Hello World using Addon Page

This tutorial will walk you through the steps for creating an addon that will replace the **(!MAINBODY!)** (or **{body_html}** if using file-based design) of a template with the text **Hello World**. In other words, it allows you to make your addon display it's own page. It works very similarly to addon tags, the difference is that this is for working with the main part of the page. With a little modification, you can even make your addon create the whole page, instead of relying on an overall page template.

If you need your addon to display it's own page, use this tutorial as a starting point.

Requires Geo 4.0.0

The following is not possible in Geo 3.1 and earlier, because the functionality for an addon to display it's own page was added to the addon system in 4.0.

1. In the **addons/** directory, create a new directory named "**hello_world**". <
2. Inside the new directory, create a file named **info.php**. This file is what tells the Geo addon system what this particular addon can do. Use the following contents for this file:

```
<?php
//file /addons/hello_world/info.php

class addon_hello_world_info {
    var $name = 'hello_world';
    var $version = '1.0.0';
    var $title = 'Hello World';
    var $author = "Hello World, Inc.";
    var $description = 'Displays "Hello World" page.';
    var $auth_tag = 'my_addons';
    var $pages = array (
        'hello_world_page'
    );
}
```

<

1. Now, create another new file named **pages.php**. This file is what determines what is displayed on the addon page. This is the contents of pages.php[1]:

```
<?php
//file: /addons/hello_world/pages.php

class addon_hello_world_pages {
    function hello_world_page(){
        //Note: The method name matches the page name used in the $pages var
in info.php
        return "Hello World!";
    }
}
```

<

1. Now, go into addon management at Addons > Manage Addons, and you should see the addon named **Hello World**. Install and enable that addon. <
2. There should now be a link for **Edit Page** for the Hello World addon. Click on this link. <
3. On this page, select what template you want to use for your page, one template for every language you have defined, and save the changes. Remember, using this tutorial the addon will be replacing the **(!MAINBODY!)** in the template. <
4. While still on the page, look for the **Link to Page**. Click on the link to the right, and if everything was done correctly, you will see the page displayed, with the text "Hello World!" displayed on the main part of it. <

This isn't the limit of what addons can do, to see everything addons can do get the example addon (should be a free addon in the client area). It further demonstrates the use of addon pages, along with all of the other current abilities of addons.

Alternate Page Display Methods

You are not limited to returning the text you want to be displayed on the page, below are some alternate things you can do.

Display the Entire Page Yourself

At the time the method is called in the pages.php file, nothing is displayed on the page yet, even if cache is turned on or if using file-based design. This means, if you wanted to you could display the entire page yourself, by doing something like this for your **pages.php** file:

```
<?php
//file: /addons/hello_world/pages.php
//Alternate, to display the entire page

class addon_hello_world_pages {
    function hello_world_page(){
        //Note: The method name matches the page name used in the $pages var
in info.php
?>
<html>
<head>
<title>Hello World!</title>
</head>
<body><h1>Hello World!</h1>
<p>The addon is displaying the entire page now, no built in templates
involved!</p>
</body>
</html>
<?php
```

```
        //now since we just displayed the entire page, prevent the main page
from displaying:
        geoView::getInstance()->setRendered(true);

        //The return value is not used when we make the preceding call.
        return "";
    }
}
```

When you use this method, you do not need to worry about setting the template for the page as described in **step 6** of the original instructions, since this method uses no templates.

Use Smarty Templates (Recommended)

Even though this is listed as an "alternate" display method, we actually recommend doing this for anything more simple than displaying something like "Hello World!" on the page. We list this as an "alternate" display method, even though it is the one we recommend, in order to keep the main instructions as simple and easy to follow as possible.

<tip c n>**Note:** You do **not** need to switch the site to use *File-Based Design* to use Smarty templates in your addon. See DB Based VS File Based Design for more info. The part that applies most here is the last comparison point, *Addon's contents generated using..* in the linked comparison table.</tip>

For help creating Smarty templates, see the official Smarty.net website and their documentation. Have you read through the Smarty site and are ready to start using Smarty templates for your addon? Good! 😃 Then follow the instructions below.

1. Follow the original instructions at the top of the page. <
2. Create a new directory, **addons/hello_world/templates/**. This is where you will place the Smarty templates, and is the location that the template system will automatically look for any addon templates needed. <
3. Create a new file, **addons/hello_world/templates/hello_world.tpl**. This will be the template we use to display the main part of the page. For the contents of the file, use:

   ```
   {if $display_hello_world}
   <h1>Hello World!</h1>
   {else}
   <h1>Hello there!</h1>
   {/if}
   ```

   <

4. Edit the contents of the file **addons/hello_world/pages.php** and change it to be:

   ```
   <?php
   //file: /addons/hello_world/pages.php

   //extend our info class, so we can use $this-> to access vars in it
   class addon_hello_world_pages extends addon_hello_world_info {
   ```

```
    function hello_world_page(){
        //Note: The method name matches the page name used in the
$pages var in info.php
        //First, get instance of the view class
        $view = geoView::getInstance();

        //use the template file
addons/hello_world/templates/hello_world.tpl:
        $view->setBodyTpl('hello_world.tpl',$this->name);

        //example of assigning a template variable that is local in
scope to the template
        //file.  see that function for more details.
        $view->setBodyVar('display_hello_world',true);

        return "";//The templates are doing the work, so don't need to
return anything.
    }
}
```

<

1)

See Alternate Page Display Methods for some examples of variations on what you can do here.