



The Handbook

Codex

http://codex.wordpress.org/Pluggable_Functions

Version Date

9 August 2005

Pluggable Functions

Pluggable functions in WordPress 1.5.1+ let you replace certain core functions with functions of your own using a plugin. A function can only be reassigned this way once, so you can't install two plugins that plug the same function for different reasons.

Reference

The default functions are defined in `wp-includes/pluggable-functions.php`.

`get_currentuserinfo()`

Grabs the information of the current logged in user, if there is one. Essentially a wrapper for `get_userdata()`, but it also stores information in global variables.

`get_userdata($userid)`

Pulls user information for the specified user from the database.

`get_userdata_by_login($user_login)`

Pulls user information for the specified user from the database.

`wp_mail($to, $subject, $message, $headers = '')`

A convenient wrapper for PHP's mail function.

`wp_login($username, $password, $already_md5 = false)`

Returns `true` if the specified username and password correspond to a registered user.

`auth_redirect()`

If a user is not logged in, he or she will be redirected to WordPress' login page before being allowed to access content on the page from which this function was called.

Upon successfully logging in, the user is sent back to the page in question.

`wp_redirect($location)`

Redirects a browser to the *absolute URI* specified by the `$location` parameter.

`wp_setcookie($username, $password, $already_md5 = false, $home = '', $siteurl = '')`

Sets the WordPress cookies for a logged in user. See [WordPress Cookies](#).

`wp_clearcookie()`

Clears the cookies for a logged in user. See [WordPress Cookies](#).

`wp_notify_postauthor($comment_id, $comment_type='')`

Emails the author of the comment's post the content of the comment specified.

`wp_notify_moderator($comment_id)`

Informs the administrative email account that the comment specified needs to be moderated. See [General Options SubPanel](#).

Example

An example of what you can do with a pluggable function is replace the default email handler. To do this, you'd need to write a plugin that defines a `wp_mail()` function. The default `wp_mail()` function looks like this:

```
function wp_mail($to, $subject, $message, $headers = '') {  
    if( $headers == '' ) {  
        $headers = "MIME-Version: 1.0\n" .
```

```

        "From: " . get_settings('admin_email') . "\n" .
        "Content-Type: text/plain; charset=\"" . get_settings('blog_charset') .
"\n\n";
    }

    return @mail($to, $subject, $message, $headers);
}

```

But, for example, if you wanted to CC all your mail to another address, you could use this code in a plugin:

```

function wp_mail($to, $subject, $message, $headers = '') {
    if( $headers == '' ) {
        $headers = "MIME-Version: 1.0\n" .
        "From: " . get_settings('admin_email') . "\n" .
        "Cc: dummy@example.com\n";
        "Content-Type: text/plain; charset=\"" . get_settings('blog_charset') .
"\n\n";
    }

    return @mail($to, $subject, $message, $headers);
}

```

Notice that if you plug a core function like this the original is no longer available. I.e., the elegant solution here would have been to write a function that tacks our Cc header on the end of the existing `$headers` string then call the original `wp_mail()` with the extra header. However this would not work as the original `wp_mail()` does not exist if you plug it.