

# The Lenya Menubar

## Table of contents

1 Defining the Menubar.....	2
1.1 The XML Structure.....	2
1.2 Showing Blocks and Items Only in Certain Tabs.....	2
1.3 Specifying Usecases and Workflow Events.....	2
2 Applying the Menubar - Meta Stylesheet Approach.....	3
2.1 The Big Picture (literally).....	3
2.1.1 Creating the Menu XHTML.....	4
2.1.2 Requesting and Applying the Menubar.....	5
2.2 Configuring the Publication.....	5
2.2.1 Requesting the lenya-page Meta Stylesheet.....	5
2.2.2 Creating the Menu XML.....	5
2.2.3 Supplying Menus in menus.xmap.....	5
2.3 Creating a Customized Menu.....	6
3 Applying the Menubar - Classic Approach.....	6
3.1 XSLT.....	6

## 1. Defining the Menubar

### 1.1. The XML Structure

A menubar declaration contains the definition of several menus. Each menu consists of a set of blocks containing items. The structure of a menu declaration looks as follows:

```
<menu xmlns="http://apache.org/cocoon/lenya/menubar/1.0">

  <menus>

    <menu label="File" name="File">
      <block>
        <item href="...">New Document</item>
        <item href="...">New Language Version</item>
        ...
      </block>
      ...
    </menu>

    ...

  </menus>

</menu>
```

### 1.2. Showing Blocks and Items Only in Certain Tabs

When you use the same menu declaration for different tabs, it might be wished that a menu item shouldn't appear in all these tabs. For instance, the *Edit* menu item should be displayed only in the *authoring* tab menu.

To disable a menu item or block, you add an attribute with the name of the tab (*authoring*, *info*) and the value *false*:

```
<item info="false">...</item>

<block authoring="false">
  ...
</block>
```

### 1.3. Specifying Usecases and Workflow Events

When a menu item calls a usecase or triggers a workflow transition, and the corresponding action is not allowed, the *href* attribute of the item is removed before the menu is rendered to disable the item. For this purpose, the *<item>* element supports the attributes

- *wf:event*,
- *uc:usecase*, and
- *uc:step*.

If the attributes are provided, they are translated into the appropriate request parameters:

- *wf:event* -> *lenya.event*
- *uc:usecase* -> *lenya.usecase*

- `uc:step -> lenya.step`

The attributes are used in the following way:

```
<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>
```

The resulting request would be

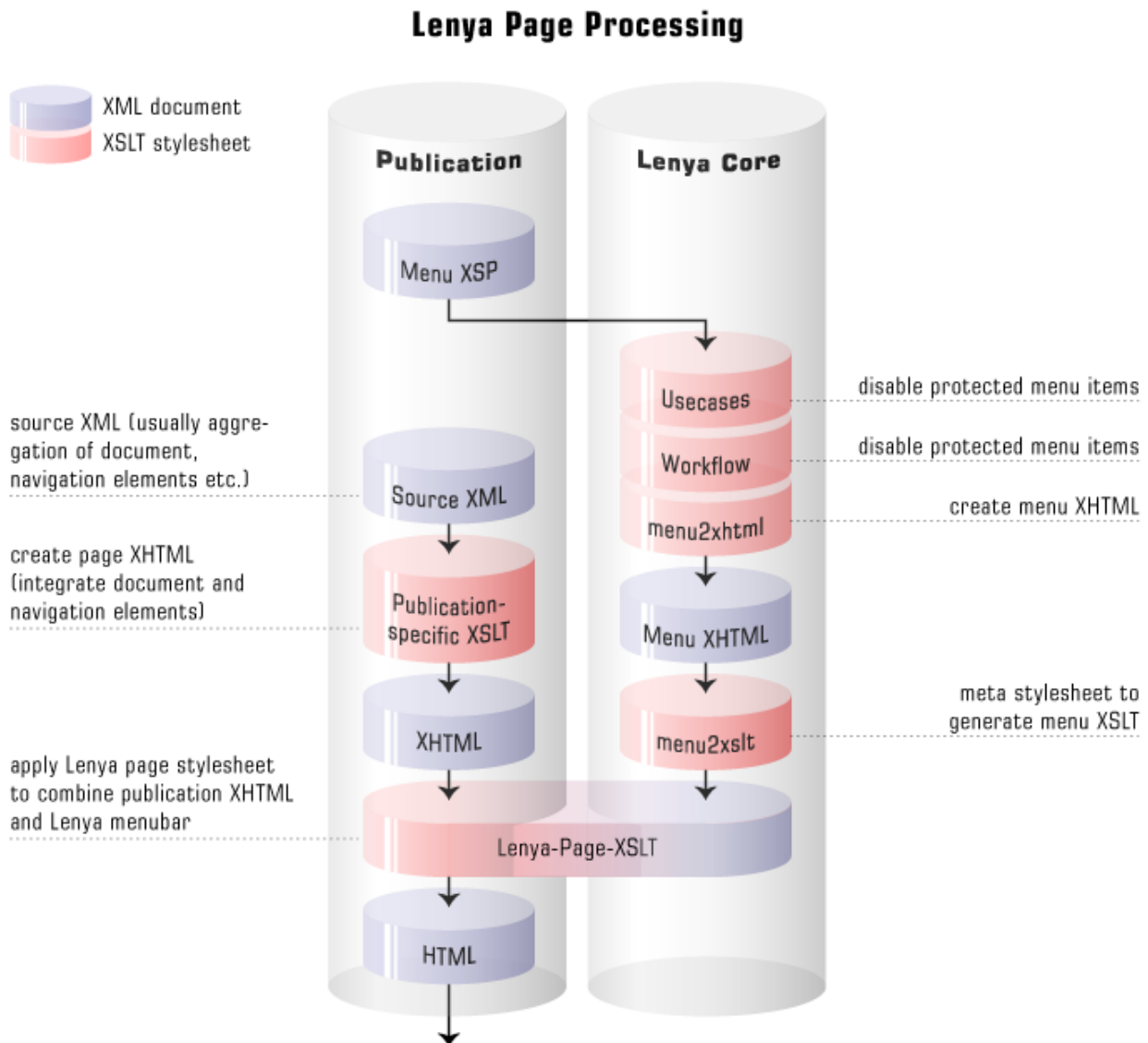
```
?lenya.event=edit&lenya.usecase=edit-forms&lenya.step=open&...
```

If a user is not allowed to invoke the event `edit` or to execute the usecase `edit-forms`, the menu item will be disabled.

## 2. Applying the Menubar - Meta Stylesheet Approach

In this approach, the publication simply builds its XHTML page without any additional XML. The Lenya menubar is added using a meta stylesheet in the last transformation step. You find more information on meta stylesheets on the page [MetaStylesheets](http://wiki.cocoondev.org/Wiki.jsp?page=MetaStylesheets) (<http://wiki.cocoondev.org/Wiki.jsp?page=MetaStylesheets>) in the cocoondev wiki.

### 2.1. The Big Picture (literally)



In the left column you see the XML flow within the publication sitemaps, the right column shows the flow within the core sitemaps.

### 2.1.1. Creating the Menu XHTML

- The actual menu XML is fetched from the sitemap menus.xmap within the publication directory.
- The XML is processed using the UsecaseMenuTransformer and the WorkflowMenuTransformer.
- The stylesheet lenya/xslt/menu/menu2xhtml.xsl transforms the menu XML into XHTML.
- Finally, this XHTML is transformed into a meta stylesheet by lenya/xslt/menu/menu2xslt.xsl.

### 2.1.2. Requesting and Applying the Menubar

- The XHTML page is generated without any additional XML.
- Afterwards, the menubar meta stylesheet is requested using a `cocoon://lenya-page/...` internal request.
- This request is matched in `global-sitemap.xmap`.
- The menubar meta stylesheet is applied to the XHTML page. It inserts the menubar at the top of the HTML body, adds some additional CSS file entries and modifies the HTML title.

## 2.2. Configuring the Publication

### 2.2.1. Requesting the lenya-page Meta Stylesheet

The lenya-page meta stylesheet request looks as follows:

```
<map:generate src="cocoon:/page/..." />
<map:transform src="cocoon://lenya-page/{page-envelope:publication-id}/{menu-url}" />
<map:serialize type="html" />
```

- The `lenya-page` part is used to match the request in `global-sitemap.xmap`.
- The `{page-envelope:publication-id}` part is used to evaluate which publication the request came from.
- The `[menu-url]` part is used to get the menu from `menus.xmap`. Here you have to add all information you need to select the appropriate menubar.

### 2.2.2. Creating the Menu XML

Usually, this is done using an XSP page.

The Lenya menubar namespace is

```
http://apache.org/cocoon/lenya/menubar/1.0
```

### 2.2.3. Supplying Menus in menus.xmap

When the lenya-page meta stylesheet is generated in the core, the actual menu XML is fetched using a request to the sitemap `menus.xmap` in the publication directory. The `[menu-url]` part from the lenya-page request is matched in this sitemap.

This simple `menus.xmap` pipeline uses different menus for authoring and live:

```
<map:pipeline>

  <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:match>
</map:pipeline>
```

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

</map:pipeline>
```

You can as well use the document type or the current URL to select a menu.

Examples for lenya-page requests:

Using the area:

```
cocoon://lenya-page/{page-envelope:publication-id}/{page-envelope:area}
```

Using the document type:

```
cocoon://lenya-page/{page-envelope:publication-id}/{page-envelope:document-type}
```

Using the document URL:

```
cocoon://lenya-page/{page-envelope:publication-id}/{page-envelope:document-url}
```

### 2.3. Creating a Customized Menu

It is also possible to get the menu XML before it is transformed to XHTML using a request to the core. For this purpose, the prefix menu-xml is used instead of lenya-page. This way, you can aggregate the menu XML together with other page parts and include it somewhere inside the XML page.

```
<map:aggregate element="lenya" label="aggregation">
  <map:part src="cocoon://lenyabody/{1}/{2}.xml"/>
  <map:part src="cocoon://menu-xml/{page-envelope:publication-id}/{1}.xml"/>
</map:aggregate>
```

## 3. Applying the Menubar - Classic Approach

### Note:

This approach is deprecated.

### 3.1. XSLT

By editing `src/webapp/lenya/xslt/menu/menu.xml` one can change the look and feel of the Lenya menu bar. This XSLT template is being included by `src/webapp/lenya/xslt/menu/root.xml`.