

Thanks to MamboServer.com for this tutorial.

Mambo comes with a built-in installer to make it easy for the normal user to install modules, components and templates. Mambo Open Source (MOS) 4.5 features powerful support for custom script integration featuring auto-installation, API functions and integrated administration.

Components are custom PHP scripts that can be run in the middle body of Mambo while Modules are usually run in side column boxes.

Common sections

Here we describe what a developer of modules and components or a template designer have to do in order to take advantage of the installer.

All types of installations require an XML file, which describes what to install. There are some common sections in all types of installation.

Common sections

```
<mosinstall type="installtype">  
</mosinstall>
```

The <mosinstall> section is required, otherwise the installer will ignore the rest.

```
<name>the name</name>
```

The <name> section is required. It's used in menus etc.

```
<creationDate>24/07/2003</creationDate>  
Date of the creation
```

```
<author>Test Designer</author>  
Author of the module/component/template
```

```
<copyright>This template is released under the GNU/GPL  
License</copyright>  
Copyright information
```

```
<authorEmail>info@mambotemplates.com</authorEmail>  
Autors email
```

```
<authorUrl>www.mambotemplates.com</authorUrl>  
Author URL
```

```
<version>1.0</version>
```

Version of module/component/template

```
<files>
<filename>filenam.xxx</filename>
</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 shall install.

There is no limit in the number of files you can have in this section. Depending on the the installation type, files are copied to /modules, /components/[component_name] or /templates/[template_name]

Modules

Module installation:

The XML for a module installation could look like this:

```
<?xml version="1.0" ?>
<mosinstall type="module">
<name>Hit Counter</name>
<creationDate>24/07/2003</creationDate>
<author>Test Designer</author>
<copyright>This template is released under the GNU/GPL
License</copyright>
<authorEmail>info@mambotemplates.com</authorEmail>
<authorUrl>www.mambotemplates.com</authorUrl>
<version>1.0</version>
<description>Show hits to the site</description>
<files>
<filename module="mod_counter">mod_counter.php</filename>
</files>
</mosinstall>
```

There need to be ONE and only one <filename> that have an attribute 'module'm this attribute is used as a system var

Components

Component installation:

A component differs from a module in the way that it's more complex, and can have extra features to maintain data.

The XML for a component installation could look like this:

```
<?xml version="1.0" ?>
<mosinstall type="component">
<name>My component</name>
<creationDate>24/07/2003</creationDate>
<author>Test developer</author>
<copyright>This component is released under the GNU/GPL
License</copyright>
<authorEmail>info@example.com</authorEmail>
<authorUrl>www.example.com</authorUrl>
<version>1.0</version>
<description>This is a brief description of the component</description>
<files>
<filename>mycomponent.php</filename>
<filename>mycomponent.html.php</filename>
</files>
<images>
<filename>images/approve.png</filename>
</images>
<install>
<queries>
<query># create a table</query>
<query># populate new table</query>
</queries>
</install>
<uninstall>
<queries>
<query># delete the table</query>
</queries>
</uninstall>
<installfile>install.mycomponent.php</installfile>
<uninstallfile>uninstall.mycomponent.php</uninstallfile>
<administration>
<menu>My component</menu>
<submenu>
<menu act="sub1">Sub menu 1</menu>
<menu act="sub2">Sub menu 2</menu>
</submenu>
```

```
<files>
<filename>admin.mycomponent.php</filename>
<filename>admin.mycomponent.html.php</filename>
<filename>toolbar.mycomponent.php</filename>
<filename>toolbar.mycomponent.html.php</filename>
</files>
<images>
<filename>administrator/images/approve.png</filename>
</images>
</administration>
</mosinstall>
```

As you can see it has more sections, the sections that differ from the common ones is described below.

```
<images>
<filename>images/approve.png</filename>
</images>
```

The <images> is optional, but can be use to copy component specific images to /components/[component_name]/images.

There is no limit in the number of <filename>

As the nature of a component is more complex you can use the <install> section to specify creation of tables, inserting default data.

In the <install> section you can have <queries>. <queries> is where you can have all the sql statements that your component require, eg. creation of new tables, and adding data to these new tables.

There is no limit in the number of <query> sections.

NOTE on queries:

If you have HTML in the queries you need to add <![CDATA[in front and]]> at the end. Example

```
<query>
<![CDATA[
INSERT INTO mos_mytable VALUES (1,
'<P>HTML tags in quieris</P><br><a href="http://www.mysite.com">My
Site</a>');
]]>
</query>
```

<installfile> section is used to specify an additional file, where you can add additional code that will be called at the end of the installation. The file

MUST have a `com_install()` function in order to work. In the `com_install()` function you can have additional functionality to make the component work. The `com_install()` function must return a string with a status, which will be displayed to the user at the end of the installation.

`<uninstallfile>` section is used to specify an additional file, where you can add additional code that will be called at the end of an install. The file MUST have a `com_uninstall()` function in order to work. In the `com_uninstall()` function you can have additional functionality to make the clean up.

The `com_uninstall()` function must return a string with a status, which will be displayed to the user at the end of the un-installation.

The `<administrator>` section is used to add functionality to the administrator part of Mambo.

The `<menu>` section is used to specify the name of the menu for maintaining the component, it will appear under the Components menu.

If your component needs it, you can have sub menus, use the `<submenu>` section to add sub menus.

The `<menu>` section of a sub menu has an additional attribute `act`, which is passed to your module. The `act` attribute is required for sub menus.

The `<administrator>` section also has `<files>` and `<images>`. These sections work the same way as described before, with the difference that files are copied to `/administrator/components/[component_name]/` and images to `/administrator/components/[component_name]/images/`

Templates

Template installation:

The XML for a template installation could look like this:

```
<?xml version="1.0" ?>
<mosinstall type="template">
<name>My template</name>
<creationDate>24/07/2003</creationDate>
<author>Test Designer</author>
<copyright>This template is released under the GNU/GPL
License</copyright>
<authorEmail>info@mambotemplates.com</authorEmail>
<authorUrl>www.mambotemplates.com</authorUrl>
<version>1.0</version>
<description>This is a breif description of the template</description>
<files>
<filename>index.php</filename>
<filename>template_thumbnail.png</filename>
</files>
<images>
<filename>images/arrow.png</filename>
<filename>images/mt_business_back.png</filename>
<filename>images/mt_business_bottom.png</filename>
<filename>images/mt_business_top.png</filename>
<filename>images/mt_menu_back.jpg</filename>
</images>
</mosinstall>
```

The sections works the same way as described before, with the difference that files are copied to `/templates/[template_name]/` and images are copied to `/templates/[template_name]/images/`

Variables

Take care of your variable names. Use unique variable names and not names reserved by Mambo (such as `$username` or `$session`).