

Making installation package for your Joomla component (Joomla! 1.0.x)

This tutorial will explain you how to make an installation package for you component for Joomla 1.0.x. version. In the upcoming Joomla 1.5.x installation xml file is different and will be explained in new article.

Introduction

Although this article should be read after you build your first component it would be good to read it before (and again later, of course). That is because we recommend you to build an empty component then install it on the working Joomla site and build (and debug) it from there.

This tutorial will explain you how to make an installation package for you component for Joomla 1.0.x. version. In the upcoming Joomla 1.5.x installation xml file is different and will be explained in new article.

Before you begin...

Installation package is a compressed file (zip) containing information about files that are included in the package (frontend and backend), database preparation and uninstall information.

If you are not experienced with Joomla component development (even if you are) it is difficult to know the structure of files and database before you complete your component. That is why we said in the introduction that it is recommended to make an empty component and then deploy it to working Joomla site. Then you will be able to see how your development is progressing, and you will be able to test and debug your component. After you are done, you will know the list of files and database structure. Then you are ready to make an complete installation package. That step is not always cakewalk as you will need to test and debug installation package so that your component works after installations (you must make sure all files are included and noted in xml file, also the database structure is fully created and uninstall must remove all traces of your component).

Making your first installation package

First we will make installation package that will install empty component called 'Test'. You should create an empty folder `com_test` in which you will create two files: `test.php` and `test.xml`.

The prefix '`com_`' is standard for all components. If you want to create a component called 'PhoneBook' you would have to create a folder `com_phonebook`, which would contain files: `phonebook.php` and `phonebook.xml`.

Let's put the following content into `test.php`.

```
<?php  
  
    echo "This is a test component";  
  
?>
```

You will later add the real code here as this is the base file for all other files that you component will use.

Now we will create a simple xml file that will explain Joomla how to install the component.

```
<?xml version="1.0" ?>  
<mosinstall type="component">  
    <name>test</name>  
    <creationDate>02.11.2006</creationDate>  
    <author>John Smith</author>  
    <copyright>(C) Copyright 2004-2007 by John Smith. Released under the GNU/GPL License.</copyright>  
    <authorEmail>john.smith@example.com</authorEmail>  
    <authorUrl>www.example.com</authorUrl>  
    <version>1.0.0</version>  
    <description>Test component.</description>  
    <files>  
        <filename>test.php</filename>  
        <filename>test.html.php</filename>  
    </files>  
</mosinstall>
```

This is a very simplified version of xml installation file, but will do what we need.

Component installation and testing

You should have a folder named `com_test` and files `test.php` and `test.xml` inside that folder. Now, create archive containing that folder (xp users can RMB click on the folder and click Send To -> Compressed (zipped) folder).

Log-in to Joomla administration with user who has administrative privileges (preferably admin) and install the component (Installers -> Components).

You should get a message that the component has been successfully installed.

Now to test it open the frontend of your Joomla site and after url add "`index.php`" (if it isn't there already) and "`?option=com_test`" (for example `http://www.joomla-example.com/index.php?option=com_test`).

This will tell Joomla to execute the component, so you should see the message "This is a test component" in the content area.

Now you can go to folder "`components/com_test`" and open `test.php` file. You can now code anything you want and create other files and folders which you will link from `test.php` file.

Administration part of component

Now we will explain how to add administration part of component. Example above created only the frontend files which is usually only half of component code.

Administration files will be installed into "`administrator/components/com_test`" folder. Administration part of Joomla is separated from frontend, and everything that is related to your component configuration should be there.

Let's put the following lines above `</mosinstall>` tag.

```
<administration>
```

```
  <menu>Test</menu>
```

```
  <submenu>
```

```
<menu task="">Test administration</menu>

</submenu>

<files>

  <filename>admin.test.html.php</filename>

  <filename>admin.test.php</filename>

</administration>
```

The first two tags tells Joomla to create an administration menu items for your component. And the 'files' tag tells which files need to be copied to administrator/com_test folder.

You can use attribute 'task' in 'menu' tag to specify a task parameter so that you know what menu item user clicked from component administration menu. Since we use only one file for this example we don't need task parameter.

That's it. You if you install the component now you should be able to access it trough administrator interface.

Other tags that can be used in the installation file

There are a few other tags that can be used to deploy your component.

To specify which image files your component needs you can use '<image>' tag, which contain '<file>' tags for each image.

```
<images>

  <filename>images/logo.png</filename>

</images>
```

You can use the '<image>' tag in frontend and administrator part of installation file.

To execute a php script after successful installation you can use '<installfile>' tag.

```
<installfile>install.test.php</installfile>
```

Also, after uninstalling the component you can use '<uninstallfile>' tag.

```
<uninstallfile>uninstall.test.php</uninstallfile>
```

These tags can be very useful if you want to give some information to user after install/uninstall, or if your component needs to perform some operations before first usage or after uninstall.

Database preparation

If your component uses database you will need to add SQL code to your install file so that Joomla can execute it in the installation process.

This can be easily done if you have functional component on your development Joomla site. All you need to do is to empty data from your tables (leave the data that needs to be there so that you component can be functional), and export

the database using a tool you use for database administration (Navicat, phpMyAdmin...)

Now you need to add the exported code line by line (not actually the text line, but the sql line which ends with delimiter ';') like this:

```
<queries>
  <query>
    CREATE TABLE IF NOT EXISTS `#__test_table1` (
      `id` int(11) NOT NULL default '0',
      `field1` varchar(255) NOT NULL default "",
      `field2` varchar(255) NOT NULL default "",
      `field3` varchar(255) NOT NULL default "",
      PRIMARY KEY (`id`)
    ) TYPE=MyISAM;
  </query>
  <query>
    CREATE TABLE IF NOT EXISTS `#__test_config` (
      `id` int(11) NOT NULL auto_increment,
      `field1` varchar(255) NOT NULL default "",
      `field2` int(11) NOT NULL default '0',
      PRIMARY KEY (`id`)
    ) TYPE=MyISAM AUTO_INCREMENT=1 ;
  </query>
</queries>
```

Every sql line is contained in the '<query>' tag. You can execute whatever you need, even modify existing Joomla tables (not recommended though, as it can give your component bad reputation if not done very carefully).

Summary

Making installation script becomes more and more complex as your component grows. Also the maintance of your installation is a hard work as you will change the code and files from version to version.

It would be a good thing to have your own organization schema that will help you track changes and implement them in the install.xml file.