* [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](http://docs.google.com/file_uploading.html)
  + [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](#gjdgxs)
* [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](http://docs.google.com/file_uploading.html)
  + [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](#gjdgxs)
* [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) »
* Zip Encoding Class

classic layout

Zip Encoding Class[**¶**](#30j0zll)

CodeIgniter’s Zip Encoding Class permits you to create Zip archives. Archives can be downloaded to your desktop or saved to a directory.

* [Using the Zip Encoding Class](#1fob9te)
  + [Initializing the Class](#3znysh7)
  + [Usage Example](#2et92p0)
* [Class Reference](#tyjcwt)

[**Using the Zip Encoding Class**](#3dy6vkm)[**¶**](#1fob9te)

[**Initializing the Class**](#1t3h5sf)[**¶**](#3znysh7)

Like most other classes in CodeIgniter, the Zip class is initialized in your controller using the $this->load->library function:

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

Once loaded, the Zip library object will be available using:

$this->zip

[**Usage Example**](#4d34og8)[**¶**](#2et92p0)

This example demonstrates how to compress a file, save it to a folder on your server, and download it to your desktop.

$name = 'mydata1.txt';  
$data = 'A Data String!';  
  
$this->zip->add\_data($name, $data);  
  
// Write the zip file to a folder on your server. Name it "my\_backup.zip"  
$this->zip->archive('/path/to/directory/my\_backup.zip');  
  
// Download the file to your desktop. Name it "my\_backup.zip"  
$this->zip->download('my\_backup.zip');

[**Class Reference**](#2s8eyo1)[**¶**](#tyjcwt)

*class* CI\_Zip[¶](#17dp8vu) $compression\_level = 2

The compression level to use.

It can range from 0 to 9, with 9 being the highest and 0 effectively disabling compression:

$this->zip->compression\_level = 0;

add\_data(*$filepath*[, *$data = NULL*])[¶](#3rdcrjn)

|  |  |
| --- | --- |
| Parameters: | * **$filepath** (*mixed*) – A single file path or an array of file => data pairs * **$data** (*array*) – File contents (ignored if $filepath is an array) |
| Return type: | void |

Adds data to the Zip archive. Can work both in single and multiple files mode.

When adding a single file, the first parameter must contain the name you would like given to the file and the second must contain the file contents:

$name = 'mydata1.txt';  
$data = 'A Data String!';  
$this->zip->add\_data($name, $data);  
  
$name = 'mydata2.txt';  
$data = 'Another Data String!';  
$this->zip->add\_data($name, $data);

When adding multiple files, the first parameter must contain *file => contents* pairs and the second parameter is ignored:

$data = array(  
 'mydata1.txt' => 'A Data String!',  
 'mydata2.txt' => 'Another Data String!'  
);  
  
$this->zip->add\_data($data);

If you would like your compressed data organized into sub-directories, simply include the path as part of the filename(s):

$name = 'personal/my\_bio.txt';  
$data = 'I was born in an elevator...';  
  
$this->zip->add\_data($name, $data);

The above example will place my\_bio.txt inside a folder called personal.

add\_dir(*$directory*)[¶](#26in1rg)

|  |  |
| --- | --- |
| Parameters: | * **$directory** (*mixed*) – Directory name string or an array of multiple directories |
| Return type: | void |

Permits you to add a directory. Usually this method is unnecessary since you can place your data into directories when using $this->zip->add\_data(), but if you would like to create an empty directory you can do so:

$this->zip->add\_dir('myfolder'); // Creates a directory called "myfolder"

read\_file(*$path*[, *$archive\_filepath = FALSE*])[¶](#lnxbz9)

|  |  |
| --- | --- |
| Parameters: | * **$path** (*string*) – Path to file * **$archive\_filepath** (*mixed*) – New file name/path (string) or (boolean) whether to maintain the original filepath |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Permits you to compress a file that already exists somewhere on your server. Supply a file path and the zip class will read it and add it to the archive:

$path = '/path/to/photo.jpg';  
  
$this->zip->read\_file($path);  
  
// Download the file to your desktop. Name it "my\_backup.zip"  
$this->zip->download('my\_backup.zip');

If you would like the Zip archive to maintain the directory structure of the file in it, pass TRUE (boolean) in the second parameter. Example:

$path = '/path/to/photo.jpg';  
  
$this->zip->read\_file($path, TRUE);  
  
// Download the file to your desktop. Name it "my\_backup.zip"  
$this->zip->download('my\_backup.zip');

In the above example, photo.jpg will be placed into the *path/to/* directory.

You can also specify a new name (path included) for the added file on the fly:

$path = '/path/to/photo.jpg';  
$new\_path = '/new/path/some\_photo.jpg';  
  
$this->zip->read\_file($path, $new\_path);  
  
// Download ZIP archive containing /new/path/some\_photo.jpg  
$this->zip->download('my\_archive.zip');

read\_dir(*$path*[, *$preserve\_filepath = TRUE*[, *$root\_path = NULL*]])[¶](#35nkun2)

|  |  |
| --- | --- |
| Parameters: | * **$path** (*string*) – Path to directory * **$preserve\_filepath** (*bool*) – Whether to maintain the original path * **$root\_path** (*string*) – Part of the path to exclude from the archive directory |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Permits you to compress a directory (and its contents) that already exists somewhere on your server. Supply a path to the directory and the zip class will recursively read and recreate it as a Zip archive. All files contained within the supplied path will be encoded, as will any sub-directories contained within it. Example:

$path = '/path/to/your/directory/';  
  
$this->zip->read\_dir($path);  
  
// Download the file to your desktop. Name it "my\_backup.zip"  
$this->zip->download('my\_backup.zip');

By default the Zip archive will place all directories listed in the first parameter inside the zip. If you want the tree preceding the target directory to be ignored, you can pass FALSE (boolean) in the second parameter. Example:

$path = '/path/to/your/directory/';  
  
$this->zip->read\_dir($path, FALSE);

This will create a ZIP with a directory named “directory” inside, then all sub-directories stored correctly inside that, but will not include the */path/to/your* part of the path.

archive(*$filepath*)[¶](#1ksv4uv)

|  |  |
| --- | --- |
| Parameters: | * **$filepath** (*string*) – Path to target zip archive |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Writes the Zip-encoded file to a directory on your server. Submit a valid server path ending in the file name. Make sure the directory is writable (755 is usually OK). Example:

$this->zip->archive('/path/to/folder/myarchive.zip'); // Creates a file named myarchive.zip

download(*$filename = 'backup.zip'*)[¶](#44sinio)

|  |  |
| --- | --- |
| Parameters: | * **$filename** (*string*) – Archive file name |
| Return type: | void |

Causes the Zip file to be downloaded from your server. You must pass the name you would like the zip file called. Example:

$this->zip->download('latest\_stuff.zip'); // File will be named "latest\_stuff.zip"

Note

Do not display any data in the controller in which you call this method since it sends various server headers that cause the download to happen and the file to be treated as binary.

get\_zip()[¶](#2jxsxqh)

|  |  |
| --- | --- |
| Returns: | Zip file content |
| Return type: | string |

Returns the Zip-compressed file data. Generally you will not need this method unless you want to do something unique with the data. Example:

$name = 'my\_bio.txt';  
$data = 'I was born in an elevator...';  
  
$this->zip->add\_data($name, $data);  
  
$zip\_file = $this->zip->get\_zip();

clear\_data()[¶](#z337ya)

|  |  |
| --- | --- |
| Return type: | void |

The Zip class caches your zip data so that it doesn’t need to recompile the Zip archive for each method you use above. If, however, you need to create multiple Zip archives, each with different data, you can clear the cache between calls. Example:

$name = 'my\_bio.txt';  
$data = 'I was born in an elevator...';  
  
$this->zip->add\_data($name, $data);  
$zip\_file = $this->zip->get\_zip();  
  
$this->zip->clear\_data();  
  
$name = 'photo.jpg';  
$this->zip->read\_file("/path/to/photo.jpg"); // Read the file's contents  
  
$this->zip->download('myphotos.zip');

[Next](http://docs.google.com/database/index.html)   [Previous](http://docs.google.com/xmlrpc.html)

© Copyright 2014 - 2018, British Columbia Institute of Technology. Last updated on Jun 12, 2018.

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).