Introduction

Getting Started

As a developer you are tasked with altering or extending the functionality of a HUBzero install or one of its extensions. You will need to be proficient in PHP and have some familiarity with such things as JavaScript or CSS. If you are new to HUBzero, this reference should help guide you through the creation of extensions such as modules and widgets (more on those later).

Thankfully, the requirements for getting started creating HUBzero extensions are minimal: knowledge of programming in PHP and a good text editor. While those are the only requirements we do, however, recommend you have working knowledge of the following:

- (X)HTML
- Cascading Stylesheets (CSS)
- JavaScript (familiarity with the MooTools 1.11 framework is a plus)
- XML
- Model-View-Controller (MVC) design pattern
- Object-Oriented Programming
Installation

Directories & File Structure

The initial directory structure of a HUBzero install.

/hubzero
   /administrator
   /cache
   /components
   /images
   /includes
   /language
   /libraries
   /logs
   /media
   /modules
   /plugins
/site
/templates
/tmp
/xmlrpc
 configuration.php
index.php
index2.php
 htaccess.txt
 robots.txt

While this looks very much like a typical Joomla! 1.5 install, there are some noticeable exceptions. Some directories vital to HUBzero functionality have been added. A quick explanation of the additional directories:

/site
   This is where HUB specific data such as member pictures, files used in wiki pages, etc. is stored.
Accessing Files

Accessing via SSH

The following tutorial should help you in using SSH to connect to and from your HUBzero server(s). You should be relatively comfortable with using a terminal (also referred to as a "command-line tool") to navigate directories and manipulate files.

**Warning:** Most accounts do not have SSH/sFTP access initially. Your system administrator must grant your account access before you will be able to connect.

From a terminal type `ssh <user>@<host>`. You will then be prompted for a password. Both the username and password will typically be the same as the account you registered on `<host>`.

```
yourmachine:~ you$ ssh username@host
yourmachine:~ you$ username@host password:
host ~
```

Windows Clients

- PuTTY (a Telnet and SSH client)

Mac OSX

All versions of Mac OSX come with Terminal.app which may be found in the /Utilities directory of your /Applications directory.

Accessing via sFTP

sFTP, or secure FTP, is a program that uses SSH to transfer files. Unlike standard FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted in the clear over the network. It is functionally similar to FTP, but because it uses a different protocol, you can't use a standard FTP client to talk to an sFTP server, nor can you connect to an FTP server with a client that supports only sFTP.

The following tutorial should help you in using sFTP to connect to and from your HUBzero server(s).

**Warning:** Most accounts do not have SSH/sFTP access initially. Your system administrator
must grant your account access before you will be able to connect.

**Graphical Clients**

Using graphical SFTP clients simplifies file transfers by allowing you to transmit files simply by dragging and dropping icons between windows. When you open the program, you will have to enter the name of the host (e.g., yourhub.org) and your HUB username and password.

**Windows Clients**

- WinSCP
- BitKinex
- FileZilla
- PuTTY

**Mac OS X Clients**

- Transmit
- Fetch
- Cyberduck
- Flow
- Fugu

**Command-line**

You can use command line SFTP from your Unix account, or from your Mac OS X or Unix workstation. To start an SFTP session, at the command prompt, enter:

```
yourmachine:~ you$ sftp username@host
yourmachine:~ you$ username@host password:
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cd</td>
<td>Change the directory on the remote computer</td>
</tr>
<tr>
<td>chmod</td>
<td>Change the permissions of files on the remote computer</td>
</tr>
<tr>
<td>chown</td>
<td>Change the owner of files on the remote computer</td>
</tr>
<tr>
<td>dir (or ls)</td>
<td>List the files in the current directory on the remote computer</td>
</tr>
<tr>
<td>exit (or quit)</td>
<td>Close the connection to the remote computer and exit SFTP</td>
</tr>
<tr>
<td>get</td>
<td>Copy a file from the remote computer to the local computer</td>
</tr>
</tbody>
</table>
INTRODUCTION

Finding Files

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help (or ?)</td>
<td>Get help on the use of SFTP commands</td>
</tr>
<tr>
<td>lcd</td>
<td>Change the directory on the local computer</td>
</tr>
<tr>
<td>lls</td>
<td>See a list of the files in the current directory on the local computer</td>
</tr>
<tr>
<td>mkdir</td>
<td>Create a directory on the local computer</td>
</tr>
<tr>
<td>ln (or symlink)</td>
<td>Create a symbolic link for a file on the remote computer</td>
</tr>
<tr>
<td>lpwd</td>
<td>Show the current directory (present working directory) on the local computer</td>
</tr>
<tr>
<td>lumask</td>
<td>Change the local umask value</td>
</tr>
<tr>
<td>mkdir</td>
<td>Create a directory on the remote computer</td>
</tr>
<tr>
<td>put</td>
<td>Copy a file from the local computer to the remote computer</td>
</tr>
<tr>
<td>pwd</td>
<td>Show the current directory (present working directory) on the remote computer</td>
</tr>
<tr>
<td>rename</td>
<td>Rename a file on the remote host</td>
</tr>
<tr>
<td>rm</td>
<td>Delete files from the remote computer</td>
</tr>
<tr>
<td>rmdir</td>
<td>Remove a directory on the remote host (the directory usually has to be empty)</td>
</tr>
<tr>
<td>version</td>
<td>Display the SFTP version</td>
</tr>
<tr>
<td>!</td>
<td>In Unix, exit to the shell prompt, where you can enter commands. Enter exit to get back to SFTP. If you follow ! with a command (e.g., !pwd), SFTP will execute the command without dropping you to the Unix prompt.</td>
</tr>
</tbody>
</table>

Once connected to a server, by either sFTP or directly with SSH, you will need to find the web root which contains the HUB install. The web root for the production version of a HUB can be found at /www/yourhub. Typically, HUBs will also have a development version of a HUB, which can be found at /www/dev.

Once in the desired directory, file layout and directory structure follows Joomla! 1.5 conventions unless otherwise noted.

See the Installation overview for details on a typical HUBzero install's directory structure.
Direct Database Access

Accessing via command-line

The following tutorial should help you in using SSH to connect to and from your HUBzero server(s) and access the database. You should be relatively comfortable with using a terminal (also referred to as a "command-line tool") to navigate directories and manipulate files.

**Warning:** Most accounts do **not** have SSH/sFTP access initially. Your system administrator must grant your account access before you will be able to connect.

See [Accessing Files](#) for further details on how to use SSH.
Libraries

Hubzero

Location:

/plugins/xhub/xlibraries

The Hubzero library contains code that is essential for a hub to run properly and altering or adding to the library without Hubzero approval is strongly discouraged.
Debugging

Joomla's Debugging Mode

To turn on Joomla!'s Debug mode:

- Login to the Joomla administration e.g. http://YOURSITE/administrator/
- At the top under the Site menu click Global Configuration.
- Click the System tab.
- Under the Debug Settings section change Debug System to Yes.
- Click the Save button.

Debug mode will output a list of all queries that were executed in order to generate the page. This will also turn on a stack trace output for error and warning pages. Hubzero components will also have PHP error reporting turned on, allowing one to see any PHP errors that may be present.

Note: Turning on debugging mode for production (live) sites is strongly discouraged and it is recommended to be avoided if at all possible.

Illegal variable ... passed to script.

One encounters the following error:

Illegal variable _files or _env or _get or _post or _cookie or _server or _session or globals passed to script.

This error is generated when the key of a key-value pair is numeric in one of the following variables: _files or _env or _get or _post or _cookie or _server or _session or globals. An example of this would be $_POST[5] = 'value'. This is most often generated by having form elements with numeric values as names. For example:

```html
<input type="text" name="5" />
```

As the error indicates, this is not allowed. Element names must include at least one non-numeric character. Examples:

```html
<input type="text" name="n5" />
```

```html
<input type="text" name="n_5" />
```