

Chapter 7. Packaging Custom Work

Last Updated Monday, 24 April 2006

Overview

{glossarbot=disable}Joomla comes with an easy-to-use installer for Modules, Mambots, Components and Templates. Using the installer, the site administrator can add features and templates by uploading one zipped file. This topic describes how to create the Mambot, Module, Component and Template setup files that are used by the installer. The XML Setup File

All installations using the installer require a text file marked up in XML. The XML file describes the installation, its authors, and the files to be installed. Only a passing familiarity with XML is required. Each XML file begins with the prologue, `<?xml version="1.0" encoding="iso-8859-1" ?>`. Thereafter, several nested sections appear, all of which are nested in the root, `<mosinstall>`. The `<mosinstall>` element has at least two attributes: "type" is the type of Joomla element you are installing and "version" is the version of Joomla that the element is written for. A third attribute called "client" is used for some Administrator elements. `<?xml version="1.0" encoding="iso-8859-1" ?>`
`<mosinstall type="installtype" version="4.5.1">`

...
`</mosinstall>`XML elements that are common to all types of installation are listed below. `<name>` The name element is required. It is used in menus, etc. `<creationDate>` The creation date for the XML file or component. There is no specifically defined format for this date; it is just a string. `<author>` The author of the Component, Module, Template or Mambot. `<copyright>` Copyright information. `<license>` The license under which the element is being released. `<authorEmail>` The email address of the author. `<authorUrl>` The author's URL. `<version>` The version of the package. `<description>` A brief description of the Component, Module, Template or Mambot. `<files>` Optional, but if you don't have it, no files will be installed. The `<files>` section is used to tell the installer which files it should install. There is no limit to the number of `<filename>` elements you can have in this section. Depending on the the installation type, files are copied to `/modules`, `/components/[component_name]` or `/templates/[template_name]`. Parameters

Module and Component setup files may also have a block defining parameters, for example: `<params>`
`<param name="count" type="text" default="5" label="Number of items"`
`description="The number of items to show" />`
`</params>`

Refer to the appendix on parameters for more information. XML Do's and Don'ts

All XML files must be well-formed without exception. The XML parser used by Mambo will not parse malformed XML documents. While there are numerous rules as to what makes a well-formed XML document, here is a list of the major checkpoints:

- Your XML file must not have any whitespace or characters in front of the XML declaration. For example, this is legal if it shows the start of a file:

```
<?xml version="1.0" encoding="iso-8859-1"?>This is not legal:
<?xml version="1.0" encoding="iso-8859-1"?>Nor is this:
bad<?xml version="1.0" encoding="iso-8859-1"?>
```

- You must match every starting element with the same ending element. Note that element names are case sensitive. in XML. For example: `<LeGaL>`This is ok`</LeGaL>`
`<illegal>`This is not`</ILLEGAL>`

You may nest element tags but they may not overlap. For example, the following is malformed: `<name>`Main
`<author>`Harry`</name></author>`

There can only be exactly one root element. For example, the following is a malformed document because there are two root elements: `<?xml version="1.0" encoding="iso-8859-1"?>`

```
<mosinstall type="module">
<!-- the XML definition -->
</mosinstall>
<mosinstall type="component">
<!-- the XML definition -->
</mosinstall>
```

All attributes in a tag must be quoted. For example, the following is malformed because of the missing quotes, as you might often see in HTML documents (where it might be tolerated): `<mosinstall type=module>`

You may not have more than one tag attribute with the same name. For example, the following is malformed: `<mosinstall type="module" type="component">`

-

You must escape all `<` and `&` signs that occur within the character data or attributes of a tag. For example: `<menu link="option=com_foo&task=bar">Sub < menu 1</menu>`

-

To use HTML markup in the character data of a tag, you must enclose the text in a CDATA section. For example, the following example allows for the `©` markup to be placed in the copyright tag: `<?xml version="1.0" encoding="iso-8859-1"?>`

```
<mosinstall type="template">
  <name>How flung</name>
  <copyright><![CDATA[&copy; 2004 Me]]></copyright>
```

There are a number of tools you can use to check that your XML is well-formed. One method is to open the XML file in a modern browser. If it is well-formed you will likely see a rendering like this (but longer):

```
- <mosinstall type="template">
  <name>How flung</name>
</mosinstall>
```

A malformed XML file may look like this:

```
XML Parsing Error: mismatched tag. Expected </name>.
Location: file:///C:/My%20Documents/test.xml
Line Number 4, Column 3:
```

```
</mosinstall>
--^Further Reading...
Template setup file
Module setup file
Mambot setup file
Component setup file
```