Table of Contents

[1 CommonContent 1](#_Toc773965)

[2 Accessing and working with the ContentContent 1](#_Toc773966)

[3 Using conditional text in CommonContent 3](#_Toc773967)

[4 Troubleshooting 3](#_Toc773968)

[4.1 CommonContent changes not appearing in infocenter 3](#_Toc773969)

[5 Promoting the CommonContent build 3](#_Toc773970)

[6 Branching CommonContent Field Care files not required 3](#_Toc773971)

[7 Using CommonContent in Field Care 19 snapshot builds 3](#_Toc773972)

[8 Using CommonContent in Field Care 19 release builds 3](#_Toc773973)

[9 Other CommonContent Field Care files 3](#_Toc773974)

# CommonContent

CommonContent provides a mechanism for sharing information between books in different projects. The original source file will be located within the SharedContent folder in the CommonContent. Books that require the common content source file will include the source file in their book files.

The Field Care svfnac and svffttx projects each have a csc.guide that includes information stored in CommonContent. The book file of each csc.guide includes files from CommonContent.

This document describes how to use the CommonContent and also how CommonContent is used in Field Care.

# Accessing and working with the ContentContent

1. Clone the CommonContent folder from Git to your local repository.

<https://gerrit.ext.net.nokia.com/gerrit/#/admin/projects/CXSPUBS/CommonContent>

1. Optional: If you have already cloned the CommonContent folder, do a Git pull in CommonContent to use the latest Git files.
2. In the pom.xml files of the svfnac and svffttx projects, ensure <commoncontent.version> contains master-SNAPSHOT.
3. Determine which files are being included from CommonContent by examining the book file of the csc.guide in the svfnac or svffttx project. For example, this snippet from the book file of the svfnac csc.guide (that is, csc.guide.xml) shows how one of the CommonContent files is included:

<xi:include href="../../../doctools/SharedContent/en\_US/FixedAccessCare-FieldCare/csc.guide/p1\_ch\_040\_Run\_Diagnostics.xml"/>

To explain how this path works:

1. The local and server builds both work in a similar manner, copying files to be built into the target work directory.
2. The CommonContent files, which are used for both local and server builds, are copied into the doctools folder in the target directory.
3. When the books in the target work directory are built, the xi:include above specifies the path to the Field Care xml file in the target\doctools\SharedContent folder.
4. To locate a particular file, navigate to:

CommonContent -> SharedContent -> en\_US-> FixedAccessCare-FieldCare -> csc.guide

1. Make the changes in the xml files of the csc.guide in CommonContent. Note the following guidelines:
2. Place images in the **images** folder of the csc.guide.
3. If you need a particular change such as an image or text to be different between the csc.guide books in the different projects, you must use **conditional text** as described in [Using conditional text in CommonContent](CommonContent.docx).
4. Save all the changes, but do not commit them yet.
5. Test the CommonContent changes:
6. Run “mvn install” in your CommonContent directory. This step is what causes the next step to use your local CommonContent changes.
7. Run “mvn clean compile” in the svfnac or svffttx projects to build the docs.  
     
   You can change your module.xml file to build only the csc.guide to reduce build time.
8. Verify your changes in the built doc.
9. If more CommonContent changes are needed, repeat these steps.

Also see [Updating your local glossary and doctools](http://pubs.motive.com/info/topic/toolsguide/section_vxj_aqv_fh.html) for more information.

1. Once you have verified the changes work in the local build, do a Git commit/pushto push all the CommonContent changes.   
   This causes the server build of CommonContent to start automatically.
2. :

<https://build4.lab.pl.alcatel-lucent.com/job/OSS/job/TechPubs/job/tools/job/CommonContent/>

1. Do server builds of the svfnac and svffttx projects.
2. Verify the changes appear in the svfnac (DSLST) and svffttx (FTTxST) files in the infocenter.

# Using conditional text in CommonContent

If you need a change to be different between the csc.guides of svfnac and svfftx, then use conditional text. If the change is identical in each project, you need **NOT** to use conditional text. The steps mentioned below describe how to update the csc.guide in CommonContent with conditional text.

Conditional text

With conditional text, you can create a single XML document to produce different generated documents. To mark text for inclusion or exclusion, use an appropriate attribute. The system supports author\_condition\_1 and author\_condition\_2 properties in your build.xml file to conditionalize text. These properties are used to indicate what elements to include or exclude. To conditionalize content within a paragraph, use these attributes with the <phrase> element in the <para> element. The <phrase> element has no effect on formatting.

**Note: In the Field Care 18 SP2 release for csc.guide, author\_condition\_2 has been used.**

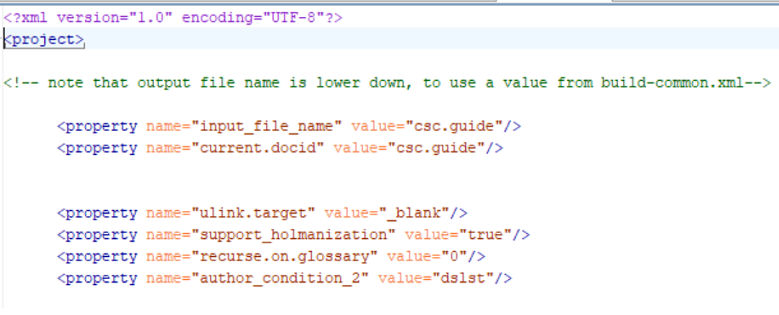
In the example shown below:

1. The part\_010\_chapter\_021\_CSC\_Dashboard.xml file located within the csc.guide shows how to use the author\_condition\_2 attribute to produce svfnac and svffttx specific content using conditional text.
2. The links to the new or updated images mentioned in the example are located within the images directory of the CommonContent csc.guide folder.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?><chapter xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://docbook.org/ns/docbook"  xmlns:xi="http://www.w3.org/2001/XInclude" version="5.0-extension BroadBook 2.0"  xml:id="CSC\_dashboard">  <title>Dashboard</title>  <info>  </info>  <indexterm>  <primary>DSL Quality of Service</primary>  </indexterm>  <para>The CSC Dashboard is a page that provides you the current status of @@ProductName@@ at a glance.</para> <para>The following figure shows the CSC Dashboard**:**  **<informalfigure author\_condition\_2="dslst">**  <mediaobject>  <imageobject>  <imagedata fileref="images/dslcsc.png"/>  </imageobject>  </mediaobject>  </informalfigure>  **<informalfigure author\_condition\_2="fttx">**  <mediaobject>  <imageobject>  <imagedata fileref="images/cscdashboard.png"/>  </imageobject>  </mediaobject>  </informalfigure></para> </chapter> |
| As mentioned above, any new images that you add to the CommonContent csc.guide need to be saved within the **images** folder located in the csc.guide. |

* Add the author\_condition\_2 property in your projects (svfnac and svfftx) csc.guide’s build.xml file as shown below. The values should match the ones mentioned in the CommonContent files for author\_condition\_2. Note that dslst is used to indicate the svfnac project and fttx to indicate the svffttx project.

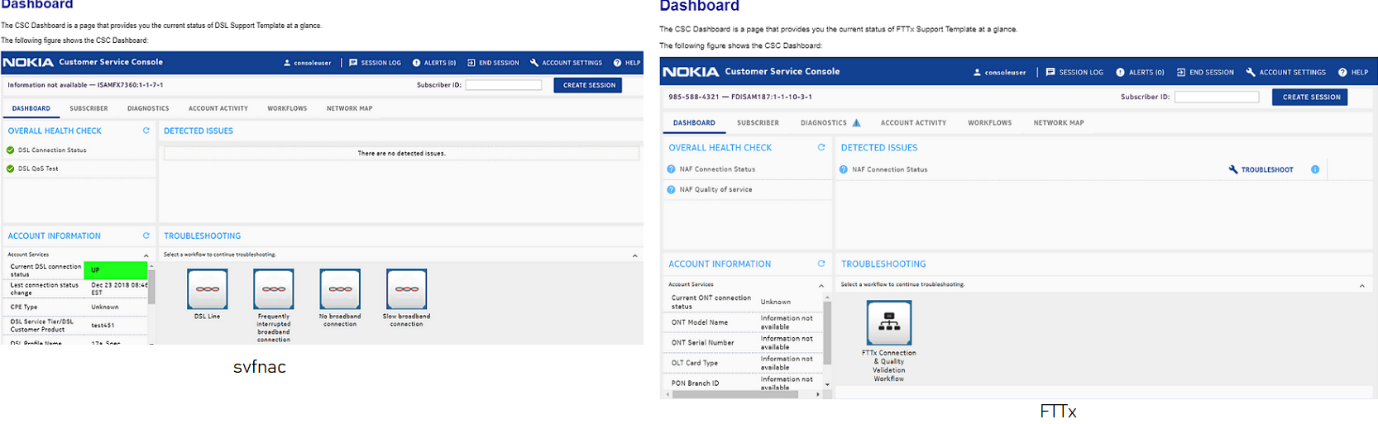
For svfnac:



For svffttx:



* At build time, the system looks at the build.xml file for the <property name="author\_condition\_2" value="xxx"/>, where xxx is either dslst or fttx. If the system finds the property, it examines elements in your XML for author\_condition\_2 attributes. If an element has the author\_condition\_2 attribute with the string “xxx” in its value, the system keeps the element and its contents.
* After a successful Jenkins build, the dashboard images for svfnac and svffttx will appear as follows in the infocenter.



# Troubleshooting

## CommonContent changes not appearing in infocenter

* The CommonContent version must be properly specified in the pom.xml file of the svcnac or svffttx projects. For example, if you specify a released version of CommonContent in the pom.xml file, then changes that later appear in the snapshot version of CommonContent will not be picked up.
* If you need a new released version of CommonContent, contact Scott Cox. But, if Scott is unavailable, you can promote the build as described in [Promoting the CommonContent build](CommonContent.docx).
* Note down the latest version number of the CommonContent and update your project's pom.xml file (<commoncontent.version>**xxx**</commoncontent.version>, where xxx is the latest version number) before the release date.
* Run the project locally and on the server. Then, check the infocenter to verify if the changes are appearing.

# Promoting the CommonContent build

NOTE: You only need to do this if Scott Cox is not available. Otherwise, ask Scott.

1. When you have a successful Jenkins build of CommonContent, promote the latest build (by clicking on the icon as shown in the figure).



1. Verify that the promote worked.
2. Visit the following site and locate the build you promoted:

<https://repo.lab.pl.alcatel-lucent.com/techpubs-mvn-releases/motive/docmodules/glossary/>

1. If the build is not present, promote it again.

# Branching CommonContent Field Care files not required

Field Care files in CommonContent do not need to be branched.

# Using CommonContent in Field Care 19 snapshot builds

For the Field Care 19 release, the CommonContent should be set to the following value until the Field Care release builds are made:

  <properties>

           <doctools.version>master-SNAPSHOT</doctools.version>

          <commoncontent.version>master-SNAPSHOT</commoncontent.version>

          <libdoctools.version>master-SNAPSHOT</libdoctools.version>

  </properties>

# Using CommonContent in Field Care 19 release builds

For Field Care 19 release builds, you must request a released version of CommonContent from the doctools team ([mpd-techpubs-buildmgr@list.nokia.com](mailto:mpd-techpubs-buildmgr@list.nokia.com)) if shared content has been added or updated since the [last CommonContent release](http://pubs.motive.com/wiki/ReleaseVersions#CommonContent.28Glossary.29versions).

When you make the Field Care release builds, specify the release version of CommonContent in the pom.xml file, such as:

<properties>

<doctools.version>2.1.9-152</doctools.version>

<commoncontent.version>2.1.5-121</commoncontent.version>

<libdoctools.version>2.1.0-13</libdoctools.version>

</properties>

# Other CommonContent Field Care files

The CommonContent -> SharedContent -> en\_US-> FixedAccessCare-FieldCare folder contains the following directories:

* csc.guide: described in the sections above.
* customizing\_SOC

The customizing\_SOC folder contains the p3\_ch\_060\_customizing\_SOC.xml file and its associated images folder. It appears that the p3\_ch\_060\_customizing\_SOC.xml file is only called from the following file, and this book is not called from the module.xml file:

fttx\_template\_deploy\_smp61/fttx\_template\_deploy\_smp61.xml

In other words, this file is not currently being used.