

```
<?php
```

```
/*
```

```
*-----
```

```
* APPLICATION ENVIRONMENT
```

```
*-----
```

```
*
```

```
* You can load different configurations depending on your
```

```
* current environment. Setting the environment also influences
```

```
* things like logging and error reporting.
```

```
*
```

```
* This can be set to anything, but default usage is:
```

```
*
```

```
*   development
```

```
*   testing
```

```
*   production
```

```
*
```

```
* NOTE: If you change these, also change the error_reporting() code below
```

```
*
```

```
*/
```

```
    define('ENVIRONMENT', 'development');
```

```
/*
```

```
*-----
```

```
* ERROR REPORTING
```

```
*-----
```

```
*
```

```
* Different environments will require different levels of error reporting.
```

```
* By default development will show errors but testing and live will hide them.
```

```
*/
```

```
if (defined('ENVIRONMENT'))
```

```

{
    switch (ENVIRONMENT)
    {
        case 'development':
            error_reporting(E_ALL);
            break;

        case 'testing':
        case 'production':
            error_reporting(0);
            break;

        default:
            exit('The application environment is not set correctly.');
```

```
    }
```

```
}
```

```
/*
```

```
*-----
```

```
* SYSTEM FOLDER NAME
```

```
*-----
```

```
*
```

```
* This variable must contain the name of your "system" folder.
```

```
* Include the path if the folder is not in the same directory
```

```
* as this file.
```

```
*
```

```
*/
```

```
$system_path = 'system';
```

```
/*
```

```
*-----
```

* APPLICATION FOLDER NAME

* _____

*

* If you want this front controller to use a different "application"

* folder then the default one you can set its name here. The folder

* can also be renamed or relocated anywhere on your server. If

* you do, use a full server path. For more info please see the user guide:

* http://codeigniter.com/user_guide/general/managing_apps.html

*

* NO TRAILING SLASH!

*

*/

```
$application_folder = 'application';
```

/*

* _____

* DEFAULT CONTROLLER

* _____

*

* Normally you will set your default controller in the routes.php file.

* You can, however, force a custom routing by hard-coding a

* specific controller class/function here. For most applications, you

* WILL NOT set your routing here, but it's an option for those

* special instances where you might want to override the standard

* routing in a specific front controller that shares a common CI installation.

*

* IMPORTANT: If you set the routing here, NO OTHER controller will be

* callable. In essence, this preference limits your application to ONE

* specific controller. Leave the function name blank if you need

* to call functions dynamically via the URI.

*

```

* Un-comment the $routing array below to use this feature
*
*/

    // The directory name, relative to the "controllers" folder. Leave blank
    // if your controller is not in a sub-folder within the "controllers" folder
    // $routing['directory'] = "";

    // The controller class file name. Example: Mycontroller
    // $routing['controller'] = "";

    // The controller function you wish to be called.
    // $routing['function'] = "";


/*
* -----
* CUSTOM CONFIG VALUES
* -----
*
* The $assign_to_config array below will be passed dynamically to the
* config class when initialized. This allows you to set custom config
* items or override any default config values found in the config.php file.
* This can be handy as it permits you to share one application between
* multiple front controller files, with each file containing different
* config values.
*
* Un-comment the $assign_to_config array below to use this feature
*
*/

    // $assign_to_config['name_of_config_item'] = 'value of config item';

```

```
// -----
// END OF USER CONFIGURABLE SETTINGS. DO NOT EDIT BELOW THIS LINE
// -----

/*
 * -----
 * Resolve the system path for increased reliability
 * -----
 */

// Set the current directory correctly for CLI requests
if (defined('STDIN'))
{
    chdir(dirname(__FILE__));
}

if (realpath($system_path) !== FALSE)
{
    $system_path = realpath($system_path).'/';
}

// ensure there's a trailing slash
$system_path = rtrim($system_path, '/').'/';

// Is the system path correct?
if ( ! is_dir($system_path) )
{
    exit("Your system folder path does not appear to be set correctly. Please open the
following file and correct this: ".pathinfo(__FILE__, PATHINFO_BASENAME));
}
```

```
}
```

```
/*
```

```
* -----
```

```
* Now that we know the path, set the main path constants
```

```
* -----
```

```
*/
```

```
// The name of THIS file
```

```
define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
```

```
// The PHP file extension
```

```
// this global constant is deprecated.
```

```
define('EXT', '.php');
```

```
// Path to the system folder
```

```
define('BASEPATH', str_replace("\\", "/", $system_path));
```

```
// Path to the front controller (this file)
```

```
define('FCPATH', str_replace(SELF, "", __FILE__));
```

```
// Name of the "system folder"
```

```
define('SYSDIR', trim(strrchr(trim(BASEPATH, '/'), '/'), '/'));
```

```
// The path to the "application" folder
```

```
if (is_dir($application_folder))
```

```
{
```

```
    define('APPPATH', $application_folder.'/');
```

```
}
```

```
else
```

```
{
```

```
        if ( ! is_dir(BASEPATH.$application_folder.'/'))
        {
            exit("Your application folder path does not appear to be set correctly. Please
open the following file and correct this: ".SELF);
        }

        define('APPPATH', BASEPATH.$application_folder.'/');
    }

/*
* -----
* LOAD THE BOOTSTRAP FILE
* -----
*
* And away we go...
*
*/
require_once BASEPATH.'core/CodeIgniter.php';

/* End of file index.php */
/* Location: ./index.php */
```