

Extending Mura CMS with Plugins - Part III - Configurable Settings

Posted At : January 6, 2010 8:40 AM | Posted By : Bob Silverberg

Related Categories: ColdFusion, Mura CMS

I'm continuing my series on Extending **Mura CMS** (an open source ColdFusion CMS) with Plugins, after a long hiatus, by looking at adding configurable settings to a plugin. In the **previous post** we looked at creating a simple Hello World plugin and also looked at how we could place the output of that plugin on a page in our Mura site.

As you may recall, all that plugin did was output the text "Hello World!" inside an h1 tag. That's not particularly useful, so let's make it a bit more interesting. How about we allow the person managing the Mura site to decide whom we should greet and what that greeting should be? We're going to do that by adding a couple of settings to our plugin. In order to add those settings we're going to have to make a few changes to our plugin's configuration, which we do by editing the */plugin/config.xml* file. Let's take a look at our new version:

```
<plugin>

<name>Hello World Plugin Step 2</name>

<package>HelloWorldPluginStep2</package>

<version>1.0</version>

<provider>SilverWare Consulting</provider>

<providerURL>http://www.silverwareconsulting.com</providerURL>

<category>Sample Application</category>

<settings>

<setting>

<name>toWhom</name>

<label>We are speaking to</label>

<hint>The name of the person to whom we're speaking</hint>

<type>text</type>

<required>true</required>

<validation></validation>

<regex></regex>

<message></message>

<defaultValue></defaultValue>

<optionlist></optionlist>

<optionlabellist></optionlabellist>

</setting>

<setting>

<name>sayWhat</name>

<label>We are going to say</label>

<hint>This is what we are going to say to them</hint>

<type>select</type>

<required>true</required>

<validation></validation>

<regex></regex>

<message></message>

<defaultValue>Hello</defaultValue>

<optionlist>Hello~Goodbye</optionlist>

<optionlabellist></optionlabellist>

</setting>

</settings>

<displayobjects location="global">

<displayobject name="Hello World Message"

displayobjectfile="displayObjects/dspHelloWorld.cfm"/>

<displayobject name="Greeting Message"

displayobjectfile="displayObjects/dspGreeting.cfm"/>

</displayobjects>

</plugin>
```

Let's review the changes. First we changed the *name* and *package* elements to reflect the fact that this is a new plugin. Next we added two settings to the *settings* element: *toWhom*, which will be used to store the name of the person that we will be greeting, and *sayWhat*, which will be used to store the greeting to use. Here's a rundown of all of the elements available for each setting:

1. *name* - The name of the setting, which you will use in code that needs to know the value of the setting.
2. *label* - The label to display for the setting on the Plugin Settings page.
3. *hint* - The hint to display for the setting on the Plugin Settings page.
4. *type* - The type of control to present to an administrator for the setting on the Plugin Settings page. Valid values are: Text, TextBox, TextArea, Select, SelectBox, MultiSelectBox, Radio, RadioGroup.
5. *required* - Is a value required when updating the plugin via the Plugin Settings page?
6. *validation* - What sort of validation should be performed when an administrator changes the setting via the Plugin Settings page? Valid values are: ""

(empty string - meaning no validation), Date, Numeric, Email, Regex.

7. *regex* - The regex to be used when the validation type is "Regex".

8. *message* - The message to display if the validation fails.

9. *defaultvalue* - The default value to present on the Plugin Settings page if no value has been specified for the setting.

10. *optionlist* - A list of options to present to an administrator for the option. This is used with the Select, SelectBox, MultiSelectBox, Radio, and RadioGroup types. The list items should be delimited with a caret (^) symbol.

11. *optionlabellist* - An optional list of labels that correspond the list of options specified in *optionlist*.

We also added a new *displayobject* element which will be used for our new configurable greeting. After adding the two settings above and the new displayobject, when we install our new plugin the Plugin Settings page for it should look something like this:

Plugin Settings

Name: Hello World Plugin Step 2

Category: Sample Application

Version: 1.0

Provider: SilverWare Consulting

Provider URL: <http://www.silverwareconsulting.com>

Plugin ID: 19

Package: HelloWorldPluginStep2

Plugin Name (Alias)

We are speaking to

We are going to say

Display Objects

Hello World Message

Greeting Message

Display Objects Location

Now all we have to do is tell our new display object to use those settings. Here's what dspGreeting.cfm will look like:

```
<cfoutput>
<h1>Greeting</h1>
<p>#pluginConfig.getSetting("sayWhat")# #pluginConfig.getSetting("toWhom")#!</p>
</cfoutput>
```

We simply use the *getSetting()* method of the *pluginConfig* object (which is available in the variables scope of your display template) to retrieve the settings that were specified via the Plugin Settings page. Now, when we add the *Greeting Message* display object to a page (as was described in the [previous post](#)), we'll see a message saying either Hello or Goodbye to whomever we specified in the settings for the plugin.

There you have it - a user-configurable plugin. Still not a very useful plugin, I'll admit, but a bit more interesting than the pervious version.

Note: The complete version of the plugin is attached to this post.