

Xopus

0.1 Documentation

by Lenya community

Table of contents

1 About Xopus.....	2
2 Demo.....	2
3 Configuration.....	2
4 Saving via POST.....	2
5 Exit Xopus.....	2
6 Making a page editable with Xopus: Schemas, Views (XSLT) and Menus.....	3
7 Things to watch out for.....	3
7.1 Check your Internet Explorer cache settings.....	3
7.2 Check your local.build.properties.....	3
7.3 Check your schema files.....	3
7.4 Check your CSS files.....	3
7.5 Missing preparexinclude.xml.....	4

1. About Xopus

Xopus is a browser based validating in-place wysiwyg XML editor. It uses Javascript, XML Schema and XSLT. At this time, it only works on Internet Explorer. If you need a cross-platform solution, take a look at BXE. Note that Xopus is not open source, but a commerical product by Q42.

2. Demo

There is an online [XOPUS demo](#) available. You need Internet Explorer 5.5 or newer to view the demo.

3. Configuration

Lenya is expecting an `@XOPUS:CONTEXT@` (e.g. Xopus-2.1.66) directory within your `@XOPUS:PATH@` (e.g. `webapps/ROOT`) directory. `@XOPUS:CONTEXT@` and `@XOPUS:PATH@` are configured within your *local.build.properties* (xopus.context resp. xopus.path). The path is relative to *usecase.xmap*.

4. Saving via POST

To enable saving via POST, one needs to modify `$XOPUS/xopusPlugins/datadriver.js` as follows

```
// Save data START
alert('Save XML via POST');
var result = Application.sendXML(doc, 'POST', name);

switch (result.status)
{
  case 200:
    //OK
    var resultXML = result.responseXML;
    alert('XML saved');
    return;
  case -1:
    alert('Error while saving ' + name + ':\n\n' + result.responseText);
    break;
  default:
    alert('Server responded not ok while saving ' + name + ':\n\nstatus:' + result.status +
'\nmessage: ' + result.responseText);
}
// Save data END
```

You may change the text of the alerts by something else.

5. Exit Xopus

To exit Xopus one needs to modify `$XOPUS/xopusPlugins/cmsdriver.js` as follows

```
getExitURL: function(path)
{
  exitURL = path.substring(0, path.indexOf('?'));
  alert('Exit to: ' + exitURL);
  return exitURL;
},
```

You may replace the text of the alert by something else.

6. Making a page editable with Xopus: Schemas, Views (XSLT) and Menus

Xopus needs an XML Schema for validation and an XSLT to view the document. The Schemas and XSLTs need to be located at:

```
$PUBLICATION/config/doctypes/schemas/$DOCTYPE.xsd
$PUBLICATION/lenya/xslt/xopus/$DOCTYPE.xsl
```

You might want to use [Trang](#) to generate XML Schemas.

Editing with Xopus is implemented as a usecase. One can add a menu item as follows:

```
<item uc:usecase="xopus" uc:step="open" href="?doctype=$DOCTYPE">Edit&#160;with&#160;Xopus</item>
```

In addition, you may want to study the oscom publication which has an Xopus integration example, as well as the root xopus.xmap, the xopus.xmap in the oscom publication, and the section in usecase.xmap pertaining to xopus. This will help you understand how the integration was done (which admittedly is a bit clunky as it predates many Lenya concepts).

7. Things to watch out for

7.1. Check your Internet Explorer cache settings

Make sure that the page is loaded from scratch everytime you visit it, because otherwise you'll always see a cached version. This cached version will be saved if you click on 'save' in Xopus.

7.2. Check your local.build.properties

Your variable settings could look like this:

```
xopus.context=Xopus-2.1.72    root directory for all Xopus files
xopus.path=../../ROOT        relative path from usecase.xmap to Xopus root
                              directory
```

7.3. Check your schema files

I had troubles with schemas including other schema files. You have to insert a separate pipeline for this kind of architecture. Unfortunately I couldn't figure out if I had to put this matcher in '\$PUBLICATION/publication-sitemap.xmap' or elsewhere. I ended up using one and only one schema file for a specific doctype.

7.4. Check your CSS files

You have to provide a stylesheet for Xopus in '\$PUBLICATION/lenya/xslt/xopus/\$DOCTYPE.xsl'. If you use a *.css file to control layout and style of the XML file displayed by Xopus, you'll have to insert a pipeline matcher in '\$PUBLICATION/publication-sitemap.xmap':

```
<map:pipeline>
  <map:match pattern="**/xopus.css">
    <map:read src="resources/shared/css/xopus.css" mime-type="text/css"/>
  </map:match>
</map:pipeline>
```

7.5. Missing preparexinclude.xsl

If Xopus complains that it cannot find `$XOPUS/xopusPlugins/preparexinclude.xsl`, then add the following at that location:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:x="http://www.w3.org/2001/XInclude"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    >

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()" />
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```