

## **Basic Setup, Gotchas, etc.**

### ***Basic Setup***

#### **Assumptions**

1. the root directory for your website has a “public\_html” directory, which contains the basic index pages, etc, that people accessing your website see.
2. there's a “lib” directory at the root of your website, which would contain libraries of code that the site uses (if not, create one, even if you're not going to use it for anything other than cs-content).
3. your website is running under the Apache webserver, under at least 1.3.x or 2.x or better.
4. You have some control over the configuration of that webserver, either directly or indirectly.

## Implementation & Keeping Up-To-Date

Really, there's a bunch of different ways of doing this, but first comes basic implementation:

1. Using a tarball of the latest release of cs-content
  1. Get the latest release from sourceforge.net (<http://sourceforge.net/projects/cs-content>).
  2. create “externals/cs-content/” in your “lib” directory, and untar the contents into that folder (so you should see “externals/cs-content/contentSystemClass.php”).
  3. New updates require destroying that directory & replacing it with the new code.
2. Using Subversion (preferred)
  1. to do this, it of course requires your website to be under active SVN (Subversion) control.
  2. Set this “svn:externals” property on your “lib” directory:
    - *externals/cs-content*     <https://svn.sourceforge.net/svnroot/cs-content/trunk>
    - you can supplement the “trunk” portion with “releases/1.0”, or whatever the latest version is: trunk is the “bleeding-edge” code, which at any one point may break your website, or have unexpected problems.
  3. When using “trunk”, simply run an “svn update” in your lib directory. Otherwise, once a new release has been made, just do an “svn switch”, with the new URL supplementing the current one.

## Gotchas

There's a variety of things that might trip-up usage of the Content System. Here's a few of them:

1. "I don't like my main page having to be '[www.mywebsite.com/content/index.php](http://www.mywebsite.com/content/index.php)'."
  - copy or create a symlink to the "content" file in your public\_html directory, called "index.php".
2. "Why would both 'templates/stuff/index.content.tmpl' AND 'templates/stuff.content.tmpl' work for the URL '/stuff.php'?"
  - It's just how the system works: cs-content doesn't care about the ".php" portion of the URL, so it just sees the "/stuff" part. Imaging creating "/stuff/index.php" and "/stuff.php" on your website...

## Basic Setup

1. In the “sample\_files” directory, move all the files into the root of your website's directory: this gives the basic amount of files needed for the Content System to do it's business.
2. Ensure that “AllowOverride All” is set for your website's root directory, so the .htaccess file can do it's job.
3. Now, make sure all the library files (like “contentSystemClass.php”) are located in “/lib/externals/cs-content/” of your website's root directory.
4. At this point, you should be able to fire up your browser, and load the website.

## Troubleshooting problems

1. If you encounter problems, like a blank page, make sure to change “display\_errors” in your php.ini file to “On”. There might be some weird option that wasn't compiled into PHP (I had issues with “posix” not being added, but that was because I'd installed from RPM, and didn't know that such things weren't in the basic PHP5).
2. If you're not running PHP5, the site won't run: it's built specifically for that version (or higher).
3. Always check the permissions of your website's directory, as bad permissions might cause Apache to not be able to access it properly (chmod 755 recursively to the root of your website's directory should suffice).