

CMS Menus How-To

Table of contents

- 1 Why care about CMS Menus..... 2
- 2 Lenya Core Documentation..... 2
- 3 Introducing a new menu option.....2
- 4 Edit generic.xsp to add the item for the menu option..... 3
- 5 Double-check usecase and workflow configuration.....4

The CMS Menus are the menu options in Lenya that are available only in authoring mode and that are used by the editor, reviewer or admin to trigger operations such as editing, deleting or publishing a page and the like. CMS Menus are completely different and have nothing in common with the navigational menus in your publication!

In other words: The site navigation is what the readers in the Live view will see, while the CMS Menus are only visible in the authoring area.

1 Why care about CMS Menus

Most of the time the CMS Menus can be seen as a part of the Lenya application and should just work. But sometimes it will be necessary to introduce additional menu options if your publication has any special operations that need to be accessible through a menu option. Sometimes you might also find that a menu option is grayed out when you think it should be accessible. In that case understanding the anatomy of the CMS menus will help you in debugging the problem.

2 Lenya Core Documentation

There is some information on the CMS Menus in the official Lenya documentation, but it's hidden in a place where you might not expect it. So here's a link: [Menubar documentation](http://../docs/1_2_x/components/layout/lenya-menubar.html) (../docs/1_2_x/components/layout/lenya-menubar.html)

Make sure you read that before continuing reading this page here!

Unfortunately the picture on the page mentioned above is a bit misleading. You might think that "Usecases" and "Workflow" steps in red in the right part of the picture refer to XSLT stylesheets which are applied using the XSLT transformer. This is not the case! They are own transformers implemented directly in the Lenya Java layer:

```
<map:transformer name="workflowmenu" logger="lenya.sitemap.transformer.workflowmenu"
src="org.apache.lenya.cms.cocoon.transformation.WorkflowMenuTransformer" />
<map:transformer name="usecasemenu" logger="lenya.sitemap.transformer.usecasemenu"
src="org.apache.lenya.cms.cocoon.transformation.UsecaseMenuTransformer" />
```

Nevertheless they do exactly what is described in the picture: They disable menu options if they are not accessible because

- executing the usecase requires a role which the user belonging to the current session is lacking or
- the current workflow state of the document forbids the usecase

3 Introducing a new menu option

First of all, before you start adding a new menu option to the CMS menus, ask yourself what usecase should be linked to that menu option and if the usecase exists already. If you find out that you need a new usecase as well, it is up to you if you first implement the new usecase and test it manually by invoking the appropriate URIs and then add the menu option or do it vice versa. It is just important that you understand that it will take both: The usecase and the menu option to enable new functionality for the user.

According to the page on the core documentation mentioned above, the CMS menus are defined in an XML document like this:

```

<menu
  xmlns="http://apache.org/cocoon/lenya/menubar/1.0"
  xmlns:uc="http://apache.org/cocoon/lenya/usecase/1.0"
  xmlns:wf="http://apache.org/cocoon/lenya/workflow/1.0">
  ...
  <item wf:event="edit"
    uc:usecase="edit-forms"
    uc:step="open" href="?...">Edit with Forms</item>
  ...
</menu>

```

4 Edit generic.xsp to add the item for the menu option

In practice this XML is generated of an XSP which you can find in *lenya/pubs/<your-pub>/config/menus/generic.xsp*

Edit this XSP to add your new menu item!

The XSP is used in the following section of global-sitemap.xmap:

```

<map:match pattern="lenya-page/*/*/*" >
  <map:generate src="cocoon:/menu-xml/{1}/{2}/{3}"/>

  <map:call resource="i18n"/>

  <map:act type="resource-exists" src="lenya/pubs/{1}/config/workflow/" >
    <map:transform type="workflowmenu"/>
  </map:act>

  <map:act type="resource-exists" src="lenya/pubs/{1}/config/ac/usecase-policies.xml" >
    <map:transform type="usecasemenu"/>
  </map:act>

  <map:transform src="lenya/xslt/menu/menu2xhtml.xsl" >

    <map:parameter name="contextprefix" value="{request:contextPath}"/>
    <map:parameter name="publicationid" value="{1}"/>
    <map:parameter name="completearea" value="{2}"/>
    <map:parameter name="documentarea" value="{page-envelope:area}"/>
    <map:parameter name="documenturl" value="{page-envelope:document-url}"/>
    <map:parameter name="documentid" value="{page-envelope:document-id}"/>

    <map:parameter name="userid" value="{access-control:user-id}"/>
    <map:parameter name="servertime" value="{date-i18n:currentDate}"/>
    <map:parameter name="workflowstate" value="{workflow:state}"/>
    <map:parameter name="islive" value="{workflow:variable.is_live}"/>
  </map:transform>

  <map:call resource="i18n"/>

  <map:transform src="lenya/xslt/menu/menu2xslt.xsl" >
    <map:parameter name="contextprefix" value="{request:contextPath}"/>
    <map:parameter name="publicationid" value="{1}"/>
    <map:parameter name="area" value="{2}"/>
    <map:parameter name="documenturl" value="{page-envelope:document-url}"/>
  </map:transform>

```

```
<map:serialize type="xml"/>
</map:match>
```

The `<map:generate src="cocoon:/menu-xml/{1}/{2}/{3}"/>` is an internal pipeline, which can be found in the same file:

```
<!-- menu-xml/{publication-id}/... -->

<map:match pattern="menu-xml/*/**" internal-only="true">
  <map:mount uri-prefix="menu-xml/{1}/" src="lenya/pubs/{1}/menus.xmap" check-reload="true"
  reload-method="synchron"/>
</map:match>
```

This mounts the `lenya/pubs/<your-pub>/menus.xmap` sitemap snippet from the publication which then will decide whether we are in live or authoring mode and generate the menu definition from the publication specific `menu.xsp`:

```
<map:match pattern="live/**">
  <map:generate type="serverpages" src="../../content/menus/live.xsp"/>
  <map:serialize type="xml"/>
</map:match>

<map:match pattern="**">

  <map:generate type="serverpages" src="config/menus/generic.xsp"/>
  <map:serialize type="xml"/>
</map:match>

</map:pipeline>
```

5 Double-check usecase and workflow configuration

If you look back at the `lenya-page/*/**` matcher in `global-sitemap.xmap` above, the menus are built by:

- generating the menu definition from the XSP (which we edited already)
- applying the `workflowmenu` and `usecasemenu` transformers
- using `lenya/xslt/menu/menu2xhtml.xsl` to generate the XHTML

This means if you added your new menu option to the XSP it should show up in the menu. But it might be grayed out.

If this is the case, either the `workflowmenu` or the `usecasemenu` transformer decided based on the workflow configuration or based on the usecase policy that this option is not available to this user at this point in time. You may want to check the appropriate configurations to check why this is the case.

The usecase configuration can be found in `lenya/pubs/<your-pub>/config/ac/usecase-policies.xml`