

# SAP COMMERCE CLOUDINARY EXTENSION SETUP GUIDE

## TABLE OF CONTENTS

<b>Introduction</b>	<b>3</b>
<b>Pre-requisites</b>	<b>3</b>
<b>B2C Accelerator Setup Steps</b>	<b>4</b>
Download and Extract	4
Clone Repository	4
Create Symlink	4
Deploy the JDBC JAR	5
Create database and user	5
Install Recipe	5
Install Required Addon	6
Start SAP Commerce Server	7
<b>B2B Accelerator Setup Steps</b>	<b>8</b>
Download and Extract	8
Clone Repository	8
Create Symlink	8
Deploy the JDBC JAR	9
Create database and user	9
Install Recipe	9
Install Required Addon	10
Start SAP Commerce Server	11

<b>Uninstall SAP Commerce B2C Cloudinary Extension</b>	<b>12</b>
Uninstall Extension	12
Start SAP Commerce Server	13
<b>Uninstall SAP Commerce B2B Cloudinary Extension</b>	<b>14</b>
Uninstall Extension	14
Start SAP Commerce Server	15

## Introduction

The Clouinary SAP Commerce Integration Setup Guide is for technical users who want to install the integration as part of their SAP Commerce system setup. The supported versions are from 1811 and later.

## Pre-requisites

- JAVA 11
- MySQL 5.7
- SAP Commerce Suite 1811 or later
- MySQL DB driver

## B2C Accelerator Setup Steps

Follow the steps below if you are installing the extensions on a new system and don't have any existing SAP Commerce setup.

### Download and Extract

To install the CX suite on your machine, follow these steps.

For CX Commerce 2005 Local Environment Setup, use the latest 2005 CX Commerce. This suite contains both CX Commerce and CEP extensions.

1. Obtain the hybris Commerce package
2. Extract the tar package in your chosen location (e.g. /home/user/ and rename it as clouinary).

### Clone Repository

To clone the Clouinary repository you need to have GitHub repository access.

Clone the repository to the custom (*{your sap cx base directory}/bin/custom*) folder from this GitHub link, and switch to the development branch:

```
https://github.com/clouinary/clouinary_sap_commerce.git
```

Ask your admin for access if required.

```
git clone https://github.com/clouinary/clouinary_sap_commerce.git custom
```

### Create Symlink

Change directory to *{your sap cx base directory}/hybris* and execute this command:

```
In -s {your sap cx base directory}/bin/custom/ext-config config
```

For windows:

```
mklink /D [Destination] [Original source]
```

## Deploy the JDBC JAR

In order to install the connector:

1. Download [mysql-connector-java-5.1.37-bin.jar](#)
2. Open a File Explorer.
3. Copy mysql-connector-java-5.1.37-bin.jar to  
*/home/user/cloudinary/hybris/bin/platform/lib/dbdriver.*

## Create database and user

Connect to your local MySQL installation and create a database (it'll be populated during initialize) and a user:

```
CREATE DATABASE cloudinary CHARACTER SET=utf8 COLLATE=utf8_bin;  
CREATE USER 'cloudinary'@'localhost' IDENTIFIED BY 'cloudinary'; GRANT ALL PRIVILEGES ON cloudinary.* TO 'cloudinary'@'localhost' WITH GRANT OPTION; flush privileges;
```

## Install Recipe

A Clouinary recipe is required for installing and initializing required Clouinary extensions and configuration.

Copy cloudinary\_b2c.zip from *{your sap cx base directory}/recipes* and extract it to your local CX commerce recipes folder.

Open a command line and go to the “installer” directory of your SAP Commerce instance.

Execute the following commands to install and initialize the recipe:

- On Windows: `install.bat -r cloudinary_b2c`
- On Linux or Mac: `./install.sh -r cloudinary_b2c`

## Install Required Addon

```
ant addoninstall -Daddonnames="cmsoccaddon"  
-DaddonStorefront.ycommercewebservices="cloudinarymediawebservices"
```

Initialize SAP Commerce by entering the following command:

- On Windows: install.bat -r cloudinary initialize
- On Linux or Mac: ./install.sh -r cloudinary initialize

Follow the steps below if you are installing the extensions on an existing SAP Commerce system.

1. Add the following Clouinary extensions to your localextensions.xml file:

```
<extension name=cmsoccaddon />  
<extension name='mediaconversion' />  
<extension name='mediaconversionbackoffice' />  
<extension name='cloudinarymediacore' />  
<extension name='cloudinarymediabackoffice' />  
<extension name='cloudinarymediasmartedit' />  
<extension name='cloudinarymediawebservices' />  
<extension name='cloudinarymediafacades' />  
<extension name='cloudinarymediaaddon' />
```

2. Install the required addons below in your platform directory:

```
ant addoninstall -Daddonnames="cmsoccaddon"  
-DaddonStorefront.ycommercewebservices="cloudinarymediawebservices"
```

```
ant addoninstall -Daddonnames="cloudinarymediaaddon"  
-DaddonStorefront.yacceleratorstorefront="<your custom b2c storefront>"
```

3. In the cloudinarymediaaddon project.properties change the property from  
“**yacceleratorstorefront.additionalWebSpringConfigs.cloudinarymediaaddon**” to  
<your custom b2c storefront  
extension>.**additionalWebSpringConfigs.cloudinarymediaaddon**
4. Execute a platform build using “ant clean all”
5. Perform a SAP Commerce system update

## Start SAP Commerce Server

To perform this step, we use the Hybris Server, which is a pre-configured application server based on Apache Tomcat.

The hybris server has two modes of operation:

- *Normal*: Set by default for the hybris server.
- *Debug*: Only used with the develop configuration template (the one we chose earlier). The debug mode is slower in performance but is useful for developing using the external debugging tools (for example Eclipse IDE with compatible Java debugger).

To start the server, from the same terminal shell:

1. Navigate to the `/bin/platform` directory and run:
  - a. `$ ./hybrisserver.sh`
  - b. If you need it in debug mode, run `$ hybrisserver.sh debug`
2. The start-up is complete when the log displays a line such as *Server Startup in XXXXms*.
3. Open backoffice <https://localhost:9002/backoffice/>.
4. Navigate to Clouinary > ClouinaryConfiguration and enable the Clouinary connection (for more details refer to the *SAP Commerce Clouinary Extension Integration* document, *Clouinary Configuration in Backoffice* section).
5. From Cronjobs, find and run the “**cloudinaryMediaUploadSyncCronjob**” cronjob in backoffice. Set the catalogVersion before running the job.
6. After completion, find and run the “**cloudinaryMediaTagUpdateCronjob**” cronjob in backoffice. Set catalogVersion for which you want to run the job.
7. Add **127.0.0.1** apparel-uk.local electronics.local in your host file entry.
8. Access apparel site using <https://apparel-uk.local:9002/yacceleratorstorefront/en/>.

**Note** - As a quick validation you can inspect a media from the storefront and verify that the source URL is pointing to Clouinary.

## B2B Accelerator Setup Steps

Follow the steps below if you are installing the extensions on a new system and don't have any existing SAP Commerce setup.

### Download and Extract

To install CX suite on your machine, follow these steps:

For CX Commerce 2005 Local Environment Setup, use the latest 2005 CX Commerce.

This suite contains both CX Commerce and CEP extensions.

1. Obtain the hybris Commerce package
2. Extract the tar package in your chosen location (e.g. */home/user/* and rename it as *clouinary*).

### Clone Repository

To clone the Clouinary repository you need to have GitHub repository access.

Clone the repository to the custom (*{your sap cx base directory}/bin/custom*) folder from [https://github.com/clouinary/clouinary\\_sap\\_commerce.git](https://github.com/clouinary/clouinary_sap_commerce.git) and switch to the development branch.

Ask your admin for access if required.

```
git clone https://github.com/clouinary/clouinary_sap_commerce.git custom
```

### Create Symlink

Change directory to *{your sap cx base directory}/hybris* and execute this command:

```
In -s {your sap cx base directory}/bin/custom/ext-config config
```

For windows:

```
mklink /D [Destination] [Original source]
```



## Deploy the JDBC JAR

In order to install the connector:

1. Download the [mysql-connector-java-5.1.37-bin.jar](#)
2. Open a File Explorer.
3. Copy mysql-connector-java-5.1.37-bin.jar to  
*/home/user/cloudinary/hybris/bin/platform/lib/dbdriver.*

## Create database and user

Connect to your local MySQL installation and create a database (it'll be populated during initialize) and a user:

```
CREATE DATABASE cloudinary CHARACTER SET=utf8 COLLATE=utf8_bin;  
CREATE USER 'cloudinary'@'localhost' IDENTIFIED BY 'cloudinary'; GRANT ALL PRIVILEGES ON cloudinary.* TO 'cloudinary'@'localhost' WITH GRANT OPTION; flush privileges;
```

## Install Recipe

A Clouinary recipe is required for installing and initializing required Clouinary extensions and configuration.

Copy cloudinary\_b2b.zip from *{your sap cx base directory}/recipes* and extract it to your local CX commerce recipes folder

Open a command line and go to the “installer” directory of your SAP Commerce instance.

Execute the following commands to install and initialize the recipe:

- On Windows: `install.bat -r cloudinary_b2b`
- On Linux or Mac: `./install.sh -r cloudinary_b2b`

## Install Required Addon

```
ant addoninstall -Daddonnames="cmsoccaddon"  
-DaddonStorefront.ycommerceservices="cloudinarymediawebservices"
```

Initialize SAP Commerce by entering the following command:

- On Windows: install.bat -r cloudinary\_b2b initialize
- On Linux or Mac: ./install.sh -r cloudinary\_b2b initialize

Follow the steps below if you are installing the extensions on an existing SAP Commerce system.

1. Add the following Clouinary extensions to your localextensions.xml file:

```
<extension name=cmsoccaddon />  
<extension name='mediaconversion' />  
<extension name='mediaconversionbackoffice' />  
<extension name='cloudinarymediacore' />  
<extension name='cloudinarymediabackoffice' />  
<extension name='cloudinarymediasmartedit' />  
<extension name='cloudinarymediawebservices' />  
<extension name='cloudinarymediafacades' />  
<extension name='cloudinarymediab2baddon' />
```

2. Install the required addons below in your platform directory:

```
ant addoninstall -Daddonnames="cmsoccaddon"  
-DaddonStorefront.ycommerceservices="cloudinarymediawebservices"
```

```
ant addoninstall -Daddonnames="cloudinarymediab2baddon"  
-DaddonStorefront.yb2bacceleratorstorefront="<your custom b2b storefront>"
```

3. In the cloudinarymediaaddon project.properties change the property from  
“**yb2bacceleratorstorefront.additionalWebSpringConfigs.cloudinarymediaaddon**”  
to <your custom b2b storefront  
extension>.**additionalWebSpringConfigs.cloudinarymediaaddon**
4. Execute a platform build using “ant clean all”.
5. Perform a SAP Commerce system update.

## Start SAP Commerce Server

To perform this step, we use the Hybris Server, which is a pre-configured application server based on Apache Tomcat.

The hybris server has two modes of operation:

- *Normal*: Set by default for the hybris server.
- *Debug*: Only used with the develop configuration template (the one we chose earlier). The debug mode is slower in performance but is useful for developing using the external debugging tools (for example Eclipse IDE with compatible Java debugger).

To start the server, from the same terminal shell:

1. Navigate to the `/bin/platform` directory and run:
  - a. **`$.hybrisserver.sh`**
  - b. If you need it in debug mode, run **`$.hybrisserver.sh debug`**
2. The start-up is complete when the log displays a line such as *Server Startup in XXXXms*.
3. Open backoffice <https://localhost:9002/backoffice/>.
4. Navigate to Clouinary > ClouinaryConfiguration and enable the Clouinary connection (for more details refer to the *SAP Commerce Clouinary Extension Integration* document, *Clouinary Configuration in Backoffice* section).
5. Find and run the “**cloudinaryMediaUploadSyncCronjob**” cronjob in backoffice. Set the `catalogVersion` before running job.
6. After completion, find and run the “**cloudinaryMediaTagUpdateCronjob**” cronjob in backoffice. Set `catalogVersion` for which you want to run the job.
7. Add **127.0.0.1** `powertools.local` in your host file entry.
8. Access apparel site using <https://powertools.local:9002/yb2bacceleratorstorefront/en/>.

**Note** - As a quick validation you can inspect a media from the storefront and verify that the source URL is pointing to Clouinary.

# Uninstall SAP Commerce B2C Clouinary Extension

## Uninstall Extension

Follow the steps below if you are uninstalling the Clouinary extensions from SAP Commerce system.

1. Remove the following Clouinary extensions from your localextensions.xml file:

```
<extension name=cmsoccaddon />  
<extension name='mediaconversion' />  
<extension name='mediaconversionbackoffice' />  
<extension name='clouinarymediacore' />  
<extension name='clouinarymediabackoffice' />  
<extension name='clouinarymediasmartedit' />  
<extension name='clouinarymediawebservices' />  
<extension name='clouinarymediafacades' />  
<extension name='clouinarymediaaddon' />
```

**Note** - "cmsoccaddon", "mediaconversion" and "mediaconversionbackoffice" are the OOB SAP Commerce extensions. We are enabling this extension as it is required to install Clouinary extensions. However If it is required for an existing project as well then we should only remove Clouinary extensions while uninstalling SAP Commerce B2C Clouinary Extensions.

2. Execute a platform build using "ant clean all".
3. After success build, execute initialization using "ant initialize" to cleanup Clouinary data.

**Note** - As part of Initialization process, the OOB data will be created in System, but if still want to use Clouinary media we should download all the originals from Clouinary and import them in SAP.

## Start SAP Commerce Server

To perform this step, we use the Hybris Server, which is a pre-configured application server based on Apache Tomcat.

The hybris server has two modes of operation:

- *Normal*: Set by default for the hybris server.
- *Debug*: Only used with the develop configuration template (the one we chose earlier). The debug mode is slower in performance but is useful for developing using the external debugging tools (for example Eclipse IDE with compatible Java debugger).

To start the server, from the same terminal shell:

1. Navigate to the `/bin/platform` directory and run:
2. The start-up is complete when the log displays a line such as *Server Startup in XXXXms*.
3. Add **127.0.0.1** apparel-uk.local electronics.local in your host file entry.
4. Access apparel site using <https://apparel-uk.local:9002/yacceleratorstorefront/en/>.

**Note** - As a quick validation you can inspect a media from the storefront and verify that the source URL is pointing to your local system not to the Clouinary.

# Uninstall SAP Commerce B2B Clouinary Extension

## Uninstall Extension

Follow the steps below if you are uninstalling the Clouinary extensions from SAP Commerce system.

1. Remove the following Clouinary extensions from your localextensions.xml file:

```
<extension name=cmsoccaddon />  
<extension name='mediaconversion' />  
<extension name='mediaconversionbackoffice' />  
<extension name='clouinarymediacore' />  
<extension name='clouinarymediabackoffice' />  
<extension name='clouinarymediasmartedit' />  
<extension name='clouinarymediawebservices' />  
<extension name='clouinarymediafacades' />  
<extension name='clouinarymediab2baddon' />
```

**Note** - "cmsoccaddon", "mediaconversion" and "mediaconversionbackoffice" are the OOB SAP Commerce extensions. We are enabling this extension as it is required to install Clouinary extensions. However If it is required for an existing project as well then we should only remove Clouinary extensions while uninstalling SAP Commerce B2B Clouinary Extensions.

2. Execute a platform build using "ant clean all".
3. After success build, execute initialization using "ant initialize" to cleanup Clouinary Data.

**Note** - As part of Initialization process, the OOB data will be created in System, but if still want to use the Clouinary media you should download all the originals from Clouinary and import them in SAP.

## Start SAP Commerce Server

To perform this step, we use the Hybris Server, which is a pre-configured application server based on Apache Tomcat.

The hybris server has two modes of operation:

- *Normal*: Set by default for the hybris server.
- *Debug*: Only used with the develop configuration template (the one we chose earlier). The debug mode is slower in performance but is useful for developing using the external debugging tools (for example Eclipse IDE with compatible Java debugger).

To start the server, from the same terminal shell:

1. Navigate to the `/bin/platform` directory and run:
2. The start-up is complete when the log displays a line such as *Server Startup in XXXXms*.
3. Add **127.0.0.1** powertools.local in your host file entry.
4. Access apparel site using <https://powertools.local:9002/yb2bacceleratorstorefront/en/>.

**Note** - As a quick validation you can inspect a media from the storefront and verify that the source URL is pointing to your local system not to the Clouinary.