



**Hewlett Packard**  
Enterprise

# Cloud Service Automation

Software Version: Content SDK 16.07

## HPE CSA Content SDK Guide

Document Release Date: July 2016  
Software Release Date: July 2016

## Legal Notices

### Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

### Restricted Rights Legend

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notice

© Copyright 2016 Hewlett Packard Enterprise Development LP

### Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

## Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <https://softwaresupport.hpe.com>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: <https://hpp12.passport.hp.com/hppcf/createuser.do>

Or click **Register for HP Passport** link at the top of the HPE Software Support page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

## Support

Visit the HPE Software Support Online web site at: <https://softwaresupport.hpe.com>

This web site provides contact information and details about the products, services, and support that HPE Software offers.

HPE Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HPE support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

**<https://hpp12.passport.hp.com/hppcf/createuser.do>**

To find more information about access levels, go to:

**<https://softwaresupport.hpe.com/web/softwaresupport/access-levels>**

**HPE Software Solutions Now** accesses the HPESW Solution and Integration Portal Web site. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this Web site is **<http://www8.hp.com/us/en/software/enterprise-software.html>**

# Contents

Introduction .....	5
Downloading CSA Content SDK and Content Capsules .....	6
Compatibility Matrix .....	7
Building your First Capsule .....	8
Preparing Capsule Directory Structure .....	8
Exporting the Content .....	9
Generating Capsule Use case Description File .....	9
Updating the Capsule Use case Description File .....	9
Creating Capsule Definition File .....	12
Automatic Document Generation Feature .....	13
Editing the XSLT File .....	13
Capsule Signing Feature .....	15
Installing HPSign Client .....	15
Setting Environment Variable .....	15
Specifying Boolean Sign Values .....	15
Installing your Capsule .....	16
Getting Support .....	17
FAQ .....	18
General .....	18
Organizing Usecase Directory .....	18
Support Matrix .....	19
Capsule Building .....	20
For More Information .....	21
Send Documentation Feedback .....	22

# Introduction

The Hewlett Packard Enterprise (HPE) CSA Content SDK is a collection of tools, documentation, and best practices to build, package and install CSA content in a standard manner. It enables the CSA ecosystem and Content Authors to build content on a large scale.

# Downloading CSA Content SDK and Content Capsules

To download the CSA Content SDK and Content Capsules, visit the below HPE Live Network link:

<https://hpln.hpe.com/product/cloud-service-automation-csa/content>

# Compatibility Matrix

The following table lists the current software component versions that can be used to build CSA content:

## Software Component Versions

Component	Version
Cloud Service Automation	4.1 + (4.70 certified)
Operations Orchestration	10.10 +(10.60 certified)
CSA Content SDK	16.07

# Building your First Capsule

## Preparing Capsule Directory Structure

### Best Practices

1. Create a master directory and name it appropriately, so that it contains all your use cases (treat this as Use Case Depot).
2. Create separate child directories for each of your use case.
3. Browse to the use case directory and create a directory to keep all the Service Designs and Service Offerings.
  - a. If you have dependent sequential Component Palettes, it is mandatory to place them in the same directory as Service Design/Offering.
  - b. If you have multiple Service Designs/Offerings and each have different Sequential Component Palette dependency, then create individual directories for the combination of Service Design/Offering and dependent Component Palette archives.
4. Create a directory for your Operation Orchestrations (OO) content.
5. If you have multiple OO contents, create sub-directories under the OO directory to place all the OO contents separately.
6. Create a directory named "tools", if you have auxiliary file system components, such as additional .jar/JSP/scripts files.
7. Create a directory for Component tools files, if you have Topology designs and there are additional components.

### Examples

```
--Usecase Depot
  -usecase1
    -oo-contents
      +content1
      +content2
    +csa-contents
    +tools
    +componenttools
```

### Examples with contents

```
--Usecase Depot
  -HPE OpenStack
    -oo-contents
      -content1
        openstack-content.jar
      -content2
        openstack-util-content.jar
    -csa-contents
      SERVICE_DESIGN_Multi_tenant-support.zip
```



```
SERVICE_DESIGN_Compute_with_Storage.zip  
SERVICE_DESIGN_Swift_container.zip  
-tools  
  aux_file.jsp  
  aux_jar_file.jar  
  aux_script.py  
-componenttools  
  capability_mapping.properties  
  metainfo.txt  
  OPENSTACK.properties
```

## Exporting the Content

Export your OO and CSA content from your existing environment.

1. Export your OO contents into the appropriate directory/sub-directory under the use case directory. If you have multiple dependent OO contents, it is advised to keep each of them separately in different sub-directories under the use case directory.
2. Export the CSA contents such as your Service Designs and Service Offerings and save them under the csa-contents directory.

## Generating Capsule Use case Description File

As you have all the required contents of your use case available inside your use case directory, you can use the *CapsulePackager.jar* file to auto-generate the use case description file *usecaseInfo.xml*.

1. Navigate to the directory from command or console prompt.
2. Run the following command:

```
java -jar CapsulePackager.jar -g <path_to_usecase_directory>
```

**example:** java -jar CapsulePackager.jar -g "/Usecase Depot/HPE OpenStack"

3. The *usecaseInfo.xml* file gets created and most of the file content will be generated by browsing and evaluating your file system content.

## Updating the Capsule Use case Description File

The tool will generate a use case manifest file that needs to be updated with your own usecase metadata.

1. Open the *usecaseInfo.xml* file.
2. Edit the file according to the following recommendations:

```
<name>The must match the usecase directory name</name>
```

```
<uuid>ae9caa9f-64fe-41eb-b63d-cd0b8e66fb52</uuid>
```

```
<description>A short description about the usecase</description>
```

```
<publisher>provide the publisher name with email contact information</publisher>
```

```
<csaData>
<content type="sequence" srcLocation="csa-contents">
<name> SERVICE_DESIGN_Multi_tenant-support.zip </name>
<displayName>provide a display name for this SD</displayName>
<description>provide a short description about the SD</description>
<version>1.0.0</version> <!-- Mandatory: Enter the version of the content in x.x.x
format, where x is a whole number -->
    </content>
</csaData>
<ooData>
    <name> openstack-content.jar </name>
    <location>oo-contents/content1</location>
    <displayName>provide a display name for this content</displayName>
    <description>provide a short description of the content </description>
    <dependencies>
        <artifact type="SELF" name="openstack-util-content.jar" location="oo-
contents/content2" version="14.12.1211"/>
        <artifact type="EXTERNAL" name="Base" location="" version="1.4.4" />
        <artifact type="EXTERNAL" name="HP Solutions" location="" version="1.1.2" />
        <artifact type="EXTERNAL" name="Cloud" location="" version="1.1.0" />
        <artifact type="EXTERNAL" name="Virtualization" location="" version="1.1.1" />
Provide any other EXTERNAL dependencies for the usecase
    </dependencies>
</ooData>
<componentTool srcLocation="componenttools">
    <components>
        <component file="metainfo.txt" mappingFile="OPENSTACK.properties"
capabilityMapping="true" />
    </components>
</componentTool>
<auxiliaryData>
    <file name="aux_file.jsp" destinationPath="CSA_JBOSS_
HOME/standalone/deployments/csa.war/propertySources" executable="false"/>
    <file name="aux_jar_file.jar" destinationPath="CSA_JBOSS_
HOME/standalone/deployments/csa.war/propertySources" executable="false"/>
    <file name="aux_script.py" destinationPath="CSA_JBOSS_
HOME/standalone/deployments/csa.war/propertySources" executable="false"/>
Provide the desired target path under "destinationPath"
</auxiliaryData>

<provider displayName="HPE Cloud Service Automation" mandatory="true" name="com.hp.csa">
    <version comparision="equals" value="4.50"/>
<!-- "value" is mandatory and must be in x.x format, where x is a whole number-->
    </provider>
    <provider displayName="HPE Helion Codar" mandatory="true" name="com.hp.codar">
        <version comparision="equals" value="1.50"/>
    </provider>
    <provider displayName="HPE Operations Orchestration" mandatory="true"
name="com.hp.oo">
        <version comparision="equals" value="10.21"/>

```

```
    </provider>  
    Provide appropriate target versions of the supported HPE CSA/Codar/00 platforms  
</supportMatrix>
```

The above mentioned usecaseInfo.xml file is for Topology design. If the use case has Sequential design, you might need to update the ooengine information inside the <processDefinitionTool> tag. For example, below is the description for Amazon EC2 Content:

```
<processDefinitionTool>  
  <ooengines>  
    <ooengine>  
      <folder path="/Library/CSA Content Pack/CSA3.2/Providers/Infrastructure/Amazon  
EC2/Actions" recursive="true" update="true"/>  
      <folder path="/Library/CSA Content Pack/CSA3.2/Providers/Infrastructure/Amazon  
EC2/Load Balancing/Actions" recursive="true" update="true"/>  
      <folder path="/Library/CSA Content Pack/CSA3.2/Providers/Infrastructure/Amazon  
EC2/Validation" recursive="true" update="true"/>  
      <folder path="/Library/CSA Content Pack/CSA3.2/Providers/Email  
Notification/Actions" recursive="true" update="true"/>  
    </ooengine>  
    <ooengine accessPointType="RESOURCE_POOL_SYNC" name="00-RESOURCE-POOL-SYNC" >  
      <folder path="/Library/CSA Content Pack/CSA3.2/Providers/Infrastructure/Amazon  
EC2/Resource Pool Sync/Actions" recursive="true" update="true" />  
    </ooengine>  
  </ooengine>  
</ooengines>  
</processDefinitionTool>
```

**Note:** Once the UUID as under <uuid> element is generated for an use case, it is recommended to maintain the same for all the future versions.

# Creating Capsule Definition File

A Capsule definition file is required to build a Capsule with the SDK packager tool. This file describes high-level metadata that can be displayed while importing and displaying Capsule level information.

To create capsule definition file:

1. Find the sample `capsule.xml` file available under `/xsd` directory of the Content SDK.
2. Copy the file to your content depot directory.
3. Enter the required details within appropriate tags by referring the below **Capsule File Example**:

## Capsule File Example

```
<?xml version="1.0" encoding="utf-8"?>
<buildCapsule sign=true/false>
  <usecases>
    <usecase name="HPE OpenStack" path="/Usecase Depot/" />
  </usecases>
  <capsuleName>HPE-OpenStack.zip</capsuleName>
  <outputDir></outputDir>
  <xsltFilePath></xsltFilePath>
  <signProjectName></signProjectName>
</buildCapsule>
```

- Name of the use case: `<usecase name="HPE OpenStack"` (This is use case directory name).
- Path of the use case: `path="/Usecase Depot/" />` (This is the path to the parent directory of the use case, which is Use case Depot)
- Name of the output capsule name: `<capsuleName>HPE-OpenStack.zip</capsuleName>`
- Location of the output directory name: `<outputDir></outputDir>`
- Complete path of the XSLT file (Optional): `<xsltFilePath></xsltFilePath>`
  - You can ignore providing the path of the XSLT file, if you do not wish to customize the output of the document as per your requirements.
  - If you wish to customize the output document by editing the XSLT file, see ["Automatic Document Generation Feature" on page 13](#).

**Note:** You must complete editing the XSLT file before moving to Step 4.

- Provide the Project name, which has the LDAP group configured for HPE Signing:  
`<signProjectName></signProjectName>`  
For more information on HPE Signing, see ["Capsule Signing Feature" on page 15](#).
4. Open the command prompt and execute the following command to generate the capsule:

```
java -jar CapsulePackager.jar -i capsule.xml
```

# Automatic Document Generation Feature

As part of the Capsule Packager of Content utilities, Automatic Document Generation feature is now provided for generating a brief document about each Content Capsule. This utility provides you with details about your chosen capsule before its installation. The packager tool uses content-as-a-source to generate a file name `usecaseInfo.xml`, as mentioned in ["Creating Capsule Definition File" on page 12](#) and uses the tags defined in this file to create a PDF document for your reference.

You can also customize this document by editing the XSLT file (Style sheet for XML Document). Before you begin to customize the document to suit your preference, you *must* know the following:

- Find the `CsaCapsule.xsl` and `whiteLabel.xsl` (without HPE branding) files available under `/xsd` directory of the Content SDK.
- Provide the complete path to the XSLT file within the `<xsltFilePath></xsltFilePath>` tag of `capsule.xml` file. For more information on XML file, see **Capsule File Example** from ["Creating Capsule Definition File" on page 12](#).
- Two types of editable XSLT files are provided:
  - a. **XSLT file with HPE Template:** This template can be used, if you wish to generate a document having HPE branding elements like logo and border. Locate the `logo.png` file available under `/xsd` directory of the Content SDK, and then provide the path to the logo image as mentioned in the image. See, ["Editing the XSLT File" below](#).
  - b. **XSLT file without HPE Template (White-labeled):** This template can be used, if you wish to generate a document without having HPE branding elements like logo and border or to generate a document having your own logo.

**Note:** For generating a document using your own logo, keep the size of the logo image to 200\*82 px (pixels) and provide its correct path as mentioned in the image below.

Before running the packager, you can edit the XSLT files as per the comments mentioned in the image. See, ["Editing the XSLT File" below](#).

- By default, the generated document is placed inside the Capsule ZIP file as part of Packager.
- The auto generated document can be seen either by unzipping the Capsule file or by entering the following commands at the command prompt using Installer jar file.

```
java -jar "InstallerjarName" "capsule_name.zip" " -d"
```

or

```
java -jar "InstallerjarName" "capsule_name.zip" " -doc"
```

## Editing the XSLT File

To edit the XSLT file:

- a. Open the XSLT file
- b. Edit the XSLT file as shown in the below image:

```

<fo:flow flow-name="xsl-region-body">
  <!-- MODIFIABLE_Remove the border-color,border-style,border-width here
  if you want no Green HPE border on the first page of the document or just
  change the border-color attribute to black color (color code is #000000) -->
  <fo:block-container break-after="page" height="9.7in"
  width="100%" border-color="#02B294" border-style="solid"
  border-width="0.1in" padding="0.1in" text-align="left" font-size="28pt"
  font-family="Helvetica, Arial, sans-serif">
    <fo:block>
      <!--MODIFIABLE if needed-Here is the logo image inclusion,which can
      be changed if required by the User, but keep the size of the image as 200
      X 82px.The logo location needs to be placed in -->
      <fo:external-graphic content-height="2cm"
      scaling="uniform" src="logo.png" />
      <fo:block>#{160}</fo:block>
      <fo:block space-after="1.2in" space-before="0.5in"
      font-weight="bold">
        Cloud Service Automation
        <fo:block font-size="14pt" font-weight="normal">
          Software Version : Capsule Content
          <xsl:value-of
            select="usecaseInfo/ooData/dependencies/artifact/@version" />
        </fo:block>
      </fo:block>
    </fo:block>
  </fo:flow>

```

You can remove the settings as advised in MODIFIABLE

"logo.png": Give a path of your own logo image here.

c. Save the XSLT file.

# Capsule Signing Feature

Capsule packager now comes with the integrated HP Code Signing Service (HPCSS) feature. HPCSS signing is the central Code Signing Service established to meet the requirements of all users having access to HPELN to centrally sign their software and firmware components. It is a Java-based client which communicates to the Code Signing Server over Secure Sockets Layer (SSL).

For successful signing, you should:

- Install HPSign Client. See ["Installing HPSign Client" below](#)
- Set Environment Variable. See ["Setting Environment Variable" below](#)
- Specify Boolean Sign Values. See [Specifying Boolean Sign Values](#)

## Installing HPSign Client

- To Install HPSign Client:
- Go to <https://hpcssweb-pro.austin.hp.com/hpcssui/ViewProject.xhtml> to download and install the client.

**Note:** You need to secure an approval from HPE manager to access the web site.

- Follow the instructions during the installation and verify if all the prerequisites are met.

## Setting Environment Variable

- Make sure the HPCSS path is present in the environment variable. If not, add it to the Class path.

## Specifying Boolean Sign Values

- Specify the Boolean sign values at the root level of the *capsule.xml* file. For more information on XML file, see **Capsule File Example** of ["Creating Capsule Definition File" on page 12](#).

For example, `<buildCapsule sign=true/false>`

- If Signing is set to true, as in, `<buildCapsule sign=true>`  
You must provide the Project name, which has the LDAP group already configured within `<signProjectName></signProjectName>` tag. On successful acceptance of the Project name, the Capsule Packager will move ahead and sign the final capsule.
- If Signing is set to false, as in, `<buildCapsule sign=false>`  
Capsule signing operation is not performed.

If all of the prerequisites are met, the Capsule will be signed and will be available at the output directory.

## Installing your Capsule

Once your capsule is generated, it is a good practice to test its installation in your CSA environment for a quick dry run.

1. Double-click on the `CapsuleInstaller.jar` file or run the below command at the command prompt to launch the installer:

```
java -jar CapsuleInstaller.jar
```

For more information, refer the "[Content Installation Guide.pdf](#)" available under `/docs` directory of the CSA Content SDK to work with the Content Installer.



# Getting Support

To post your queries and get support on the CSA Content SDK, go to the [Cloud Service Automation Practitioners Forum](#).

# FAQ

## General

### **Q. What is this SDK about?**

A: CSA Content SDK allows you to package your CSA/OO content in a standard and portable manner allowing you to move Content around from one environment to another.

It is also the preferred format and packaging for Content that will be pushed to HPELN for customers, partners, and internal teams.

### **Q. What is the SDK compatibility matrix?**

A. SDK is compatible with both CSA 4.5 and 4.2 releases.

### **Q. What are the capsules available so far?**

CSL Capsules are being regenerated with SDK packager via automated build process. You can download the SDK from the below HPENL Node:

<https://hpln.hpe.com/product/cloud-service-automation-csa/content>

We are looking forward to your capsules.

### **Q. Are we looking at a new installer to deploy the CSA capsules?**

The SDK installer is same as the CSA Content installer available in the current CSA software distribution.

### **Q. What content types do you support?**

Content of different types can be embedded into the capsule including OO Content Pack, Providers, Components, Service Designs (Sequence or Topology based), and Service Offering.

### **Q: Where do I go for help during the Beta?**

Visit the [HPELN](#) Web site and the [CSA Content SDK Forum](#)

## Organizing Usecase Directory

### **Q: Can I keep the auxiliary files randomly inside the use-case directory?**

A: No, all the auxiliary files need to be kept in a sub-directory named "tools" inside the use-case directory.

### **Q: What are the PATH variables supported for "destinationPath" of "auxiliaryData" section?**

A: CSA\_HOME and CSA\_JBOSS\_HOME are supported here. The variables must not be pre/post-fixed by any special characters like % or \$

**Q: How can I package my dependent sequential Component Palette along with the Service Design?**

A: Keep the dependent component palette archive along with the service design in the same directory. If you have multiple Service Designs and each have distinct component palette dependency, then create multiple service designs directories and store the combination of archives in the respective directories. If all the service designs depend on the same set of component palettes, then a single service design directory is sufficient to store the files.

**Q: How do I instruct the CapsuleInstaller to create a new Provider Type for my capsule?**

A: If your capsule needs any specific provider type, which might not be present in the target CSA environment, you can instruct as follows in your usecaseInfo.xml file:

```
<csaData>
  <content type="sequence" srcLocation="service-designs">
    <name>...
    ...
    <dependencies>
      <providerTypes>
        <providerType>
          <displayName>MY PROVIDER TYPE</displayName>
          <description>A short description about the provider type</description>
        </providerType>
      </providerTypes>
    </dependencies>
  </content>
  ...
</csaData>
```

## Support Matrix

**Q: What version of CSA/Codar and OO should be mentioned in the support matrix section?**

A: It is recommended to provide the versions of HPE CSA/Codar and HPE OO Studio used to create the respective contents

**Q: Can my capsule, with support matrix of CSA 4.2 be installed in CSA 4.5 or higher environment?**

A: Yes, with installer version 4.00, it can be (with the following limitations). Capsule with support matrix CSA 4.x can be installed in CSA 4.y environment, where y >= x. The same is applicable for HPE Codar and HPE OO support matrix as well.

## Capsule Building

**Q: While creating capsule, what should be done if it fails with an error message "Error while validating the UsecaseInfo document"**

A: The usecaseInfo.xml file is not as per its schema definition. Please confirm that the usecase definition file (usecaseInfo.xml) is according to the usecaseInfo.xsd specifications

**Q: How can I assign a version number to the capsule?**

A: Use "-bv" switch to pass capsule version dynamically to the CapsulePackager. e.g. "java -jar CapsulePackager.jar -i capsule.xml -bv 15.09". Default value if not provided is YY.MM

**Q: How can I assign a build number to the capsule?**

A: Use "-bn" switch to pass capsule build number dynamically to the CapsulePackager. e.g. "java -jar CapsulePackager.jar -i capsule.xml -bn 123". Default value if not provided is 1

**Q: How can I assign a source control revision number to the capsule?**

A: Use "-br" switch to pass source control revision number dynamically to the CapsulePackager. e.g. "java -jar CapsulePackager.jar -i capsule.xml -br 4567". Default value if not provided is 0

## For More Information

HPE software product documentation can be found at <https://softwaresupport.hpe.com/>

On the HPE Software Support portal, select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, white papers, and other information sources.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

# Send Documentation Feedback

If you have comments about this document, you can contact the [documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on HPE CSA Content SDK Guide (Cloud Service Automation Content SDK 16.07)**

Just add your feedback to the email and click send.

If no email client is available, copy the information to a new message in a web mail client, and send your feedback to [csl\\_doc\\_feedback@hpe.com](mailto:csl_doc_feedback@hpe.com).

We appreciate your feedback!