Plugins

Overview

Plugins serve a variety of purposes. As modules enhance the presentation of the final output of the Web site, plugins enhance the data and can also provide additional, installable functionality. Plugins enable you to execute code in response to certain events, either Joomla! core events or custom events that are triggered from your own code. This is a powerful way of extending the basic Joomla! functionality.

See System Events for a list of core plugin events.

See Component Events for a list of component plugin events.

Core Types

Plug-ins are managed at a group level that is defined in the plug-in's XML manifest file. While the number of possible types of plugins is almost limitless, there are a number of core plugin types that are used by Joomla!. These core types are grouped into directories under /plugins. They are:

- authentication
- content
- editors
- editors-xtd
- search
- system
- user
- xmlrpc

Authentication
: plugins allow you to authenticate (to allow you to login) against different sources. By default you will authenticate against the user database when you try to login. However, there are other methods available such as by OpenID, by a Google account, LDAP, and many others. Wherever a source has a public API, you can write an authentication plugin to verify the login credentials against this source. For example, you could write a plugin to authenticate against Twitter accounts because they have a public API.

Content
: plugins modify and add features to displayed content. For example, content plugins can cloak email address or can convert URL's into SEF format. Content plugins can also look for markers in content and replace them with other text or HTML. For example, the Load Module plugin will take {*loadmodule banner1*} (you would remove the *'s in practice. They are included to actually prevent the plugin from working in this article), load all the modules in the banner1 position and replace the marker with that output.

Editor
: plugins allow you to add new content editors (usually WYSIYWG).

Editor-XTD
: (extended) plugins allow you to add additional buttons to the editors. For example, the *Image*, *Pagebreak* and *Read more* buttons below the default editor are actually plugins.

Search
: plugins allow you to search different content from different components. For example, search plugins for Articles, Contacts and Weblinks are already provided in Joomla!.

System
: plugins allow you to perform actions at various points in the execution of the PHP code that runs a Joomla! Web site.

    User
    : plugins allow you to perform actions at different times with respect to users. Such times include logging in and out and also saving a user. User plugins are typically user to "bridge" between web applications (such as creating a Joomla! to phpBB bridge).

    XML-RPC
    : plugins allow you to provide additional XML-RPC web services for your site. When your Web site exposes web services, it gives you the ability to interact remotely, possibly from a desktop application. Web services are a fairly advanced topic and will not be covered in much detail here.

Directory & File Structure

While a plugin can have any number of files, there are two you need as a minimum and there are specific naming conventions you must follow. Before we look at the files, we must decide what sort of plugin we are going to create. It must either fall under one of the built-in types (authentication, content, editors, editors-xtd, search, system, user or xmlrpc) or your can create your own type by adding a new folder under /plugins. So, files for an authentication plugin will be saved under /plugins/authentication, files for a system plugin will be saved under /plugins/system, and so on.

The typical plugin install location and files:

```
/hubzero
  /plugins
    /{PluginType}
        /{PluginName}
          {PluginName}.php
          {PluginName}.xml
```

As mentioned, a plugin has a minimum of two files: a PHP file, test.php, which is the file actually loaded by the system and an XML file, text.xml, which contains meta and installation information for the plugin as well as the definition of the plugin parameters.

There is no restriction on the file name for the plugin (although we recommend sticking with alpha-numeric characters and underscores only), but once you decide on the file name, it will set the naming convention for other parts of the plugin.

Examples

A plugin demonstrating basic setup:

Download: System Test plugin (.zip)