

ELXIS CMS – DEVELOPERS GUIDE

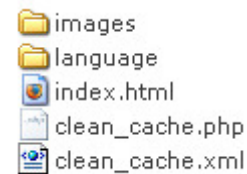
Tools

General

Elxis Tools are small semi-autonomous administration applications attached to component admin, the main administration component. The purpose of Tools creation and implementation in Elxis CMS had been to easily and quickly add extra features in Elxis administration using existing or not PHP scripts. At this moment (Elxis version 2006.3) there is no available installer for Tools as this has been scheduled for the next massive release of Elxis. In order to install a tool at this moment, you must upload its files using ftp directly, to the relative administration folder. If you wish to uninstall a tool you just delete its folder and files.

Tools structure

Each tool has a unique name and is located inside a folder with the same name in directory /administrator/tools. For instance the tool floodblocker is located inside folder /administrator/tools/floodblocker. Each tool consists of a number of files that we will refer to as "package" in general. Each package has at least 2 files with the same name as the tool's name.



sample tool structure

The first one is a PHP file and is the main tool, and the second one is an XML file containing some information about the tool.

XML file

A tool's XML file does not differ so much regarding to the module or component XML files. There are only 3 differences. Firstly, the extension's type is "**tool**". This can be seen in the XML's second line:

```
<mosinstall type="tool" version="2006">
```

There is also an optional line included inside the tags `<icon></icon>`. This sets the tool's icon that will be displayed in the tool's list. If we do not add this line the default icon will be displayed instead. Example:

```
<icon>images/myicon.png</icon>
```

Every path in the XML file starts from the folder that our tool is installed in.

XML, apart from the information located in its header, also contains a list of the relative path of all the files that constitute the tool's "package" (like in modules and components). This list is included inside the tags `<files></files>`. The main tool's file is declared in this list such as:

```
<filename tool="tool_name">tool_name.php</filename>
```

PHP file

As we mentioned, apart from the XML file, each tool "package" contains at least one more file, which is a PHP file named after the tool. This file's contents can vary depending on the tool's functionality. There are no coding standards as this would be contrary to the purpose of the tools existence (for any PHP script to be easily attached to Elxis administration). Further down we will see how we can create such tools through some examples beginning from simple cases and concluding to more complicated ones.

Example 1 – time and date

For our first example, we will create a tool that will display the current date and time. We will name this tool showtime.

We begin by creating the XML file.

```
<?xml version="1.0"?>
<mosinstall type="tool" version="2006">
  <name>Show Time</name>
  <author>Ioannis Sannos</author>
  <creationDate>19/11/2006</creationDate>
  <copyright>(C) 2006 Elxis.org</copyright>
  <license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
  <authorEmail>admin@elxis.org</authorEmail>
  <authorUrl>www.elxis.org</authorUrl>
  <version>1.0</version>
  <icon>images/time_image.png</icon>
  <description>This tool displays current date and time</description>
  <files>
    <filename tool="showtime">showtime.php</filename>
    <filename>images/time_image.png</filename>
    <filename>images/index.html</filename>
  </files>
</mosinstall>
```

Regarding the above XML file, the tool's package will be consisted of 5 files whose paths will be:

```
/administrator/tools/showtime/showtime.php
/administrator/tools/showtime/showtime.xml
/administrator/tools/showtime/index.html
/administrator/tools/showtime/images/time_image.png
/administrator/tools/showtime/images/index.html
```

Index files (index.html) are empty and we add them to prevent directory browsing if the server that hosts our site permits such thing.

We create the showtime.php file and we open it to edit it. We insert the first lines of the code:

```
<?php  
  
defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' );  
  
echo date('d-m-Y H:i:s');  
  
?>
```

The first line prevents direct access to this file from outside Elxis CMS. We must never forget to add this line in any Elxis extension! The second line prints the current date and time. We save and close this file. Our tool is ready! If you wish, you can use HTML and/or CSS to beautify its appearance.

If we upload the files we created in the folder /administrator/tools/showtime, our tool will be displayed in the list of installed tools and by clicking on its name the tool will be executed.

The above tool was too simple. Let's now create something more complicated, interactive to Elxis, with sub-routines. We will build a tool that will check if there is a renamed installation folder and if yes, will prompt us to delete it.

Example 2 – Check installation folder's existence.

The existence of the installation folder (/installation) in a functional site is dangerous. Many users have the habit to rename this folder instead of delete it. The tool that we will create will check if this folder (renamed or not) exist, and if yes will prompt us to delete it. Let's name our tool check_install_dir.

We begin by creating the PHP file. The path of this file when we finish will be:

```
/administrator/tools/check_install_dir/check_install_dir.php
```

We add the first lines of code in this file:

```
<?php  
  
defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' );
```

We declare some global variables that we will use. File Manager (\$fmanager) for handling files and folders, mosConfig_absolute_path that gives us the Elxis absolute path and administration language (\$adminLanguage).

```
global $fmanager, $mosConfig_absolute_path, $adminLanguage;
```

At this point we must add some lines of code that we will talk about later. Let's leave a notice here to come back later:

//block 1, we will add code here later

Installation folder, if exists, will be inside the Elxis root directory. We get a list of all the directories located inside the root folder:

```
$dirs = $fmanager->listFolders($mosConfig_absolute_path);
```

Installation folder contains some special files. In order to determine if any of the folders we got above is a renamed installation folder we will check if there is a file named install3.php inside any of these folders. If such a folder exist its name will be stored inside the variable \$idir. To make sure that this is the right folder we will also check if there is a file named elxis.xml inside the sub-directory "schema" of that folder.

```
$idir = "";
foreach ($dirs as $dir) {
    $files = $fmanager->listFiles($mosConfig_absolute_path.'/'.$dir);
    if (count($files) >0) {
        foreach ($files as $file) {
            if ($file == 'install3.php') {
                if (file_exists($mosConfig_absolute_path.'/'.$dir.'/schema/elxis.xml')) {
                    $idir = $dir;
                    break;
                }
            }
        }
    }
}
}
```

We now check if we found a renamed installation folder and if yes we create a link in order the user to be able to delete this folder if he wishes:

```
if ($idir != "") {
    $turl = 'index2.php?option=com_admin&task=tools&name=check_install_dir';
    echo 'Installation folder found under name: '.$idir;
    echo '<br />Do you want to delete this folder?';
    echo '<br /><a href="'. $turl. '&act=del&deldir='.base64_encode($idir). '"><b>';
    echo $adminLanguage->A_DELETE.'</b></a>';
} else {
    echo 'Installation folder not found. Your site is secure.';
}
}
```

We now go back to the notice we left before as block 1. We will add there the delete procedure of the renamed folder we found. Of course user will have to click on the link we created before in order for this action to take place.

```
$act = mosGetParam($_REQUEST, 'act', "");
$deldir = mosGetParam($_REQUEST, 'deldir', "");
if (($act == 'del') && ($deldir != "")) {
    $deldir = $mosConfig_absolute_path.base64_decode($deldir);
    if (file_exists($deldir) && is_dir($deldir) && file_exists($deldir.'/install3.php')) {
        $deleted = $fmanager->deleteFolder($deldir);
    }
    echo ($deleted) ? 'Folder deleted successfully! ' : 'Could not delete folder '.$deldir;
    echo "<br /><br />\n";
}
```

We almost finish working on this file. We add the PHP closing tag at the end of the file and we save this file as check_install_dir.php

```
?>
```

We create the required XML file (check_install_dir.xml) and we upload everything in folder /administrator/tools/check_install_dir. The tool is ready for use!

Of course we can beautify tool's appearance, add more functions, images, CSS, full language support and anything else we like.

Epilogue

I hope you got an idea of how to create tools for Elxis CMS and that I motivate you to create your own useful tools. Elxis Team is always available to provide you technical support through the Elxis CMS forum located at this URL: <http://forum.elxis.org>. If you ask for professional support and solutions by experts, GoUp (the company that created Elxis CMS) is always at your disposal (<http://www.goup.gr>).

Sannos Ioannis,
Contact: datahell@elxis.org

Tuesday, November 22, 2006
Elxis Team / Go Up Inc.