* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/downloads.html)
  + [Installation Instructions](http://docs.google.com/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/upgrading.html)
  + [Troubleshooting](http://docs.google.com/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/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/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/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/downloads.html)
  + [Installation Instructions](http://docs.google.com/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/upgrading.html)
  + [Troubleshooting](http://docs.google.com/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/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/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) »
* [Installation Instructions](http://docs.google.com/index.html) »
* [Upgrading From a Previous Version](http://docs.google.com/upgrading.html) »
* Upgrading from 2.2.x to 3.0.x

classic layout

Upgrading from 2.2.x to 3.0.x[**¶**](#gjdgxs)

Before performing an update you should take your site offline by replacing the index.php file with a static one.

Step 1: Update your CodeIgniter files[**¶**](#30j0zll)

**Replace** all files and directories in your *system/* directory and replace your index.php file. If any modifications were made to your index.php they will need to be made fresh in this new one.

Important

You have to delete the old *system/* directory first and then put the new one in its place. A simple copy-paste may cause issues.

Note

If you have any custom developed files in these folders please make copies of them first.

Step 2: Update your classes file names[**¶**](#1fob9te)

Starting with CodeIgniter 3.0, all class filenames (libraries, drivers, controllers and models) must be named in a Ucfirst-like manner or in other words - they must start with a capital letter.

For example, if you have the following library file:

application/libraries/mylibrary.php

... then you’ll have to rename it to:

application/libraries/Mylibrary.php

The same goes for driver libraries and extensions and/or overrides of CodeIgniter’s own libraries and core classes.

application/libraries/MY\_email.php application/core/MY\_log.php

The above files should respectively be renamed to the following:

application/libraries/MY\_Email.php application/core/MY\_Log.php

Controllers:

application/controllers/welcome.php -> application/controllers/Welcome.php

Models:

application/models/misc\_model.php -> application/models/Misc\_model.php

Please note that this DOES NOT affect directories, configuration files, views, helpers, hooks and anything else - it is only applied to classes.

You must now follow just one simple rule - class names in Ucfirst and everything else in lowercase.

Step 3: Replace config/mimes.php[**¶**](#3znysh7)

This config file has been updated to contain more user mime-types, please copy it to *application/config/mimes.php*.

Step 4: Remove $autoload[‘core’] from your config/autoload.php[**¶**](#2et92p0)

Use of the $autoload['core'] config array has been deprecated as of CodeIgniter 1.4.1 and is now removed. Move any entries that you might have listed there to $autoload['libraries'] instead.

Step 5: Move your Log class overrides or extensions[**¶**](#tyjcwt)

The Log Class is considered as a “core” class and is now located in the **system/core/** directory. Therefore, in order for your Log class overrides or extensions to work, you need to move them to **application/core/**:

application/libraries/Log.php -> application/core/Log.php application/libraries/MY\_Log.php -> application/core/MY\_Log.php

Step 6: Update your Session library usage[**¶**](#3dy6vkm)

The [*Session Library*](http://docs.google.com/libraries/sessions.html) has been completely re-written in CodeIgniter 3 and now comes with a bunch of new features, but that also means that there are changes that you should make ...

Most notably, the library now uses separate storage drivers instead of always relying on (encrypted) cookies. In fact, cookies as storage have now been removed and you must always use some kind of server-side storage engine, with the file-system being the default option.

The Session Class now utilizes PHP’s own mechanisms for building custom session handlers, which also means that your session data is now accessible via the $\_SESSION superglobal (though, we’ve kept the possibility to use it as “userdata”, like you’ve done until now).

A few configuration options have been removed and a few have been added. You should really read the whole [*Session library manual*](http://docs.google.com/libraries/sessions.html) for the details, but here’s a short list of changes that you should make:

* Set your $config['sess\_driver'] value  
  It will default to ‘files’, unles you’ve previously used $config['sess\_use\_database'], in which case it will be set to ‘database’.
* Set a $config['sess\_save\_path'] value  
  For the ‘database’ driver, a fallback to $config['sess\_table\_name'] is in place, but otherwise requires you to read the manual for the specific driver of your choice.
* Update your ci\_sessions table (‘database’ driver only)  
  The table structure has changed a bit, and more specifically:
  + session\_id field is renamed to id
  + user\_agent field is dropped
  + user\_data field is renamed to data and under MySQL is now of type BLOB
  + last\_activity field is renamed to timestamp

This is accompanied by a slight change in the table indexes too, so please read the manual about the [Session Database Driver](http://docs.google.com/libraries/sessions.html#database-driver) for more information.Important  
Only MySQL and PostgreSQL are officially supported now. Other databases may still work, but due to lack of advisory locking features, they are unsafe for concurrent requests and you should consider using another driver instead.

* Remove $config['sess\_match\_useragent']  
  The user-agent string is input supplied by the user’s browser, or in other words: client side input. As such, it is an ineffective feature and hence why it has been removed.
* Remove $config['sess\_encrypt\_cookie']  
  As already noted, the library no longer uses cookies as a storage mechanism, which renders this option useless.
* Remove $config['sess\_expire\_on\_close']  
  This option is still usable, but only for backwards compatibility purposes and it should be otherwise removed. The same effect is achieved by setting $config['sess\_expiration'] to 0.
* Check “flashdata” for collisions with “userdata”  
  Flashdata is now just regular “userdata”, only marked for deletion on the next request. In other words: you can’t have both “userdata” and “flashdata” with the same name, because it’s the same thing.
* Check usage of session metadata  
  Previously, you could access the ‘session\_id’, ‘ip\_address’, ‘user\_agent’ and ‘last\_activity’ metadata items as userdata. This is no longer possible, and you should read the notes about [Session Metadata](http://docs.google.com/libraries/sessions.html#accessing-session-metadata) if your application relies on those values.
* Check unset\_userdata() usage  
  Previously, this method used to accept an associative array of 'key' => 'dummy value' pairs for unsetting multiple keys. That however makes no sense and you now have to pass *only* the keys, as the elements of an array.  
  // Old  
  $this->session->unset\_userdata(array('item' => '', 'item2' => ''));  
    
  // New  
  $this->session->unset\_userdata(array('item', 'item2'));

Finally, if you have written a Session extension, you must now move it to the *application/libraries/Session/* directory, although chances are that it will now also have to be re-factored.

Step 7: Update your config/database.php[**¶**](#1t3h5sf)

Due to 3.0.0’s renaming of Active Record to Query Builder, inside your **config/database.php**, you will need to rename the $active\_record variable to $query\_builder:

$active\_group = 'default';  
// $active\_record = TRUE;  
$query\_builder = TRUE;

Step 8: Replace your error templates[**¶**](#4d34og8)

In CodeIgniter 3.0, the error templates are now considered as views and have been moved to the *application/views/errors* directory.

Furthermore, we’ve added support for CLI error templates in plain-text format that unlike HTML, is suitable for the command line. This of course requires another level of separation.

It is safe to move your old templates from *application/errors* to *application/views/errors/html*, but you’ll have to copy the new *application/views/errors/cli* directory from the CodeIgniter archive.

Step 9: Update your config/routes.php file[**¶**](#2s8eyo1)

Routes containing :any[**¶**](#17dp8vu)

Historically, CodeIgniter has always provided the **:any** wildcard in routing, with the intention of providing a way to match any character **within** an URI segment.

However, the **:any** wildcard is actually just an alias for a regular expression and used to be executed in that manner as **.+**. This is considered a bug, as it also matches the / (forward slash) character, which is the URI segment delimiter and that was never the intention.

In CodeIgniter 3, the **:any** wildcard will now represent **[^/]+**, so that it will not match a forward slash.

There are certainly many developers that have utilized this bug as an actual feature. If you’re one of them and want to match a forward slash, please use the **.+** regular expression:

(.+) // matches ANYTHING  
(:any) // matches any character, except for '/'

Directories and ‘default\_controller’, ‘404\_override’[**¶**](#3rdcrjn)

As you should know, the $route['default\_controller'] and $route['404\_override'] settings accept not only a controller name, but also *controller/method* pairs. However, a bug in the routing logic has made it possible for some users to use that as *directory/controller* instead.

As already said, this behavior was incidental and was never intended, nor documented. If you’ve relied on it, your application will break with CodeIgniter 3.0.

Another notable change in version 3 is that ‘default\_controller’ and ‘404\_override’ are now applied *per directory*. To explain what this means, let’s take the following example:

$route['default\_controller'] = 'main';

Now, assuming that your website is located at *example.com*, you already know that if a user visits http://example.com/, the above setting will cause your ‘Main’ controller to be loaded.

However, what happens if you have an *application/controllers/admin/* directory and the user visits http://example.com/admin/? In CodeIgniter 3, the router will look for a ‘Main’ controller under the admin/ directory as well. If not found, a Not Found (404) will be triggered.

The same rule applies to the ‘404\_override’ setting.

Step 10: Many functions now return NULL instead of FALSE on missing items[**¶**](#26in1rg)

Many methods and functions now return NULL instead of FALSE when the required items don’t exist:

* [*Common functions*](http://docs.google.com/general/common_functions.html)
  + config\_item()
* [*Config Class*](http://docs.google.com/libraries/config.html)
  + config->item()
  + config->slash\_item()
* [*Input Class*](http://docs.google.com/libraries/input.html)
  + input->get()
  + input->post()
  + input->get\_post()
  + input->cookie()
  + input->server()
  + input->input\_stream()
  + input->get\_request\_header()
* [*Session Class*](http://docs.google.com/libraries/sessions.html)
  + session->userdata()
  + session->flashdata()
* [*URI Class*](http://docs.google.com/libraries/uri.html)
  + uri->segment()
  + uri->rsegment()
* [*Array Helper*](http://docs.google.com/helpers/array_helper.html)
  + element()
  + elements()

Step 11: Usage of XSS filtering[**¶**](#lnxbz9)

Many functions in CodeIgniter allow you to use its XSS filtering feature on demand by passing a boolean parameter. The default value of that parameter used to be boolean FALSE, but it is now changed to NULL and it will be dynamically determined by your $config['global\_xss\_filtering'] value.

If you used to manually pass a boolean value for the $xss\_filter parameter or if you’ve always had $config['global\_xss\_filtering'] set to FALSE, then this change doesn’t concern you.

Otherwise however, please review your usage of the following functions:

* [*Input Library*](http://docs.google.com/libraries/input.html)
  + input->get()
  + input->post()
  + input->get\_post()
  + input->cookie()
  + input->server()
  + input->input\_stream()
* [*Cookie Helper*](http://docs.google.com/helpers/cookie_helper.html) [get\_cookie()](http://docs.google.com/helpers/cookie_helper.html#get_cookie)

Important

Another related change is that the $\_GET, $\_POST, $\_COOKIE and $\_SERVER superglobals are no longer automatically overwritten when global XSS filtering is turned on.

Step 12: Check for potential XSS issues with URIs[**¶**](#35nkun2)

The [*URI Library*](http://docs.google.com/libraries/uri.html) used to automatically convert a certain set of “programmatic characters” to HTML entities when they are encountered in a URI segment.

This was aimed at providing some automatic XSS protection, in addition to the $config['permitted\_uri\_chars'] setting, but has proven to be problematic and is now removed in CodeIgniter 3.0.

If your application has relied on this feature, you should update it to filter URI segments through $this->security->xss\_clean() whenever you output them.

Step 13: Check for usage of the ‘xss\_clean’ Form validation rule[**¶**](#1ksv4uv)

A largely unknown rule about XSS cleaning is that it should *only be applied to output*, as opposed to input data.

We’ve made that mistake ourselves with our automatic and global XSS cleaning feature (see previous step about XSS above), so now in an effort to discourage that practice, we’re also removing ‘xss\_clean’ from the officially supported list of [*form validation*](http://docs.google.com/libraries/form_validation.html) rules.

Because the [*Form Validation library*](http://docs.google.com/libraries/form_validation.html) generally validates *input* data, the ‘xss\_clean’ rule simply doesn’t belong in it.

If you really, really need to apply that rule, you should now also load the [*Security Helper*](http://docs.google.com/helpers/security_helper.html), which contains xss\_clean() as a regular function and therefore can be also used as a validation rule.

Step 14: Update usage of Input Class’s get\_post() method[**¶**](#44sinio)

Previously, the [*Input Class*](http://docs.google.com/libraries/input.html) method get\_post() was searching first in POST data, then in GET data. This method has been modified so that it searches in GET then in POST, as its name suggests.

A method has been added, post\_get(), which searches in POST then in GET, as get\_post() was doing before.

Step 15: Update usage of Directory Helper’s directory\_map() function[**¶**](#2jxsxqh)

In the resulting array, directories now end with a trailing directory separator (i.e. a slash, usually).

Step 16: Update usage of Database Forge’s drop\_table() method[**¶**](#z337ya)

Up until now, drop\_table() added an IF EXISTS clause by default or it didn’t work at all with some drivers. In CodeIgniter 3.0, the IF EXISTS condition is no longer added by default and has an optional second parameter that allows that instead and is set to FALSE by default.

If your application relies on IF EXISTS, you’ll have to change its usage.

// Now produces just DROP TABLE `table\_name`  
$this->dbforge->drop\_table('table\_name');  
  
// Produces DROP TABLE IF EXISTS `table\_name`  
$this->dbforge->drop\_table('table\_name', TRUE);

Note

The given example uses MySQL-specific syntax, but it should work across all drivers with the exception of ODBC.

Step 17: Change usage of Email library with multiple emails[**¶**](#3j2qqm3)

The [*Email Library*](http://docs.google.com/libraries/email.html) will automatically clear the set parameters after successfully sending emails. To override this behaviour, pass FALSE as the first parameter in the send() method:

if ($this->email->send(FALSE))  
{  
 // Parameters won't be cleared  
}

Step 18: Update your Form\_validation language lines[**¶**](#1y810tw)

Two improvements have been made to the [*Form Validation Library*](http://docs.google.com/libraries/form_validation.html)‘s [*language*](http://docs.google.com/libraries/language.html) files and error messages format:

* [*Language Library*](http://docs.google.com/libraries/language.html) line keys now must be prefixed with **form\_validation\_** in order to avoid collisions:  
  // Old  
  $lang['rule'] = ...  
    
  // New  
  $lang['form\_validation\_rule'] = ...
* The error messages format has been changed to use named parameters, to allow more flexibility than what sprintf() offers:  
  // Old  
  'The %s field does not match the %s field.'  
    
  // New  
  'The {field} field does not match the {param} field.'

Note

The old formatting still works, but the non-prefixed line keys are DEPRECATED and scheduled for removal in CodeIgniter 3.1+. Therefore you’re encouraged to update its usage sooner rather than later.

Step 19: Make sure your ‘base\_url’ config value is not empty[**¶**](#4i7ojhp)

When $config['base\_url'] is not set, CodeIgniter tries to automatically detect what your website’s base URL is. This is done purely for convenience when you are starting development of a new application.

Auto-detection is never reliable and also has security implications, which is why you should **always** have it manually configured!

One of the changes in CodeIgniter 3.0.3 is how this auto-detection works, and more specifically it now falls back to the server’s IP address instead of the hostname requested by the client. Therefore, if you’ve ever relied on auto-detection, it will change how your website works now.

In case you need to allow e.g. multiple domains, or both http:// and https:// prefixes to be dynamically used depending on the request, remember that *application/config/config.php* is still a PHP script, in which you can create this logic with a few lines of code. For example:

$allowed\_domains = array('domain1.tld', 'domain2.tld');  
$default\_domain = 'domain1.tld';  
  
if (in\_array($\_SERVER['HTTP\_HOST'], $allowed\_domains, TRUE))  
{  
 $domain = $\_SERVER['HTTP\_HOST'];  
}  
else  
{  
 $domain = $default\_domain;  
}  
  
if ( ! empty($\_SERVER['HTTPS']))  
{  
 $config['base\_url'] = 'https://'.$domain;  
}  
else  
{  
 $config['base\_url'] = 'http://'.$domain;  
}

Step 20: Remove usage of (previously) deprecated functionalities[**¶**](#2xcytpi)

In addition to the $autoload['core'] configuration setting, there’s a number of other functionalities that have been removed in CodeIgniter 3.0.0:

The SHA1 library[**¶**](#1ci93xb)

The previously deprecated SHA1 library has been removed, alter your code to use PHP’s native sha1() function to generate a SHA1 hash.

Additionally, the sha1() method in the [*Encrypt Library*](http://docs.google.com/libraries/encrypt.html) has been removed.

The EXT constant[**¶**](#3whwml4)

Usage of the EXT constant has been deprecated since dropping support for PHP 4. There’s no longer a need to maintain different filename extensions and in this new CodeIgniter version, the EXT constant has been removed. Use just ‘.php’ instead.

Smiley helper[**¶**](#2bn6wsx)

The [*Smiley Helper*](http://docs.google.com/helpers/smiley_helper.html) is a legacy feature from EllisLab’s ExpressionEngine product. However, it is too specific for a general purpose framework like CodeIgniter and as such it is now deprecated.

Also, the previously deprecated js\_insert\_smiley() (since version 1.7.2) is now removed.

The Encrypt library[**¶**](#qsh70q)

Following numerous vulnerability reports, the [*Encrypt Library*](http://docs.google.com/libraries/encrypt.html) has been deprecated and a new, [*Encryption Library*](http://docs.google.com/libraries/encryption.html) is added to take its place.

The new library requires either the [MCrypt extension](http://php.net/mcrypt) (and /dev/urandom availability) or PHP 5.3.3 and the [OpenSSL extension](http://php.net/openssl). While this might be rather inconvenient, it is a requirement that allows us to have properly implemented cryptographic functions.

Note

The [*Encrypt Library*](http://docs.google.com/libraries/encrypt.html) is still available for the purpose of keeping backwards compatibility.

Important

You are strongly encouraged to switch to the new [*Encryption Library*](http://docs.google.com/libraries/encryption.html) as soon as possible!

The Cart library[**¶**](#3as4poj)

The [*Cart Library*](http://docs.google.com/libraries/cart.html), similarly to the [*Smiley Helper*](http://docs.google.com/helpers/smiley_helper.html) is too specific for CodeIgniter. It is now deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

The library is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Database drivers ‘mysql’, ‘sqlite’, ‘mssql’, ‘pdo/dblib’[**¶**](#1pxezwc)

The **mysql** driver utilizes the old ‘mysql’ PHP extension, known for its aging code base and many low-level problems. The extension is deprecated as of PHP 5.5 and CodeIgniter deprecates it in version 3.0, switching the default configured MySQL driver to **mysqli**.

Please use either the ‘mysqli’ or ‘pdo/mysql’ drivers for MySQL. The old ‘mysql’ driver will be removed at some point in the future.

The **sqlite**, **mssql** and **pdo/dblib** (also known as pdo/mssql or pdo/sybase) drivers all depend on PHP extensions that for different reasons no longer exist since PHP 5.3.

Therefore we are now deprecating these drivers as we will have to remove them in one of the next CodeIgniter versions. You should use the more advanced, **sqlite3**, **sqlsrv** or **pdo/sqlsrv** drivers respectively.

Note

These drivers are still available, but you’re strongly encouraged to switch to other ones sooner rather than later.

Security helper do\_hash()[**¶**](#49x2ik5)

[*Security Helper*](http://docs.google.com/helpers/security_helper.html) function do\_hash() is now just an alias for PHP’s native hash() function. It is deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This function is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

The $config[‘global\_xss\_filtering’] setting[**¶**](#2p2csry)

As already explained above, XSS filtering should not be done on input data, but on output instead. Therefore, the $config['global\_xss\_filtering'], which automatically filters *input* data, is considered a bad practice and is now deprecated.

Instead, you should manually escape any user-provided data via the [xss\_clean()](http://docs.google.com/helpers/security_helper.html#xss_clean) function when you need to output it, or use a library like [HTML Purifier](http://htmlpurifier.org/) that does that for you.

Note

The setting is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

File helper read\_file()[**¶**](#147n2zr)

[*File Helper*](http://docs.google.com/helpers/file_helper.html) function read\_file() is now just an alias for PHP’s native file\_get\_contents() function. It is deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This function is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

String helper repeater()[**¶**](#3o7alnk)

[*String Helper*](http://docs.google.com/helpers/string_helper.html) function [repeater()](http://docs.google.com/helpers/string_helper.html#repeater) is now just an alias for PHP’s native str\_repeat() function. It is deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This function is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

String helper trim\_slashes()[**¶**](#23ckvvd)

[*String Helper*](http://docs.google.com/helpers/string_helper.html) function [trim\_slashes()](http://docs.google.com/helpers/string_helper.html#trim_slashes) is now just an alias for PHP’s native trim() function (with a slash passed as its second argument). It is deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This function is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Form helper form\_prep()[**¶**](#ihv636)

[*Form Helper*](http://docs.google.com/helpers/form_helper.html) function [form\_prep()](http://docs.google.com/helpers/form_helper.html#form_prep) is now just an alias for [*common function*](http://docs.google.com/general/common_functions.html) html\_escape(). It is deprecated and will be removed in the future.

Please use [html\_escape()](http://docs.google.com/general/common_functions.html#html_escape) instead.

Note

This function is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Email helper functions[**¶**](#32hioqz)

[*Email Helper*](http://docs.google.com/helpers/email_helper.html) only has two functions

* [valid\_email()](http://docs.google.com/helpers/email_helper.html#valid_email)
* [send\_email()](http://docs.google.com/helpers/email_helper.html#send_email)

Both of them are now aliases for PHP’s native filter\_var() and mail() functions, respectively. Therefore the [*Email Helper*](http://docs.google.com/helpers/email_helper.html) altogether is being deprecated and is scheduled for removal in CodeIgniter 3.1+.

Note

These functions are still available, but you’re strongly encouraged to remove their usage sooner rather than later.

Date helper standard\_date()[**¶**](#1hmsyys)

[*Date Helper*](http://docs.google.com/helpers/date_helper.html) function standard\_date() is being deprecated due to the availability of native PHP [constants](http://php.net/manual/en/class.datetime.php#datetime.constants.types), which when combined with date() provide the same functionality. Furthermore, they have the exact same names as the ones supported by standard\_date(). Here are examples of how to replace its usage:

// Old way  
standard\_date(); // defaults to standard\_date('DATE\_RFC822', now());  
  
// Replacement  
date(DATE\_RFC822, now());  
  
// Old way  
standard\_date('DATE\_ATOM', $time);  
  
// Replacement  
date(DATE\_ATOM, $time);

Note

This function is still available, but you’re strongly encouraged to remove its usage sooner rather than later as it is scheduled for removal in CodeIgniter 3.1+.

HTML helpers nbs(), br()[**¶**](#41mghml)

[*HTML Helper*](http://docs.google.com/helpers/html_helper.html) functions nbs() and br() are just aliases for the native str\_repeat() function used with   and <br > respectively.

Because there’s no point in just aliasing native PHP functions, they are now deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

These functions are still available, but you’re strongly encouraged to remove their usage sooner rather than later.

Pagination library ‘anchor\_class’ setting[**¶**](#2grqrue)

The [*Pagination Library*](http://docs.google.com/libraries/pagination.html) now supports adding pretty much any HTML attribute to your anchors via the ‘attributes’ configuration setting. This includes passing the ‘class’ attribute and using the separate ‘anchor\_class’ setting no longer makes sense. As a result of that, the ‘anchor\_class’ setting is now deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This setting is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

String helper random\_string() types ‘unique’ and ‘encrypt’[**¶**](#vx1227)

When using the [*String Helper*](http://docs.google.com/helpers/string_helper.html) function [random\_string()](http://docs.google.com/helpers/string_helper.html#random_string), you should no longer pass the **unique** and **encrypt** randomization types. They are only aliases for **md5** and **sha1** respectively and are now deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

These options are still available, but you’re strongly encouraged to remove their usage sooner rather than later.

URL helper url\_title() separators ‘dash’ and ‘underscore’[**¶**](#3fwokq0)

When using the [*URL Helper*](http://docs.google.com/helpers/url_helper.html) function [url\_title()](http://docs.google.com/helpers/url_helper.html#url_title), you should no longer pass **dash** or **underscore** as the word separator. This function will now accept any character and you should just pass the chosen character directly, so you should write ‘-‘ instead of ‘dash’ and ‘\_’ instead of ‘underscore’.

**dash** and **underscore** now act as aliases and are deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

These options are still available, but you’re strongly encouraged to remove their usage sooner rather than later.

Session Library method all\_userdata()[**¶**](#1v1yuxt)

As seen in the [*Change Log*](http://docs.google.com/changelog.html), [*Session Library*](http://docs.google.com/libraries/sessions.html) method userdata() now allows you to fetch all userdata by simply omitting its parameter:

$this->session->userdata();

This makes the all\_userdata() method redudant and therefore it is now just an alias for userdata() with the above shown usage and is being deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This method is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Database Forge method add\_column() with an AFTER clause[**¶**](#4f1mdlm)

If you have used the **third parameter** for [*Database Forge*](http://docs.google.com/database/forge.html) method add\_column() to add a field for an AFTER clause, then you should change its usage.

That third parameter has been deprecated and scheduled for removal in CodeIgniter 3.1+.

You should now put AFTER clause field names in the field definition array instead:

// Old usage:  
$field = array(  
 'new\_field' => array('type' => 'TEXT')  
);  
  
$this->dbforge->add\_column('table\_name', $field, 'another\_field');  
  
// New usage:  
$field = array(  
 'new\_field' => array('type' => 'TEXT', 'after' => 'another\_field')  
);  
  
$this->dbforge->add\_column('table\_name', $field);

Note

The parameter is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Note

This is for MySQL and CUBRID databases only! Other drivers don’t support this clause and will silently ignore it.

URI Routing methods fetch\_directory(), fetch\_class(), fetch\_method()[**¶**](#2u6wntf)

With properties CI\_Router::$directory, CI\_Router::$class and CI\_Router::$method being public and their respective fetch\_\*() no longer doing anything else to just return the properties - it doesn’t make sense to keep them.

Those are all internal, undocumented methods, but we’ve opted to deprecate them for now in order to maintain backwards-compatibility just in case. If some of you have utilized them, then you can now just access the properties instead:

$this->router->directory;  
$this->router->class;  
$this->router->method;

Note

Those methods are still available, but you’re strongly encouraged to remove their usage sooner rather than later.

Input library method is\_cli\_request()[**¶**](#19c6y18)

Calls to the CI\_Input::is\_cli\_request() method are necessary at many places in the CodeIgniter internals and this is often before the [*Input Library*](http://docs.google.com/libraries/input.html) is loaded. Because of that, it is being replaced by a common function named [is\_cli()](http://docs.google.com/general/common_functions.html#is_cli) and this method is now just an alias.

The new function is both available at all times for you to use and shorter to type.

// Old  
$this->input->is\_cli\_request();  
  
// New  
is\_cli();

CI\_Input::is\_cli\_request() is now now deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This method is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Config library method system\_url()[**¶**](#3tbugp1)

Usage of CI\_Config::system\_url() encourages insecure coding practices. Namely, your CodeIgniter *system/* directory shouldn’t be publicly accessible from a security point of view.

Because of this, this method is now deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This method is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

The Javascript library[**¶**](#28h4qwu)

The [*Javascript Library*](http://docs.google.com/libraries/javascript.html) has always had an ‘experimental’ status and was never really useful, nor a proper solution.

It is now deprecated and scheduled for removal in CodeIgniter 3.1+.

Note

This library is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Form Validation method prep\_for\_form()[**¶**](#nmf14n)

The [*Form Validation Library*](http://docs.google.com/libraries/form_validation.html) has a prep\_for\_form() method, which is/can also be used as a rule in set\_rules() to automatically perform HTML encoding on input data.

Automatically encoding input (instead of output) data is a bad practice in the first place, and CodeIgniter and PHP itself offer other alternatives to this method anyway. For example, [*Form Helper*](http://docs.google.com/helpers/form_helper.html) functions will automatically perform HTML escaping when necessary.

Therefore, the *prep\_for\_form* method/rule is pretty much useless and is now deprecated and scheduled for removal in 3.1+.

Note

The method is still available, but you’re strongly encouraged to remove its usage sooner rather than later.

Step 21: Check your usage of Text helper highlight\_phrase()[**¶**](#37m2jsg)

The default HTML tag used by [*Text Helper*](http://docs.google.com/helpers/text_helper.html) function highlight\_phrase() has been changed from <strong> to the new HTML5 tag <mark>.

Unless you’ve used your own highlighting tags, this might cause trouble for your visitors who use older web browsers such as Internet Explorer 8. We therefore suggest that you add the following code to your CSS files in order to avoid backwards compatibility with old browsers:

mark {  
 background: #ff0;  
 color: #000;  
};

[Next](http://docs.google.com/upgrade_223.html)   [Previous](http://docs.google.com/upgrade_301.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).