

Site Navigation How-To

Table of contents

1 Technical Background: How Lenya builds the site navigation.....	2
2 Tabs as main navigation, menubar as sub-navigation.....	3
3 Simple solution: Hide the menus using CSS.....	4
4 Advanced solution: Exclude the menus using XSLT.....	4

One of the ingredients for a good website is a well structured navigation. The Default publication in Lenya comes with a [navigation framework](#) (../docs/1_2_x/components/layout/navigation.html) that is made up of horizontal tabs, breadcrumbs and a menu on the left. A lot of websites use these navigational elements, but depending on the site you're building, you might want to change the navigation or structure it differently.

1. Technical Background: How Lenya builds the site navigation

The source of all navigation elements is the sitetree, which you can find in *PUB_HOME/authoring/sitetree.xml*, *PUB_HOME/live/sitetree.xml*, *PUB_HOME/trash/sitetree.xml* or *PUB_HOME/archive/sitetree.xml* depending on the relevant area.

The tree you can see in the left half of the screen when you access the *Site* tab in authoring mode is pretty much a 1:1 representation of the site tree.

The sitetree.xml document of the default publication which we will use as an example here looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<site il8n:attr="label" label="Authoring"
  xmlns="http://apache.org/cocoon/lenya/sitetree/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://apache.org/cocoon/lenya/sitetree/1.0
    ../../../../resources/entities/sitetree.xsd"
  xmlns:il8n="http://apache.org/cocoon/il8n/2.1"
>

  <node id="index">
    <label xml:lang="en">Home</label>
    <label xml:lang="de">Home</label>
  </node>

  <node id="tutorial">
    <label xml:lang="en">Tutorial</label>
    <label xml:lang="de">Tutorial</label>

    <node id="new_doctype">
      <label xml:lang="en">Create new doctype</label>
    </node>

  </node>

  <node id="concepts">
    <label xml:lang="en">Concepts (english only)</label>
  </node>

  <node id="features">
    <label xml:lang="en">Features</label>
    <label xml:lang="de">Funktionen</label>
  </node>

  <node id="doctype">
    <label xml:lang="en">Document Type Examples</label>
    <label xml:lang="de">Dokumenttyp Beispiele</label>
    <node id="xhtml-document">
      <label xml:lang="en">XHTML Doctype</label>
      <label xml:lang="de">XHTML Dokumenttyp</label>
    </node>
  </node>
</site>
```

```
</site>
```

The default publication defines three navigation elements:

- Tabs (horizontal)
- Breadcrumbs
- Menu (left)

Each of them is built by applying a specific XSLT to sitetree.xml (compare: [navigation framework](#) (../../docs/1_2_x/components/layout/navigation.html)).

In case you need additional navigation elements besides tabs, breadcrumbs and menus, you can set up an additional XSLT file and integrate it into the sitemap. (More on this later.)

2. Tabs as main navigation, menubar as sub-navigation

Depending on the amount of content on your site, you may opt for different styles of navigation. You might for example get rid of the tabs and only use the menu on the left. You can also use the tabs and the menu as parallel means of navigation, which is what the default publication does. If you have a lot of content organized on multiple levels, you might also choose to use the tabs as the first level of navigation and the menu as a second level.

If your sitetree for example looks like this:

- Home
- Products
 - Consumer Products
 - Product Line A
 - Product Line B
 - Product Line C
 - Industrial Products
- Services
 - Maintenance Contracts
 - 24-hour Emergency Hotline

a typical navigation could look this this:

```
+-----+
|               | *HOME* | Products | Services |
+-----+
|
```

when the user clicks on Products, a menu will show up:

```
+-----+
|               | Home | *PRODUCTS* | Services |
+-----+
| - Consumer Products
| - Industrial Products
```

clicking on Services will show up a different menu:

```
+-----+
|               | Home | Products | *SERVICES* |
+-----+
| - Maintenance
|   Contracts
| - 24-hour Emergency
|   Hotline
```

3. Simple solution: Hide the menus using CSS

This behaviour can be achieved by changing the CSS of the default publication. Edit the file

```
$PUB_HOME/resources/shared/css/page.css
```

and add the following snippet, preferably beneath the existing *menublock* declarations:

```
.menublock-1 {
    display: none;
}
```

This way, all non-selected first level menu entries are hidden, which comprises all menu entries belonging to the other tabs.

4. Advanced solution: Exclude the menus using XSLT

In this approach, you use a custom XSLT stylesheet to generate the menubar XHTML snippet. The file has to be placed at

```
$PUB_HOME/lenya/xslt/navigation/menu.xsl
```

Note the step */lenya* in the path, this is necessary to tell Lenya that you're overriding a file from the core. The stylesheet might look as follows:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:nav="http://apache.org/cocoon/lenya/navigation/1.0"
  xmlns="http://www.w3.org/1999/xhtml"
  exclude-result-prefixes="nav"
>

<xsl:import href="../../../../xslt/navigation/menu.xsl"/>

<xsl:template match="nav:site/nav:node[not(descendant-or-self::nav:node[@current = 'true'])]"/>

<xsl:template match="nav:site/nav:node[descendant-or-self::nav:node[@current = 'true']]>
  <div class="menublock-selected-{count(ancestor-or-self::nav:node)}">

    <xsl:apply-templates select="nav:node"/>
  </div>
</xsl:template>

</xsl:stylesheet>
```

- First, you import the core menu XSL so that you can reuse most of the core functionality.
- The first `<xsl:template>` excludes all top-level `<nav:node>` entries which don't contain the current item.
- The second `<xsl:template>` is optional. If you include it, the top-level menu items won't be displayed in the menubar.