* [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
  + [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](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
  + [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](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) »
* Email Class

classic layout

Email Class[**¶**](#gjdgxs)

CodeIgniter’s robust Email Class supports the following features:

* Multiple Protocols: Mail, Sendmail, and SMTP
* TLS and SSL Encryption for SMTP
* Multiple recipients
* CC and BCCs
* HTML or Plaintext email
* Attachments
* Word wrapping
* Priorities
* BCC Batch Mode, enabling large email lists to be broken into small BCC batches.
* Email Debugging tools
* [Using the Email Library](#30j0zll)
  + [Sending Email](#1fob9te)
  + [Setting Email Preferences](#3znysh7)
    - [Setting Email Preferences in a Config File](#2et92p0)
  + [Email Preferences](#tyjcwt)
  + [Overriding Word Wrapping](#3dy6vkm)
* [Class Reference](#1t3h5sf)

[**Using the Email Library**](#4d34og8)[**¶**](#30j0zll)

[**Sending Email**](#2s8eyo1)[**¶**](#1fob9te)

Sending email is not only simple, but you can configure it on the fly or set your preferences in a config file.

Here is a basic example demonstrating how you might send email. Note: This example assumes you are sending the email from one of your [*controllers*](http://docs.google.com/general/controllers.html).

$this->load->library('email');  
  
$this->email->from('your@example.com', 'Your Name');  
$this->email->to('someone@example.com');  
$this->email->cc('another@another-example.com');  
$this->email->bcc('them@their-example.com');  
  
$this->email->subject('Email Test');  
$this->email->message('Testing the email class.');  
  
$this->email->send();

[**Setting Email Preferences**](#17dp8vu)[**¶**](#3znysh7)

There are 21 different preferences available to tailor how your email messages are sent. You can either set them manually as described here, or automatically via preferences stored in your config file, described below:

Preferences are set by passing an array of preference values to the email initialize method. Here is an example of how you might set some preferences:

$config['protocol'] = 'sendmail';  
$config['mailpath'] = '/usr/sbin/sendmail';  
$config['charset'] = 'iso-8859-1';  
$config['wordwrap'] = TRUE;  
  
$this->email->initialize($config);

Note

Most of the preferences have default values that will be used if you do not set them.

[**Setting Email Preferences in a Config File**](#3rdcrjn)[**¶**](#2et92p0)

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 email.php, add the $config array in that file. Then save the file at config/email.php and it will be used automatically. You will NOT need to use the $this->email->initialize() method if you save your preferences in a config file.

[**Email Preferences**](#26in1rg)[**¶**](#tyjcwt)

The following is a list of all the preferences that can be set when sending email.

| Preference | Default Value | Options | Description |
| --- | --- | --- | --- |
| **useragent** | CodeIgniter | None | The “user agent”. |
| **protocol** | mail | mail, sendmail, or smtp | The mail sending protocol. |
| **mailpath** | /usr/sbin/sendmail | None | The server path to Sendmail. |
| **smtp\_host** | No Default | None | SMTP Server Address. |
| **smtp\_user** | No Default | None | SMTP Username. |
| **smtp\_pass** | No Default | None | SMTP Password. |
| **smtp\_port** | 25 | None | SMTP Port. |
| **smtp\_timeout** | 5 | None | SMTP Timeout (in seconds). |
| **smtp\_keepalive** | FALSE | TRUE or FALSE (boolean) | Enable persistent SMTP connections. |
| **smtp\_crypto** | No Default | tls or ssl | SMTP Encryption |
| **wordwrap** | TRUE | TRUE or FALSE (boolean) | Enable word-wrap. |
| **wrapchars** | 76 |  | Character count to wrap at. |
| **mailtype** | text | text or html | Type of mail. If you send HTML email you must send it as a complete web page. Make sure you don’t have any relative links or relative image paths otherwise they will not work. |
| **charset** | $config['charset'] |  | Character set (utf-8, iso-8859-1, etc.). |
| **validate** | FALSE | TRUE or FALSE (boolean) | Whether to validate the email address. |
| **priority** | 3 | 1, 2, 3, 4, 5 | Email Priority. 1 = highest. 5 = lowest. 3 = normal. |
| **crlf** | \n | “\r\n” or “\n” or “\r” | Newline character. (Use “\r\n” to comply with RFC 822). |
| **newline** | \n | “\r\n” or “\n” or “\r” | Newline character. (Use “\r\n” to comply with RFC 822). |
| **bcc\_batch\_mode** | FALSE | TRUE or FALSE (boolean) | Enable BCC Batch Mode. |
| **bcc\_batch\_size** | 200 | None | Number of emails in each BCC batch. |
| **dsn** | FALSE | TRUE or FALSE (boolean) | Enable notify message from server |

[**Overriding Word Wrapping**](#lnxbz9)[**¶**](#3dy6vkm)

If you have word wrapping enabled (recommended to comply with RFC 822) and you have a very long link in your email it can get wrapped too, causing it to become un-clickable by the person receiving it. CodeIgniter lets you manually override word wrapping within part of your message like this:

The text of your email that  
gets wrapped normally.  
  
{unwrap}http://example.com/a\_long\_link\_that\_should\_not\_be\_wrapped.html{/unwrap}  
  
More text that will be  
wrapped normally.

Place the item you do not want word-wrapped between: {unwrap} {/unwrap}

[**Class Reference**](#35nkun2)[**¶**](#1t3h5sf)

*class* CI\_Email[¶](#1ksv4uv) from(*$from*[, *$name = ''*[, *$return\_path = NULL*]])[¶](#44sinio)

| Parameters: | * **$from** (*string*) – “From” e-mail address * **$name** (*string*) – “From” display name * **$return\_path** (*string*) – Optional email address to redirect undelivered e-mail to |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the email address and name of the person sending the email:

$this->email->from('you@example.com', 'Your Name');

You can also set a Return-Path, to help redirect undelivered mail:

$this->email->from('you@example.com', 'Your Name', 'returned\_emails@example.com');

Note

Return-Path can’t be used if you’ve configured ‘smtp’ as your protocol.

reply\_to(*$replyto*[, *$name = ''*])[¶](#2jxsxqh)

| Parameters: | * **$replyto** (*string*) – E-mail address for replies * **$name** (*string*) – Display name for the reply-to e-mail address |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the reply-to address. If the information is not provided the information in the :meth:from method is used. Example:

$this->email->reply\_to('you@example.com', 'Your Name');

to(*$to*)[¶](#z337ya)

| Parameters: | * **$to** (*mixed*) – Comma-delimited string or an array of e-mail addresses |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the email address(s) of the recipient(s). Can be a single e-mail, a comma-delimited list or an array:

$this->email->to('someone@example.com');

$this->email->to('one@example.com, two@example.com, three@example.com');

$this->email->to(  
 array('one@example.com', 'two@example.com', 'three@example.com')  
);

cc(*$cc*)[¶](#3j2qqm3)

| Parameters: | * **$cc** (*mixed*) – Comma-delimited string or an array of e-mail addresses |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the CC email address(s). Just like the “to”, can be a single e-mail, a comma-delimited list or an array.

bcc(*$bcc*[, *$limit = ''*])[¶](#1y810tw)

| Parameters: | * **$bcc** (*mixed*) – Comma-delimited string or an array of e-mail addresses * **$limit** (*int*) – Maximum number of e-mails to send per batch |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the BCC email address(s). Just like the to() method, can be a single e-mail, a comma-delimited list or an array.

If $limit is set, “batch mode” will be enabled, which will send the emails to batches, with each batch not exceeding the specified $limit.

subject(*$subject*)[¶](#4i7ojhp)

| Parameters: | * **$subject** (*string*) – E-mail subject line |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the email subject:

$this->email->subject('This is my subject');

message(*$body*)[¶](#2xcytpi)

| Parameters: | * **$body** (*string*) – E-mail message body |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the e-mail message body:

$this->email->message('This is my message');

set\_alt\_message(*$str*)[¶](#1ci93xb)

| Parameters: | * **$str** (*string*) – Alternative e-mail message body |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Sets the alternative e-mail message body:

$this->email->set\_alt\_message('This is the alternative message');

This is an optional message string which can be used if you send HTML formatted email. It lets you specify an alternative message with no HTML formatting which is added to the header string for people who do not accept HTML email. If you do not set your own message CodeIgniter will extract the message from your HTML email and strip the tags.

set\_header(*$header*, *$value*)[¶](#3whwml4)

| Parameters: | * **$header** (*string*) – Header name * **$value** (*string*) – Header value |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Appends additional headers to the e-mail:

$this->email->set\_header('Header1', 'Value1');  
$this->email->set\_header('Header2', 'Value2');

clear([*$clear\_attachments = FALSE*])[¶](#2bn6wsx)

| Parameters: | * **$clear\_attachments** (*bool*) – Whether or not to clear attachments |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Initializes all the email variables to an empty state. This method is intended for use if you run the email sending method in a loop, permitting the data to be reset between cycles.

foreach ($list as $name => $address)  
{  
 $this->email->clear();  
  
 $this->email->to($address);  
 $this->email->from('your@example.com');  
 $this->email->subject('Here is your info '.$name);  
 $this->email->message('Hi '.$name.' Here is the info you requested.');  
 $this->email->send();  
}

If you set the parameter to TRUE any attachments will be cleared as well:

$this->email->clear(TRUE);

send([*$auto\_clear = TRUE*])[¶](#qsh70q)

| Parameters: | * **$auto\_clear** (*bool*) – Whether to clear message data automatically |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

The e-mail sending method. Returns boolean TRUE or FALSE based on success or failure, enabling it to be used conditionally:

if ( ! $this->email->send())  
{  
 // Generate error  
}

This method will automatically clear all parameters if the request was successful. To stop this behaviour pass FALSE:

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

Note

In order to use the print\_debugger() method, you need to avoid clearing the email parameters.

attach(*$filename*[, *$disposition = ''*[, *$newname = NULL*[, *$mime = ''*]]])[¶](#3as4poj)

| Parameters: | * **$filename** (*string*) – File name * **$disposition** (*string*) – ‘disposition’ of the attachment. Most email clients make their own decision regardless of the MIME specification used here. <https://www.iana.org/assignments/cont-disp/cont-disp.xhtml> * **$newname** (*string*) – Custom file name to use in the e-mail * **$mime** (*string*) – MIME type to use (useful for buffered data) |
| --- | --- |
| Returns: | CI\_Email instance (method chaining) |
| Return type: | CI\_Email |

Enables you to send an attachment. Put the file path/name in the first parameter. For multiple attachments use the method multiple times. For example:

$this->email->attach('/path/to/photo1.jpg');  
$this->email->attach('/path/to/photo2.jpg');  
$this->email->attach('/path/to/photo3.jpg');

To use the default disposition (attachment), leave the second parameter blank, otherwise use a custom disposition:

$this->email->attach('image.jpg', 'inline');

You can also use a URL:

$this->email->attach('http://example.com/filename.pdf');

If you’d like to use a custom file name, you can use the third paramater:

$this->email->attach('filename.pdf', 'attachment', 'report.pdf');

If you need to use a buffer string instead of a real - physical - file you can use the first parameter as buffer, the third parameter as file name and the fourth parameter as mime-type:

$this->email->attach($buffer, 'attachment', 'report.pdf', 'application/pdf');

attachment\_cid(*$filename*)[¶](#1pxezwc)

| Parameters: | * **$filename** (*string*) – Existing attachment filename |
| --- | --- |
| Returns: | Attachment Content-ID or FALSE if not found |
| Return type: | string |

Sets and returns an attachment’s Content-ID, which enables your to embed an inline (picture) attachment into HTML. First parameter must be the already attached file name.

$filename = '/img/photo1.jpg';  
$this->email->attach($filename);  
foreach ($list as $address)  
{  
 $this->email->to($address);  
 $cid = $this->email->attachment\_cid($filename);  
 $this->email->message('<img src="cid:'. $cid .'" alt="photo1" />');  
 $this->email->send();  
}

Note

Content-ID for each e-mail must be re-created for it to be unique.

print\_debugger([*$include = array('headers'*, *'subject'*, *'body')*])[¶](#49x2ik5)

| Parameters: | * **$include** (*array*) – Which parts of the message to print out |
| --- | --- |
| Returns: | Formatted debug data |
| Return type: | string |

Returns a string containing any server messages, the email headers, and the email messsage. Useful for debugging.

You can optionally specify which parts of the message should be printed. Valid options are: **headers**, **subject**, **body**.

Example:

// You need to pass FALSE while sending in order for the email data  
// to not be cleared - if that happens, print\_debugger() would have  
// nothing to output.  
$this->email->send(FALSE);  
  
// Will only print the email headers, excluding the message subject and body  
$this->email->print\_debugger(array('headers'));

Note

By default, all of the raw data will be printed.

[Next](http://docs.google.com/encrypt.html)   [Previous](http://docs.google.com/config.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).