratorProductConverter"/>
  <bean id="acceleratorProductConverter" class="de.hybris.platform.accelerator.converter.ApparelProductConverter"/>
  <property name="apparelProductPopulator" ref="apparelProductPopulator"/>
</bean>

```

5. Execute the command `ant clean all` in the `yacceleratorfacades` extension to ensure the extension still compiles.

yacceleratorstorefront Extension

The storefront controllers are agnostic of apparel, therefore few changes are required here.

Spring Application Context

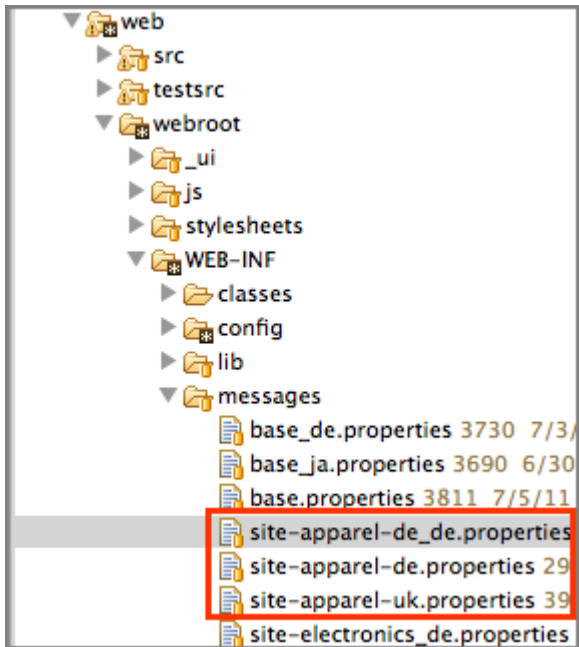
In the `yacceleratorstorefront/web/webroot/WEB-INF/config/spring-mvc-config.xml` file of the `yacceleratorstorefront` extension, remove the following line:

```
<entry key="size" value-ref="sizeAttributeComparator"/>
```

Site Resource Files

Remove the following apparel site resource files, which are located in the `yacceleratorfacades/web/webroot/WEB-INF/messages` folder:

- `site-apparel-de_de.properties`
- `site-apparel-de_en.properties`
- `site-apparel-uk_en.properties`



apparelstore Extension

Remove the apparelstore extension from the localextensions.xml file.

Recompiling

Execute the command `ant clean all` in the apparelstore extension to ensure the extension still compiles.

Reinitializing

Reinitialize the SAP Commerce Platform database to remove the apparel data model from the type system and any orphaned data no longer added by the scripts. For more information on initialization, see the **Initialize System** section of the [Initialization and Update of the SAP Commerce](#) document.

Related Information

[yacceleratorcore Extension](#)

[acceleratorfacades Extension](#)

[yacceleratorstorefront Extension](#)

[apparelstore Extension](#)

[electronicsstore Extension](#)