* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/index.html)
  + [Benchmarking Class](http://docs.google.com/benchmark.html)
  + [Caching Driver](http://docs.google.com/caching.html)
  + [Calendaring Class](http://docs.google.com/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/cart.html)
  + [Config Class](http://docs.google.com/config.html)
  + [Email Class](http://docs.google.com/email.html)
  + [Encrypt Class](http://docs.google.com/encrypt.html)
  + [Encryption Library](http://docs.google.com/encryption.html)
  + File Uploading Class
  + [Form Validation](http://docs.google.com/form_validation.html)
  + [FTP Class](http://docs.google.com/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/image_lib.html)
  + [Input Class](http://docs.google.com/input.html)
  + [Javascript Class](http://docs.google.com/javascript.html)
  + [Language Class](http://docs.google.com/language.html)
  + [Loader Class](http://docs.google.com/loader.html)
  + [Migrations Class](http://docs.google.com/migration.html)
  + [Output Class](http://docs.google.com/output.html)
  + [Pagination Class](http://docs.google.com/pagination.html)
  + [Template Parser Class](http://docs.google.com/parser.html)
  + [Security Class](http://docs.google.com/security.html)
  + [Session Library](http://docs.google.com/sessions.html)
  + [HTML Table Class](http://docs.google.com/table.html)
  + [Trackback Class](http://docs.google.com/trackback.html)
  + [Typography Class](http://docs.google.com/typography.html)
  + [Unit Testing Class](http://docs.google.com/unit_testing.html)
  + [URI Class](http://docs.google.com/uri.html)
  + [User Agent Class](http://docs.google.com/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/zip.html)
* [Database Reference](http://docs.google.com/database/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/database/examples.html)
  + [Database Configuration](http://docs.google.com/database/configuration.html)
  + [Connecting to a Database](http://docs.google.com/database/connecting.html)
  + [Running Queries](http://docs.google.com/database/queries.html)
  + [Generating Query Results](http://docs.google.com/database/results.html)
  + [Query Helper Functions](http://docs.google.com/database/helpers.html)
  + [Query Builder Class](http://docs.google.com/database/query_builder.html)
  + [Transactions](http://docs.google.com/database/transactions.html)
  + [Getting MetaData](http://docs.google.com/database/metadata.html)
  + [Custom Function Calls](http://docs.google.com/database/call_function.html)
  + [Query Caching](http://docs.google.com/database/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/database/forge.html)
  + [Database Utilities Class](http://docs.google.com/database/utilities.html)
  + [Database Driver Reference](http://docs.google.com/database/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

Toggle Table of Contents

[CodeIgniter](http://docs.google.com/index.html)

* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/index.html)
  + [Benchmarking Class](http://docs.google.com/benchmark.html)
  + [Caching Driver](http://docs.google.com/caching.html)
  + [Calendaring Class](http://docs.google.com/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/cart.html)
  + [Config Class](http://docs.google.com/config.html)
  + [Email Class](http://docs.google.com/email.html)
  + [Encrypt Class](http://docs.google.com/encrypt.html)
  + [Encryption Library](http://docs.google.com/encryption.html)
  + File Uploading Class
  + [Form Validation](http://docs.google.com/form_validation.html)
  + [FTP Class](http://docs.google.com/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/image_lib.html)
  + [Input Class](http://docs.google.com/input.html)
  + [Javascript Class](http://docs.google.com/javascript.html)
  + [Language Class](http://docs.google.com/language.html)
  + [Loader Class](http://docs.google.com/loader.html)
  + [Migrations Class](http://docs.google.com/migration.html)
  + [Output Class](http://docs.google.com/output.html)
  + [Pagination Class](http://docs.google.com/pagination.html)
  + [Template Parser Class](http://docs.google.com/parser.html)
  + [Security Class](http://docs.google.com/security.html)
  + [Session Library](http://docs.google.com/sessions.html)
  + [HTML Table Class](http://docs.google.com/table.html)
  + [Trackback Class](http://docs.google.com/trackback.html)
  + [Typography Class](http://docs.google.com/typography.html)
  + [Unit Testing Class](http://docs.google.com/unit_testing.html)
  + [URI Class](http://docs.google.com/uri.html)
  + [User Agent Class](http://docs.google.com/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/zip.html)
* [Database Reference](http://docs.google.com/database/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/database/examples.html)
  + [Database Configuration](http://docs.google.com/database/configuration.html)
  + [Connecting to a Database](http://docs.google.com/database/connecting.html)
  + [Running Queries](http://docs.google.com/database/queries.html)
  + [Generating Query Results](http://docs.google.com/database/results.html)
  + [Query Helper Functions](http://docs.google.com/database/helpers.html)
  + [Query Builder Class](http://docs.google.com/database/query_builder.html)
  + [Transactions](http://docs.google.com/database/transactions.html)
  + [Getting MetaData](http://docs.google.com/database/metadata.html)
  + [Custom Function Calls](http://docs.google.com/database/call_function.html)
  + [Query Caching](http://docs.google.com/database/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/database/forge.html)
  + [Database Utilities Class](http://docs.google.com/database/utilities.html)
  + [Database Driver Reference](http://docs.google.com/database/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

[CodeIgniter](http://docs.google.com/index.html)

* [Docs](http://docs.google.com/index.html) »
* [Libraries](http://docs.google.com/index.html) »
* File Uploading Class

classic layout

File Uploading Class[**¶**](#gjdgxs)

CodeIgniter’s File Uploading Class permits files to be uploaded. You can set various preferences, restricting the type and size of the files.

* [The Process](#30j0zll)
  + [Creating the Upload Form](#1fob9te)
  + [The Success Page](#3znysh7)
  + [The Controller](#2et92p0)
  + [The Upload Directory](#tyjcwt)
  + [Try it!](#3dy6vkm)
* [Reference Guide](#1t3h5sf)
  + [Initializing the Upload Class](#4d34og8)
  + [Setting Preferences](#2s8eyo1)
  + [Preferences](#17dp8vu)
  + [Setting preferences in a config file](#3rdcrjn)
* [Class Reference](#26in1rg)

[**The Process**](#lnxbz9)[**¶**](#30j0zll)

Uploading a file involves the following general process:

* An upload form is displayed, allowing a user to select a file and upload it.
* When the form is submitted, the file is uploaded to the destination you specify.
* Along the way, the file is validated to make sure it is allowed to be uploaded based on the preferences you set.
* Once uploaded, the user will be shown a success message.

To demonstrate this process here is brief tutorial. Afterward you’ll find reference information.

[**Creating the Upload Form**](#35nkun2)[**¶**](#1fob9te)

Using a text editor, create a form called upload\_form.php. In it, place this code and save it to your **application/views/** directory:

<html>  
<head>  
<title>Upload Form</title>  
</head>  
<body>  
  
<?php echo $error;?>  
  
<?php echo form\_open\_multipart('upload/do\_upload');?>  
  
<input type="file" name="userfile" size="20" />  
  
<br /><br />  
  
<input type="submit" value="upload" />  
  
</form>  
  
</body>  
</html>

You’ll notice we are using a form helper to create the opening form tag. File uploads require a multipart form, so the helper creates the proper syntax for you. You’ll also notice we have an $error variable. This is so we can show error messages in the event the user does something wrong.

[**The Success Page**](#1ksv4uv)[**¶**](#3znysh7)

Using a text editor, create a form called upload\_success.php. In it, place this code and save it to your **application/views/** directory:

<html>  
<head>  
<title>Upload Form</title>  
</head>  
<body>  
  
<h3>Your file was successfully uploaded!</h3>  
  
<ul>  
<?php foreach ($upload\_data as $item => $value):?>  
<li><?php echo $item;?>: <?php echo $value;?></li>  
<?php endforeach; ?>  
</ul>  
  
<p><?php echo anchor('upload', 'Upload Another File!'); ?></p>  
  
</body>  
</html>

[**The Controller**](#44sinio)[**¶**](#2et92p0)

Using a text editor, create a controller called Upload.php. In it, place this code and save it to your **application/controllers/** directory:

<?php  
  
class Upload extends CI\_Controller {  
  
 public function \_\_construct()  
 {  
 parent::\_\_construct();  
 $this->load->helper(array('form', 'url'));  
 }  
  
 public function index()  
 {  
 $this->load->view('upload\_form', array('error' => ' ' ));  
 }  
  
 public function do\_upload()  
 {  
 $config['upload\_path'] = './uploads/';  
 $config['allowed\_types'] = 'gif|jpg|png';  
 $config['max\_size'] = 100;  
 $config['max\_width'] = 1024;  
 $config['max\_height'] = 768;  
  
 $this->load->library('upload', $config);  
  
 if ( ! $this->upload->do\_upload('userfile'))  
 {  
 $error = array('error' => $this->upload->display\_errors());  
  
 $this->load->view('upload\_form', $error);  
 }  
 else  
 {  
 $data = array('upload\_data' => $this->upload->data());  
  
 $this->load->view('upload\_success', $data);  
 }  
 }  
}  
?>

[**The Upload Directory**](#2jxsxqh)[**¶**](#tyjcwt)

You’ll need a destination directory for your uploaded images. Create a directory at the root of your CodeIgniter installation called uploads and set its file permissions to 777.

[**Try it!**](#z337ya)[**¶**](#3dy6vkm)

To try your form, visit your site using a URL similar to this one:

example.com/index.php/upload/

You should see an upload form. Try uploading an image file (either a jpg, gif, or png). If the path in your controller is correct it should work.

[**Reference Guide**](#3j2qqm3)[**¶**](#1t3h5sf)

[**Initializing the Upload Class**](#1y810tw)[**¶**](#4d34og8)

Like most other classes in CodeIgniter, the Upload class is initialized in your controller using the $this->load->library() method:

$this->load->library('upload');

Once the Upload class is loaded, the object will be available using: $this->upload

[**Setting Preferences**](#4i7ojhp)[**¶**](#2s8eyo1)

Similar to other libraries, you’ll control what is allowed to be upload based on your preferences. In the controller you built above you set the following preferences:

$config['upload\_path'] = './uploads/';  
$config['allowed\_types'] = 'gif|jpg|png';  
$config['max\_size'] = '100';  
$config['max\_width'] = '1024';  
$config['max\_height'] = '768';  
  
$this->load->library('upload', $config);  
  
// Alternately you can set preferences by calling the ``initialize()`` method. Useful if you auto-load the class:  
$this->upload->initialize($config);

The above preferences should be fairly self-explanatory. Below is a table describing all available preferences.

[**Preferences**](#2xcytpi)[**¶**](#17dp8vu)

The following preferences are available. The default value indicates what will be used if you do not specify that preference.

| Preference | Default Value | Options | Description |
| --- | --- | --- | --- |
| **upload\_path** | None | None | The path to the directory where the upload should be placed. The directory must be writable and the path can be absolute or relative. |
| **allowed\_types** | None | None | The mime types corresponding to the types of files you allow to be uploaded. Usually the file extension can be used as the mime type. Can be either an array or a pipe-separated string. |
| **file\_name** | None | Desired file name | If set CodeIgniter will rename the uploaded file to this name. The extension provided in the file name must also be an allowed file type. If no extension is provided in the original file\_name will be used. |
| **file\_ext\_tolower** | FALSE | TRUE/FALSE (boolean) | If set to TRUE, the file extension will be forced to lower case |
| **overwrite** | FALSE | TRUE/FALSE (boolean) | If set to true, if a file with the same name as the one you are uploading exists, it will be overwritten. If set to false, a number will be appended to the filename if another with the same name exists. |
| **max\_size** | 0 | None | The maximum size (in kilobytes) that the file can be. Set to zero for no limit. Note: Most PHP installations have their own limit, as specified in the php.ini file. Usually 2 MB (or 2048 KB) by default. |
| **max\_width** | 0 | None | The maximum width (in pixels) that the image can be. Set to zero for no limit. |
| **max\_height** | 0 | None | The maximum height (in pixels) that the image can be. Set to zero for no limit. |
| **min\_width** | 0 | None | The minimum width (in pixels) that the image can be. Set to zero for no limit. |
| **min\_height** | 0 | None | The minimum height (in pixels) that the image can be. Set to zero for no limit. |
| **max\_filename** | 0 | None | The maximum length that a file name can be. Set to zero for no limit. |
| **max\_filename\_increment** | 100 | None | When overwrite is set to FALSE, use this to set the maximum filename increment for CodeIgniter to append to the filename. |
| **encrypt\_name** | FALSE | TRUE/FALSE (boolean) | If set to TRUE the file name will be converted to a random encrypted string. This can be useful if you would like the file saved with a name that can not be discerned by the person uploading it. |
| **remove\_spaces** | TRUE | TRUE/FALSE (boolean) | If set to TRUE, any spaces in the file name will be converted to underscores. This is recommended. |
| **detect\_mime** | TRUE | TRUE/FALSE (boolean) | If set to TRUE, a server side detection of the file type will be performed to avoid code injection attacks. DO NOT disable this option unless you have no other option as that would cause a security risk. |
| **mod\_mime\_fix** | TRUE | TRUE/FALSE (boolean) | If set to TRUE, multiple filename extensions will be suffixed with an underscore in order to avoid triggering [Apache mod\_mime](http://httpd.apache.org/docs/2.0/mod/mod_mime.html#multipleext). DO NOT turn off this option if your upload directory is public, as this is a security risk. |

[**Setting preferences in a config file**](#1ci93xb)[**¶**](#3rdcrjn)

If you prefer not to set preferences using the above method, you can instead put them into a config file. Simply create a new file called the upload.php, add the $config array in that file. Then save the file in: **config/upload.php** and it will be used automatically. You will NOT need to use the $this->upload->initialize() method if you save your preferences in a config file.

[**Class Reference**](#3whwml4)[**¶**](#26in1rg)

*class* CI\_Upload[¶](#2bn6wsx) initialize([*array $config = array()*[, *$reset = TRUE*]])[¶](#qsh70q)

| Parameters: | * **$config** (*array*) – Preferences * **$reset** (*bool*) – Whether to reset preferences (that are not provided in $config) to their defaults |
| --- | --- |
| Returns: | CI\_Upload instance (method chaining) |
| Return type: | CI\_Upload |

do\_upload([*$field = 'userfile'*])[¶](#3as4poj)

| Parameters: | * **$field** (*string*) – Name of the form field |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Performs the upload based on the preferences you’ve set.

Note

By default the upload routine expects the file to come from a form field called userfile, and the form must be of type “multipart”.

<form method="post" action="some\_action" enctype="multipart/form-data" />

If you would like to set your own field name simply pass its value to the do\_upload() method:

$field\_name = "some\_field\_name";  
$this->upload->do\_upload($field\_name);

display\_errors([*$open = '<p>'*[, *$close = '</p>'*]])[¶](#1pxezwc)

| Parameters: | * **$open** (*string*) – Opening markup * **$close** (*string*) – Closing markup |
| --- | --- |
| Returns: | Formatted error message(s) |
| Return type: | string |

Retrieves any error messages if the do\_upload() method returned false. The method does not echo automatically, it returns the data so you can assign it however you need.

**Formatting Errors**

By default the above method wraps any errors within <p> tags. You can set your own delimiters like this:

$this->upload->display\_errors('<p>', '</p>');

data([*$index = NULL*])[¶](#49x2ik5)

| Parameters: | * **$data** (*string*) – Element to return instead of the full array |
| --- | --- |
| Returns: | Information about the uploaded file |
| Return type: | mixed |

This is a helper method that returns an array containing all of the data related to the file you uploaded. Here is the array prototype:

Array  
(  
 [file\_name] => mypic.jpg  
 [file\_type] => image/jpeg  
 [file\_path] => /path/to/your/upload/  
 [full\_path] => /path/to/your/upload/jpg.jpg  
 [raw\_name] => mypic  
 [orig\_name] => mypic.jpg  
 [client\_name] => mypic.jpg  
 [file\_ext] => .jpg  
 [file\_size] => 22.2  
 [is\_image] => 1  
 [image\_width] => 800  
 [image\_height] => 600  
 [image\_type] => jpeg  
 [image\_size\_str] => width="800" height="200"  
)

To return one element from the array:

$this->upload->data('file\_name'); // Returns: mypic.jpg

Here’s a table explaining the above-displayed array items:

| Item | Description |
| --- | --- |
| file\_name | Name of the file that was uploaded, including the filename extension |
| file\_type | File MIME type identifier |
| file\_path | Absolute server path to the file |
| full\_path | Absolute server path, including the file name |
| raw\_name | File name, without the extension |
| orig\_name | Original file name. This is only useful if you use the encrypted name option. |
| client\_name | File name as supplied by the client user agent, prior to any file name preparation or incrementing |
| file\_ext | Filename extension, period included |
| file\_size | File size in kilobytes |
| is\_image | Whether the file is an image or not. 1 = image. 0 = not. |
| image\_width | Image width |
| image\_height | Image height |
| image\_type | Image type (usually the file name extension without the period) |
| image\_size\_str | A string containing the width and height (useful to put into an image tag) |

[Next](http://docs.google.com/form_validation.html)   [Previous](http://docs.google.com/encryption.html)

© Copyright 2014 - 2016, British Columbia Institute of Technology. Last updated on Oct 28, 2016.

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/snide/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org).