Filter Reference

- Post, Page, and Attachment (Upload) Filters
  - Database Reads
  - Database Writes
- Comment, Trackback, and Ping Filters
  - Database Reads
  - Database Writes
- Category and Term Filters
  - Database Reads
  - Database Writes
- Link Filters
- Date and Time Filters
- Author and User Filters
  - Database Reads
  - Database Writes
- Blogroll Filters
- Blog Information and Option Filters
- General Text Filters
- Administrative Filters
- Rich Text Editor Filters
- Template Filters
  - Kubrick Filters
- Registration & Login Filters
- Redirect/Rewrite Filters
- WP_Query Filters
- Media Filters
- Advanced WordPress Filters
- Widgets
- Admin Bar

- Further Reading

This article contains an extensive (but not 100% comprehensive) list of the filter hooks available for use in plugin development in Version 2.1 and above of WordPress. For more information:

- To learn more about what filter and action hooks are, see Plugin API.
- To learn about writing plugins in general, see Writing a Plugin.
- For a reference list of action hooks, see Plugin API/Action Reference.
- For an automatically-generated list of *all* WordPress hooks, see the WordPress Hooks Database

Note: If you want to add to or clarify this documentation, please follow the style of the existing entries. Describe what data the filter is applied to, and if the filter function takes additional arguments, describe the argument list.

Post, Page, and Attachment (Upload) Filters

See also #Category and Term Filters, #Author and User Filters, #Link Filters, #Date and Time Filters, and #Administrative Filters below.

Database Reads

Filters in this section are applied to information read from the database, prior to displaying on a page or editing screen.

attachment_fields_to_edit : applied to the form fields to be displayed when editing an attachment. Called in the `get_attachment_fields_to_edit` function. Filter function arguments: an array of form fields, the post object.

attachment_icon : applied to the icon for an attachment in the `get_attachment_icon` function. Filter function arguments: icon file as an HTML IMG tag, attachment ID.

attachment_innerHTML : applied to the title to be used for an attachment if there is no icon, in the `get_attachment_innerHTML` function. Filter function arguments: inner HTML (defaults to the title), attachment ID.

author_edit_pre : applied to post author prior to display for editing.

body_class : applied to the classes for the HTML

content_edit_pre : applied to post content prior to display for editing.

content_filtered_edit_pre : applied to post content filtered prior to display for editing.

excerpt_edit_pre : applied to post excerpt prior to display for editing.

date_edit_pre : applied to post date prior to display for editing.

date_gmt_edit_pre : applied to post date prior to display for editing.

get_attached_file : applied to the attached file information retrieved by the `get_attached_file` function. Filter function arguments: file information, attachment ID.

get_enclosed : applied to the enclosures list for a post by the `get_enclosed` function.

get_pages : applied to the list of pages returned by the `get_pages` function. Filter function arguments: list of pages (each item of which contains a page data array), `get_pages` function argument list (telling which pages were requested).

get_pung : applied to the list of pinged URLs for a post by the `get_pung` function.

get_the_archive_title : applied to the archive's title in the `get_the_archive_title` function.

get_the_excerpt : applied to the post's excerpt in the `get_the_excerpt` function.

get_the_guid : applied to the post's GUID in the `get_the_guid` function.

get_to_ping : applied to the list of URLs to ping for a post by the `get_to_ping` function.

icon_dir : applied to the template's image directory in several functions. Basically allows a plugin to specify that icons for MIME types should come from a different location.

icon_dir_uri : applied to the template's image directory URI in several functions. Basically allows a plugin to specify that icons for MIME types should come from a different location.

image_size_names_choose : applied to the list of image sizes selectable in the Media Library. Commonly used to make custom image sizes selectable.

mime_type_edit_pre : applied to post mime type prior to display for editing.

modified_edit_pre : applied to post modification date prior to display for editing.

modified_gmt_edit_pre : applied to post modification gmt date prior to display for editing.

no_texturize_shortcodes : applied to registered shortcodes. Can be used to exempt shortcodes from the automatic texturize function.

parent_edit_pre : applied to post parent id prior to display for editing.

password_edit_pre : applied to post password prior to display for editing.

post_class : applied to the classes of the outermost HTML element for a post. Called in the `get_post_class` function. Filter function arguments: an array of class names, an array of additional class names that were added to the first array, and the post ID.

pre_kses : applied to various content prior to being processed/sanitized by KSES. This hook allows developers to customize what types of scripts/tags should either be allowed in content or stripped.

prepend_attachment : applied to the HTML to be prepended by the `prepend_attachment` function.

protected_title_format : Used to the change or manipulate the post title when the post is password protected.

private_title_format : Used to the change or manipulate the post title when its status is private.

sanitize_title : applied to a post title by the `sanitize_title` function, after stripping out HTML tags.

single_post_title : applied to the post title when used to create a blog page title by the `wp_title` and `single_post_title` functions.

status_edit_pre : applied to post status prior to display for editing.

the_content : applied to the post content retrieved from the database, prior to printing on the screen (also used in some other operations, such as trackbacks).

the_content_rss : applied to the post content prior to including in an RSS feed. (Deprecated)

the_content_feed : applied to the post content prior to including in an RSS feed.

the_editor_content : applied to post content before putting it into a rich editor window.

the_excerpt : applied to the post excerpt (or post content, if there is no excerpt) retrieved from the database, prior to printing on the screen (also used in some other operations, such as trackbacks).

the_excerpt_rss : applied to the post excerpt prior to including in an RSS feed.

the_password_form : applied to the password form for protected posts.

the_tags : applied to the tags retrieved from the database, prior to printing on the screen.

the_title : applied to the post title retrieved from the database, prior to printing on the screen (also used in some other operations, such as trackbacks).

the_title_rss : applied to the post title before including in an RSS feed (after first filtering with `the_title`.

title_edit_pre : applied to post title prior to display for editing.

type_edit_pre : applied to post type prior to display for editing.

wp_dropdown_pages : applied to the HTML dropdown list of WordPress pages generated by the `wp_dropdown_pages` function.

wp_list_pages : applied to the HTML list generated by the `wp_list_pages` function.

wp_list_pages_excludes : applied to the list of excluded pages (an array of page IDs) in the `wp_list_pages` function.

wp_get_attachment_metadata : applied to the attachment metadata retrieved by the `wp_get_attachment_metadata` function. Filter function arguments: meta data, attachment ID.

wp_get_attachment_thumb_file : applied to the attachment thumbnail file retrieved by the `wp_get_attachment_thumb_file` function. Filter function arguments: thumbnail file, attachment ID.

wp_get_attachment_thumb_url : applied to the attachment thumbnail URL retrieved by the `wp_get_attachment_thumb_URL` function. Filter function arguments: thumbnail URL, attachment ID.

wp_get_attachment_url : applied to the attachment URL retrieved by the `wp_get_attachment_url` function. Filter function arguments: URL, attachment ID.

wp_mime_type_icon : applied to the MIME type icon for an attachment calculated by the `wp_mime_type_icon` function. Filter function arguments: icon URI calculated, MIME type, post ID.

wp_title : applied to the blog page title before sending to the browser in the `wp_title` function.

Database Writes

Filters in this section are applied to information prior to saving to the database.

add_ping : applied to the new value of the pinged field on a post when a ping is added, prior to saving the new information in the database.

attachment_fields_to_save : applied to fields associated with an attachment prior to saving them in the database. Called in the `media_upload_form_handler` function. Filter function arguments: an array of post attributes, an array of attachment fields including the changes submitted from the form.

attachment_max_dims : applied to the maximum image dimensions before reducing an image size. Filter function input (and return value) is either false (if no maximum dimensions have been specified) or a two-item list (width, height).

category_save_pre : applied to post category comma-separated list prior to saving it in the database (also used for attachments).

comment_status_pre : applied to post comment status prior to saving it in the database (also used for attachments).

content_filtered_save_pre : applied to filtered post content prior to saving it in the database (also used for attachments).

content_save_pre : applied to post content prior to saving it in the database (also used for attachments).

excerpt_save_pre : applied to post excerpt prior to saving it in the database (also used for attachments).

image_save_pre (deprecated) : use `image_editor_save_pre` instead.

jpeg_quality (deprecated) : use `wp_editor_set_quality` or `WP_Image_Editor::set_quality()` instead.

name_save_pre (Deprecated): applied to post name prior to saving it in the database (also used for attachments).

phone_content : applied to the content of a post submitted by email, before saving.

ping_status_pre : applied to post ping status prior to saving it in the database (also used for attachments).

post_mime_type_pre : applied to the MIME type for an attachment prior to saving it in the database.

status_save_pre : applied to post status prior to saving it in the database.

thumbnail_filename : applied to the file name for the thumbnail when uploading an image.

title_save_pre : applied to post title prior to saving it in the database (also used for attachments).

update_attached_file : applied to the attachment information prior to saving in post metadata in the `update_attached_file` function. Filter function arguments: attachment information, attachment ID.

wp_create_thumbnail (deprecated)

wp_delete_file : applied to an attachment file name just before deleting.

wp_generate_attachment_metadata : applied to the attachment metadata array before saving in the database.

wp_save_image_file (deprecated) : use `wp_save_image_editor_file` instead.

wp_thumbnail_creation_size_limit : applied to the size of the thumbnail when uploading an image. Filter function arguments: max file size, attachment ID, attachment file name.

wp_thumbnail_max_side_length : applied to the size of the thumbnail when uploading an image. Filter function arguments: image side max size, attachment ID, attachment file name.

wp_update_attachment_metadata : applied to the attachment metadata just before saving in the `wp_update_attachment_metadata` function. Filter function arguments: meta data, attachment ID.

Comment, Trackback, and Ping Filters

See also #Author and User Filters, #Link Filters, #Date and Time Filters, and #Administrative Filters below.

Database Reads

Filters in this section are applied to information read from the database, prior to displaying on a page or editing screen.

comment_excerpt : applied to the comment excerpt by the `comment_excerpt` function. See also `get_comment_excerpt`.

comment_flood_filter : applied when someone appears to be flooding your blog with comments. Filter function arguments: already blocked (true/false, whether a previous filtering plugin has already blocked it; set to true and return true to block this comment in a plugin), time of previous comment, time of current comment.

comment_post_redirect : applied to the redirect location after someone adds a comment. Filter function arguments: redirect location, comment info array.

comment_text : applied to the comment text before displaying on the screen by the `comment_text` function, and in the admin menus.

comment_text_rss : applied to the comment text prior to including in an RSS feed.

comments_array : applied to the array of comments for a post in the `comments_template` function. Filter function arguments: array of comment information structures, post ID.

comments_number : applied to the formatted text giving the number of comments generated by the `comments_number` function. See also `get_comments_number`.

get_comment_excerpt : applied to the comment excerpt read from the database by the `get_comment_excerpt` function (which is also called by `comment_excerpt`. See also `comment_excerpt`.

get_comment_ID : applied to the comment ID read from the global `$comments` variable by the `get_comment_ID` function.

get_comment_text : applied to the comment text of the current comment in the `get_comment_text` function, which is also called by the `comment_text` function.

get_comment_type : applied to the comment type (“comment”, “trackback”, or “pingback”) by the `get_comment_type` function (which is also called by `comment_type`).

get_comments_number : applied to the comment count read from the `$post` global variable by the `get_comments_number` function (which is also called by the `comments_number` function; see also `comments_number` filter).

post_comments_feed_link : applied to the feed URL generated for the comments feed by the `comments_rss` function.

Database Writes

Filters in this section are applied to information prior to saving to the database.

comment_save_pre : applied to the comment data just prior to updating/editing comment data. Function arguments: comment data array, with indices “comment_post_ID”, “comment_author”, “comment_author_email”, “comment_author_url”, “comment_content”, “comment_type”, and “user_ID”.

pre_comment_approved : applied to the current comment’s approval status (true/false) to allow a plugin to override. Return true/false and set first argument to true/false to approve/disapprove the comment, and use global variables such as `$comment_ID` to access information about this comment.

pre_comment_content : applied to the content of a comment prior to saving the comment in the database.

preprocess_comment : applied to the comment data prior to any other processing, when saving a new comment in the database. Function arguments: comment data array, with indices “comment_post_ID”, “comment_author”, “comment_author_email”, “comment_author_url”, “comment_content”, “comment_type”, and “user_ID”.

wp_insert_post_data : applied to modified and unmodified post data in wp_insert_post() prior to update or insertion of post into database. Function arguments: modified and extended post array and sanitized post array.

Category and Term Filters

See also #Administrative Filters below.

Database Reads

Filters in this section are applied to information read from the database, prior to displaying on a page or editing screen.

category_description : applied to the “description” field categories by the `category_description` and `wp_list_categories` functions. Filter function arguments: description, category ID when called from `category_description`; description, category information array (all fields from the category table for that particular category) when called from `wp_list_categories`.

category_feed_link : applied to the feed URL generated for the category feed by the `get_category_feed_link` function.

category_link : applied to the URL created for a category by the `get_category_link` function. Filter function arguments: link URL, category ID.

get_ancestors : applied to the list of ancestor IDs returned by the `get_ancestors` function (which is in turn used by many other functions). Filter function arguments: ancestor IDs array, given object ID, given object type.

get_categories : applied to the category list generated by the `get_categories` function (which is in turn used by many other functions). Filter function arguments: category list, `get_categories` options list.

get_category : applied to the category information that the `get_category` function looks up, which is basically an array of all the fields in WordPress’s category table for a particular category ID.

list_cats : called for two different purposes:

1. the `wp_dropdown_categories` function uses it to filter the `show_option_all` and `show_option_none` arguments (which are used to put options “All” and “None” in category drop-down lists). No additional filter function arguments.
2. the `wp_list_categories` function applies it to the category names. Filter function arguments: category name, category information list (all fields from the category table for that particular category).

list_cats_exclusions : applied to the SQL WHERE statement giving the categories to be excluded by the `get_categories` function. Typically, a plugin would add to this list, in order to exclude certain categories or groups of categories from category lists. Filter function arguments: excluded category WHERE clause, `get_categories` options list.

single_cat_title : applied to the category name when used to create a blog page title by the `wp_title` and `single_cat_title` functions.

the_category : applied to the list of categories (an HTML list with links) created by the `get_the_category_list` function. Filter function arguments: generated HTML text, list separator being used (empty string means it is a default LI list), `parents` argument to `get_the_category_list`.

the_category_rss : applied to the category list (a list of category XML elements) for a post by the `get_the_category_rss` function, before including in an RSS feed. Filter function arguments are the list text and the type (“rdf” or “rss” generally).

wp_dropdown_cats : applied to the drop-down category list (a text string containing HTML option elements) generated by the `wp_dropdown_categories` function.

wp_list_categories : applied to the category list (an HTML list) generated by the `wp_list_categories` function.

wp_get_object_terms : applied to the list of terms (an array of objects) generated by the `wp_get_object_terms` function, which is called by a number of category/term related functions, such as `get_the_terms` and `get_the_category`.

Database Writes

Filters in this section are applied to information prior to saving to the database.

pre_category_description : applied to the category desription prior to saving in the database.

wp_update_term_parent : filter term parent before update to term is applied, hook to this filter to see if it will cause a hierarchy loop.

edit_terms : (actually an action, but often used like a filter) hooked in prior to saving taxonomy/category change in the database

pre_category_name : applied to the category name prior to saving in the database.

pre_category_nicename : applied to the category nice name prior to saving in the database.

Link Filters

Note: This section contains filters related to links to posts, pages, archives, feeds, etc. For blogroll links, see the #Blogroll Filters section below.

attachment_link : applied to the calculated attachment permalink by the `get_attachment_link` function. Filter function arguments: link URL, attachment ID.

author_feed_link : applied to the feed URL generated for the author feed by the `get_author_rss_link` function.

author_link : applied to the author's archive permalink created by the `get_author_posts_url` function. Filter function arguments: link URL, author ID, author's "nice" name. Note that `get_author_posts_url` is called within functions `wp_list_authors` and `the_author_posts_link`.

comment_reply_link : applied to the link generated for replying to a specific comment by the `get_comment_reply_link` function which is called within function `comments_template`. Filter function arguments: link (string), custom options (array), current comment (object), current post (object).

day_link : applied to the link URL for a daily archive by the `get_day_link` function. Filter function arguments: URL, year, month number, day number.

feed_link : applied to the link URL for a feed by the `get_feed_link` function. Filter function arguments: URL, type of feed (e.g. "rss2", "atom", etc.).

get_comment_author_link : applied to the HTML generated for the author's link on a comment, in the `get_comment_author_link` function (which is also called by `comment_author_link`. Action function arguments: user name

get_comment_author_url_link : applied to the HTML generated for the author's link on a comment, in the `get_comment_author_url_link` function (which is also called by `comment_author_link`).

month_link : applied to the link URL for a monthly archive by the `get_month_link` function. Filter function arguments: URL, year, month number.

page_link : applied to the calculated page URL by the `get_page_link` function. Filter function arguments: URL, page ID. Note that there is also an internal filter called `_get_page_link` that can be used to filter the URLS of pages that are not designated as the blog's home page (same arguments). Note that this only applies to WordPress pages, not posts, custom post types, or attachments.

post_link : applied to the calculated post permalink by the `get_permalink` function, which is also called by the `the_permalink`, `post_permalink`, `previous_post_link`, and `next_post_link` functions. Filter function arguments: permalink URL, post data list. Note that this only applies to WordPress default posts, and not custom post types (nor pages or attachments).

post_type_link : applied to the calculated custom post type permalink by the `get_post_permalink` function.

the_permalink : applied to the permalink URL for a post prior to printing by function `the_permalink`.

year_link : applied to the link URL for a yearly archive by the `get_year_link` function. Filter function arguments: URL, year.

tag_link : applied to the URL created for a tag by the get_tag_link function. Filter function arguments: link URL, tag ID.

term_link : applied to the URL created for a term by the get_term_link function. Filter function arguments: term link URL, term object and taxonomy slug.

Date and Time Filters

See also #Link Filters above.

get_comment_date : applied to the formatted comment date generated by the `get_comment_date` function (which is also called by `comment_date`).

get_comment_time : applied to the formatted comment time in the `get_comment_time` function (which is also called by `comment_time`).

get_the_modified_date : applied to the formatted post modification date generated by the `get_the_modified_date` function (which is also called by the `the_modified_date` function).

get_the_modified_time : applied to the formatted post modification time generated by the `get_the_modified_time` and `get_post_modified_time` functions (which are also called by the `the_modified_time` function).

get_the_time : applied to the formatted post time generated by the `get_the_time` and `get_post_time` functions (which are also called by the `the_time` function).

the_date : applied to the formatted post date generated by the `the_date` function.

the_modified_date : applied to the formatted post modification date generated by the `the_modified_date` function.

the_modified_time : applied to the formatted post modification time generated by the `the_modified_time` function.

the_time : applied to the formatted post time generated by the `the_time` function.

the_weekday : applied to the post date weekday name generated by the `the_weekday` function.

the_weekday_date : applied to the post date weekday name generated by the `the_weekday_date` function. Function arguments are the weekday name, before text, and after text (before text and after text are added to the weekday name if the current post's weekday is different from the previous post's weekday).

Author and User Filters

See also #Link Filters and #Administrative Filters sections.

login_body_class : Allows filtering of the body class applied to the login screen in `login_header()`.

login_redirect : applied to the `redirect_to` post/get variable during the user login process.

user_contactmethods : applied to the contact methods fields on the user profile page. (old page is here: contactmethods)

update_(meta_type)_metadata : applied before a (user) metadata gets updated.

Database Reads

Filters in this section are applied to information read from the database, prior to displaying on a page or editing screen.

author_email : applied to the comment author's email address retrieved from the database by the `comment_author_email` function. See also `get_comment_author_email`.

comment_author : applied to the comment author's name retrieved from the database by the `comment_author` function. See also `get_comment_author`.

comment_author_rss : applied to the comment author's name prior to including in an RSS feed.

comment_email : applied to the comment author's email address retrieved from the database by the `comment_author_email_link` function.

comment_url : applied to the comment author's URL retrieved from the database by the `comment_author_url` function (see also `get_comment_author_url`).

get_comment_author : applied to the comment author's name retrieved from the database by `get_comment_author`, which is also called by `comment_author`. See also `comment_author`.

get_comment_author_email : applied to the comment author's email address retrieved from the database by `get_comment_author_email`, which is also called by `comment_author_email`. See also `author_email`.

get_comment_author_IP : applied to the comment author's IP address retrieved from the database by the `get_comment_author_IP` function, which is also called by `comment_author_IP`.

get_comment_author_url : applied to the comment author's URL retrieved from the database by the `get_comment_author_url` function, which is also called by `comment_author_url`. See also `comment_url`.

login_errors : applied to the login error message printed on the login screen.

login_headertitle : applied to the title for the login header URL (Powered by WordPress by default) printed on the login screen.

login_headerurl : applied to the login header URL (points to wordpress.org by default) printed on the login screen.

login_message : applied to the login message printed on the login screen.

role_has_cap : applied to a role's capabilities list in the `WP_Role->has_cap` function. Filter function arguments are the capabilities list to be filtered, the capability being questioned, and the role's name.

sanitize_user : applied to a user name by the `sanitize_user` function. Filter function arguments: user name (after some cleaning up), raw user name, strict (true or false to use strict ASCII or not).

the_author : applied to a post author's displayed name by the `get_the_author` function, which is also called by the `the_author` function.

the_author_email : applied to a post author's email address by the `the_author_email` function.

user_search_columns : applied to the list of columns in the wp_users table to include in the `WHERE` clause inside WP_User_Query.

Database Writes

Filters in this section are applied to information prior to saving to the database.

pre_comment_author_email : applied to a comment author's email address prior to saving the comment in the database.

pre_comment_author_name : applied to a comment author's user name prior to saving the comment in the database.

pre_comment_author_url : applied to a comment author's URL prior to saving the comment in the database.

pre_comment_user_agent : applied to the comment author's user agent prior to saving the comment in the database.

pre_comment_user_ip : applied to the comment author's IP address prior to saving the comment in the database.

pre_user_id : applied to the comment author's user ID prior to saving the comment in the database.

pre_user_description : applied to the user's description prior to saving in the database.

pre_user_display_name : applied to the user's displayed name prior to saving in the database.

pre_user_email : applied to the user's email address prior to saving in the database.

pre_user_first_name : applied to the user's first name prior to saving in the database.

pre_user_last_name : applied to the user's last name prior to saving in the database.

pre_user_login : applied to the user's login name prior to saving in the database.

pre_user_nicename : applied to the user's "nice name" prior to saving in the database.

pre_user_nickname : applied to the user's nickname prior to saving in the database.

pre_user_url : applied to the user's URL prior to saving in the database.

registration_errors : applied to the list of registration errors generated while registering a user for a new account.

user_registration_email : applied to the user's email address read from the registration page, prior to trying to register the person as a new user.

validate_username : applied to the validation result on a new user name. Filter function arguments: valid (true/false), user name being validated.

Blogroll Filters

Note: This section contains filters related to blogroll links. For filters related to links to posts, pages, categories, etc., see section #Link Filters above.

get_bookmarks : applied to link/blogroll database query results by the `get_bookmarks` function. Filter function arguments: database query results list, `get_bookmarks` arguments list.

link_category : applied to the link category by the `get_links_list` and `wp_list_bookmarks` functions (as of WordPress 2.2).

link_description : applied to the link description by the `get_links` and `wp_list_bookmarks` functions (as of WordPress 2.2).

link_rating : applied to the link rating number by the `get_linkrating` function.

link_title : applied to the link title by the `get_links` and `wp_list_bookmarks` functions (as of WordPress 2.2)

pre_link_description : applied to the link description prior to saving in the database.

pre_link_image : applied to the link image prior to saving in the database.

pre_link_name : applied to the link name prior to saving in the database.

pre_link_notes : applied to the link notes prior to saving in the database.

pre_link_rel : applied to the link relation information prior to saving in the database.

pre_link_rss : applied to the link RSS URL prior to saving in the database.

pre_link_target : applied to the link target information prior to saving in the database.

pre_link_url : applied to the link URL prior to saving in the database.

Blog Information and Option Filters

all_options : applied to the option list retrieved from the database by the `get_alloptions` function.

all_plugins : applied to the list of plugins retrieved for display in the plugins list table.

bloginfo : applied to the blog option information retrieved from the database by the `bloginfo` function, after first retrieving the information with the `get_bloginfo` function. A second argument `$show` gives the name of the bloginfo option that was requested. Note that `bloginfo("url")`, `bloginfo("directory")` and `bloginfo("home")` do *not* use this filtering function (see `bloginfo_url`).

bloginfo_rss : applied to the blog option information by function `get_bloginfo_rss` (which is also called from `bloginfo_rss`), after first retrieving the information with the `get_bloginfo` function, stripping out HTML tags, and converting characters appropriately. A second argument `$show` gives the name of the bloginfo option that was requested.

bloginfo_url : applied to the the output of `bloginfo("url")`, `bloginfo("directory")` and `bloginfo("home")` before returning the information.

loginout : applied to the HTML link for logging in and out (generally placed in the sidebar) generated by the `wp_loginout` function.

lostpassword_url : applied to the URL that allows users to reset their passwords.

option_(option name) : applied to the option value retrieved from the database by the `get_option` function, after unserializing (which decodes array-based options). To use this filter, you will need to add filters for specific options names, such as “option_foo” to filter the output of `get_option("foo")`.

pre_get_space_used: applied to the `get_space_used()` function to provide an alternative way of displaying storage space used. Returning false from this filter will revert to default display behavior (used wp_upload_dir() directory space in megabytes).

pre_option_(option name) : applied to the option value retrieved from the database by the `get_alloptions` function, after unserializing (which decodes array-based options). To use this filter, you will need to add filters for specific options names, such as “pre_option_foo” to filter the option “foo”.

pre_update_option_(option name) : applied the option value before being saving to the database to allow overriding the value to be stored. To use this filter, you will need to add filters for specific options names, such as “pre_update_option_foo” to filter the option “foo”.

register : applied to the sidebar link created for the user to register (if allowed) or visit the admin panels (if already logged in) by the `wp_register` function.

upload_dir : applied to the directory to be used for uploads calculated by the `wp_upload_dir` function. Filter function argument is an array with components “dir” (the upload directory path), “url” (the URL of the upload directory), and “error” (which you can set to true if you want to generate an error).

upload_mimes : allows a filter function to return a list of MIME types for uploads, if there is no MIME list input to the `wp_check_filetype` function. Filter function argument is an associated list of MIME types whose component names are file extensions (separated by vertical bars) and values are the corresponding MIME types.

General Text Filters

attribute_escape : applied to post text and other content by the `attribute_escape` function, which is called in many places in WordPress to change certain characters into HTML attributes before sending to the browser.

js_escape : applied to JavaScript code before sending to the browser in the `js_escape` function.

sanitize_key : applied to key before using it for your settings, field, or other needs, generated by `sanitize_key` function

Administrative Filters

The filters in this section are related to the administration screens of WordPress, including content editing screens.

admin_user_info_links : applied to the user profile and info links in the WordPress admin quick menu.

autosave_interval : applied to the interval for auto-saving posts.

bulk_actions : applied to an array of bulk items in admin bulk action dropdowns.

bulk_post_updated_messages : applied to an array of bulk action updated messages.

cat_rows : applied to the category rows HTML generated for managing categories in the admin menus.

comment_edit_pre : applied to comment content prior to display in the editing screen.

comment_edit_redirect : applied to the redirect location after someone edits a comment in the admin menus. Filter function arguments: redirect location, comment ID.

comment_moderation_subject : applied to the mail subject before sending email notifying the administrator of the need to moderate a new comment. Filter function arguments: mail subject, comment ID..

comment_moderation_text : applied to the body of the mail message before sending email notifying the administrator of the need to moderate a new comment. Filter function arguments: mail body text, comment ID.

comment_notification_headers : applied to the mail headers before sending email notifying the post author of a new comment. Filter function arguments: mail header text, comment ID.

comment_notification_subject : applied to the mail subject before sending email notifying the post author of a new comment. Filter function arguments: mail subject, comment ID.

comment_notification_text : applied to the body of the mail message before sending email notifying the post author of a new comment. Filter function arguments: mail body text, comment ID.

comment_row_actions : applied to the list of action links under each comment row (like Reply, Quick Edit, Edit).

cron_request : Allows filtering of the URL, key and arguments passed to wp_remote_post() in spawn_cron().

cron_schedules : applied to an empty array to allow a plugin to generate cron schedules in the `wp_get_schedules` function.

custom_menu_order : used to activate the ‘menu_order’ filter.

default_content : applied to the default post content prior to opening the editor for a new post.

default_excerpt : applied to the default post excerpt prior to opening the editor for a new post.

default_title : applied to the default post title prior to opening the editor for a new post.

editable_slug : applied to the post, page, tag or category slug by the `get_sample_permalink` function.

format_to_edit : applied to post content, excerpt, title, and password by the `format_to_edit` function, which is called by the admin menus to set up a post for editing. Also applied to when editing comments in the admin menus.

format_to_post : applied to post content by the `format_to_post` function, which is not used in WordPress by default.

manage_edit-${post_type}_columns : applied to the list of columns to print on the manage posts screen for a custom post type. Filter function argument/return value is an associative array where the element key is the name of the column, and the value is the header text for that column. See also action `manage_${post_type}_posts_custom_column`, which puts the column information into the edit screen.

manage_link-manager_columns : was `manage_link_columns` until wordpress 2.7. applied to the list of columns to print on the blogroll management screen. Filter function argument/return value is an associative list where the element key is the name of the column, and the value is the header text for that column. See also action `manage_link_custom_column`, which puts the column information into the edit screen.

manage_posts_columns : applied to the list of columns to print on the manage posts screen. Filter function argument/return value is an associative array where the element key is the name of the column, and the value is the header text for that column. See also action `manage_posts_custom_column`, which puts the column information into the edit screen. (see Scompt's tutorial for examples and use.)

manage_pages_columns : applied to the list of columns to print on the manage pages screen. Filter function argument/return value is an associative array where the element key is the name of the column, and the value is the header text for that column. See also action `manage_pages_custom_column`, which puts the column information into the edit screen.

manage_users_columns

manage_users_custom_column

manage_users_sortable_columns

media_row_actions : applied to the list of action links under each file in the Media Library (like View, Edit).

menu_order : applied to the array for the admin menu order. Must be activated with the ‘custom_menu_order’ filter before.

nonce_life : applied to the lifespan of a nonce to generate or verify the nonce. Can be used to generate nonces which expire earlier. The value returned by the filter should be in seconds.

nonce_user_logged_out : applied to the current user ID used to generate or verify a nonce when the user is logged out.

plugin_row_meta : add additional links below each plugin on the plugins page.

postmeta_form_limit : applied to the number of post-meta information items shown on the post edit screen.

post_row_actions : applied to the list of action links (like Quick Edit, Edit, View, Preview) under each post in the Posts > All Posts section.

post_updated_messages : applied to the array storing user-visible administrative messages when working with posts, pages and custom post types. This filter is used to change the text of said messages, not to trigger them. See “customizing the messages” in the register_post_type documentation.

pre_upload_error : applied to allow a plugin to create an XMLRPC error for uploading files.

preview_page_link : applied to the link on the page editing screen that shows the page preview at the bottom of the screen.

preview_post_link : applied to the link on the post editing screen that shows the post preview at the bottom of the screen.

richedit_pre : applied to post content by the `wp_richedit_pre` function, before displaying in the rich text editor.

schedule_event : applied to each recurring and single event as it is added to the cron schedule.

set-screen-option : Filter a screen option value before it is set.

show_password_fields : applied to the true/false variable that controls whether the user is presented with the opportunity to change their password on the user profile screen (true means to show password changing fields; false means don’t).

terms_to_edit : applied to the CSV of terms (for each taxonomy) that is used to show which terms are attached to the post.

the_editor : applied to the HTML DIV created to house the rich text editor, prior to printing it on the screen. Filter function argument/return value is a string.

user_can_richedit : applied to the calculation of whether the user's browser has rich editing capabilities, and whether the user wants to use the rich editor, in the `user_can_richedit` function. Filter function argument and return value is true/false if the current user can/cannot use the rich editor.

user_has_cap : applied to a user's capabilities list in the `WP_User->has_cap` function (which is called by the `current_user_can` function). Filter function arguments are the capabilities list to be filtered, the capability being questioned, and the argument list (which has things such as the post ID if the capability is to edit posts, etc.)

wp_handle_upload_prefilter : applied to the upload information when uploading a file. Filter function argument: array which represents a single element of $_FILES.

wp_handle_upload : applied to the upload information when uploading a file. Filter function argument: array with elements "file" (file name), "url", "type".

wp_revisions_to_keep : alters how many revisions are kept for a given post. Filter function arguments: number representing desired revisions saved (default is unlimited revisions), the post object.

wp_terms_checklist_args : applied to arguments of the wp_terms_checklist() function. Filter function argument: array of checklist arguments, post ID.

wp_upload_tabs : applied to the list of custom tabs to display on the upload management admin screen. Use action `upload_files_(tab)` to display a page for your custom tab.

media_upload_tabs : applied to the list of custom tabs to display on the upload management admin screen. Use action `upload_files_(tab)` to display a page for your custom tab.

plugin_action_links_(plugin file name) : applied to the list of links to display on the plugins page (beside the activate/deactivate links).

views_edit-post : applied to the list posts eg All (30) | Published (22) | Draft (5) | Pending (2) | Trash (1)

Rich Text Editor Filters

These filters modify the configuration of the rich text editor, TinyMCE.

mce_spellchecker_languages : applied to the language selection available in the spell checker.

mce_buttons, mce_buttons_2, mce_buttons_3, mce_buttons_4 : applied to the rows of buttons for the rich editor toolbar (each is an array of button names).

mce_css : applied to the CSS file URL for the rich text editor.

mce_external_plugins : applied to the array of external plugins to be loaded by the rich text editor.

mce_external_languages : applied to the array of language files loaded by external plugins, allowing them to use the standard translation method (see tinymce/langs/wp-langs.php for reference).

tiny_mce_before_init : applied to the whole init array for the editor.

Template Filters

This section contains links related to themes, templates, and style files.

locale_stylesheet_uri : applied to the locale-specific stylesheet URI returned by the `get_locale_stylesheet_uri` function. Filter function arguments: URI, stylesheet directory URI.

stylesheet : applied to the stylesheet returned by the `get_stylesheet` function.

stylesheet_directory : applied to the stylesheet directory returned by the `get_stylesheet_directory` function. Filter function arguments: stylesheet directory, stylesheet.

stylesheet_directory_uri : applied to the stylesheet directory URI returned by the `get_stylesheet_directory_uri` function. Filter function arguments: stylesheet directory URI, stylesheet.

stylesheet_uri : applied to the stylesheet URI returned by the `get_stylesheet_uri` function. Filter function arguments: stylesheet URI, stylesheet.

template : applied to the template returned by the `get_template` function.

template_directory : applied to the template directory returned by the `get_template_directory` function. Filter function arguments: template directory, template.

template_directory_uri : applied to the template directory URI returned by the `get_template_directory_uri` function. Filter function arguments: template directory URI, template.

theme_root : applied to the theme root directory (normally wp-content/themes) returned by the `get_theme_root` function.

theme_root_uri : applied to the theme root directory URI returned by the `get_theme_root_uri` function. Filter function arguments: URI, site URL.

You can also replace individual template files from your theme, by using the following filter hooks. See also the `template_redirect` action hook. Each of these filters takes as input the path to the corresponding template file in the current theme. A plugin can modify the file to be used by returning a new path to a template file.

404_template :

archive_template : You can use this for example to enforce a specific template for a custom post type archive. This way you can keep all the code in a plugin.

attachment_template :

author_template :

category_template :

comments_popup_template :

comments_template : The “comments_template” filter can be used to load a custom template form a plugin which replace the themes default comment template.

date_template :

home_template :

page_template :

paged_template :

search_template :

single_template : You can use this for example to enforce a specific template for a custom post type. This way you can keep all the code in a plugin.

shortcut_link : applied to the “Press This” bookmarklet link.

template_include :

wp_nav_menu_args : applied to the arguments of the `wp_nav_menu` function.

wp_nav_menu_items : Filter the HTML list content for navigation menus.

Kubrick Filters

These filters were present in the pre-3.0 default theme kubrick.

kubrick_header_color : applied to the color for the header of the kubrick theme.

kubrick_header_display : applied to the display option for the header of the kubrick theme.

kubrick_header_image : applied to the header image file for the kubrick theme.

Registration & Login Filters

authenticate : allows basic authentication to be performed on login based on username and password.

registration_errors : applied to the list of registration errors generated while registering a user for a new account.

user_registration_email : applied to the user's email address read from the registration page, prior to trying to register the person as a new user.

validate_username : applied to the validation result on a new user name. Filter function arguments: valid (true/false), user name being validated.

wp_authenticate_user : applied when a user attempted to log in, after WordPress validates username and password, but before validation errors are checked.

Redirect/Rewrite Filters

These advanced filters relate to WordPress's handling of rewrite rules.

allowed_redirect_hosts : applied to the list of host names deemed safe for redirection. wp-login.php uses this to defend against a dangerous 'redirect_to' request parameter

author_rewrite_rules : applied to the author-related rewrite rules after they are generated.

category_rewrite_rules : applied to the category-related rewrite rules after they are generated.

comments_rewrite_rules : applied to the comment-related rewrite rules after they are generated.

date_rewrite_rules : applied to the date-related rewrite rules after they are generated.

mod_rewrite_rules : applied to the list of rewrite rules given to the user to put into their .htaccess file when they change their permalink structure. (Note: replaces deprecated filter `rewrite_rules`.)

page_rewrite_rules : applied to the page-related rewrite rules after they are generated.

post_rewrite_rules : applied to the post-related rewrite rules after they are generated.

redirect_canonical : Can be used to cancel a “canonical” URL redirect. Accepts 2 parameters: `$redirect_url`, `$requested_url`. To cancel the redirect return `FALSE`, to allow the redirect return `$redirect_url`

rewrite_rules_array : applied to the entire rewrite rules array after it is generated.

root_rewrite_rules : applied to the root-level rewrite rules after they are generated.

search_rewrite_rules : applied to the search-related rewrite rules after they are generated.

wp_redirect : applied to a redirect URL by the default `wp_redirect` function. Filter function arguments: URL, HTTP status code.

wp_redirect_status : applied to the HTTP status code when redirecting by the default `wp_redirect` function. Filter function arguments: HTTP status code, URL.

WP_Query Filters

These are filters run by the WP_Query object in the course of building

and executing a query to retrieve posts. See also #Advanced WordPress Filters for queries relating to users, meta values, and more generic queries.

found_posts : applied to the list of posts, just after querying from the database.

found_posts_query : after the list of posts to display is queried from the database, WordPress selects rows in the query results. This filter allows you to do something other than `SELECT FOUND_ROWS()` at that step.

post_limits : applied to the `LIMIT` clause of the query that returns the post array.

posts_clauses : applied to the entire SQL query, divided into a keyed array for each clause type, that returns the post array. Can be easier to work with than `posts_request`.

posts_distinct : allows a plugin to add a `DISTINCTROW` clause to the query that returns the post array.

posts_fields : applied to the field list for the query that returns the post array.

posts_groupby : applied to the `GROUP BY` clause of the query that returns the post array (normally empty).

posts_join : applied to the `JOIN` clause of the query that returns the post array. This is typically used to add a table to the `JOIN`, in combination with the `posts_where` filter.

posts_join_paged : applied to the `JOIN` clause of the query that returns the post array, after the paging is calculated (though paging does not affect the JOIN, so this is actually equivalent to `posts_join`).

posts_orderby : applied to the `ORDER BY` clause of the query that returns the post array.

posts_request : applied to the entire SQL query that returns the post array, just prior to running the query.

posts_results : allows you to manipulate the resulting array returned from the query.

posts_search : applied to the search SQL that is used in the `WHERE` clause of WP_Query.

posts_where : applied to the `WHERE` clause of the query that returns the post array.

posts_where_paged : applied to the `WHERE` clause of the query that returns the post array, after the paging is calculated (though paging does not affect the WHERE, so this is actually equivalent to `posts_where`).

the_posts : applied to the list of posts queried from the database after minimal processing for permissions and draft status on single-post pages.

Media Filters

This section contains media filters that are used to integrated different types of media.

editor_max_image_size :

image_downsize :

get_image_tag_class :

get_image_tag :

image_resize_dimensions :

intermediate_image_sizes :

icon_dir :

wp_get_attachment_image_attributes :

img_caption_shortcode :

post_gallery :

use_default_gallery_style :

gallery_style :

(adjacent)_image_link :

embed_defaults :

load_default_embeds :

embed_oembed_html :

embed_googlevideo :

oembed_result :

upload_size_limit :

wp_image_editors :

plupload_default_settings :

plupload_default_params :

image_size_names_choose :

wp_prepare_attachment_for_js :

media_upload_tabs :

disable_captions :

media_view_settings :

media_view_strings :

wp_handle_upload_prefilter :

Advanced WordPress Filters

This section contains advanced filters related to internationalization, miscellaneous queries, and other fundamental WordPress functions.

create_user_query : applied to the query used to save a new user's information to the database, just prior to running the query.

get_editable_authors : applied to the list of post authors that the current user is authorized to edit in the `get_editable_authors` function.

get_next_post_join : in function `get_next_post` (which finds the post after the currently-displayed post), applied to the SQL JOIN clause (which normally joins to the category table if user is viewing a category archive). Filter function arguments: JOIN clause, stay in same category (true/false), list of excluded categories.

get_next_post_sort : in function `get_next_post` (which finds the post after the currently-displayed post), applied to the SQL ORDER BY clause (which normally orders by post date in ascending order

with a limit of 1 post). Filter function arguments: ORDER BY clause.

get_next_post_where : in function `get_next_post` (which finds the post after the currently-displayed post), applied to the SQL WHERE clause (which normally looks for the next dated published post). Filter function arguments: WHERE clause, stay in same category (true/false), list of excluded categories.

get_previous_post_join : in function `get_previous_post` (which finds the post before the currently-displayed post), applied to the SQL JOIN clause (which normally joins to the category table if user is viewing a category archive). Filter function arguments: join clause, stay in same category (true/false), list of excluded categories.

get_previous_post_sort : in function `get_previous_post` (which finds the post before the currently-displayed post), applied to the SQL ORDER BY clause (which normally orders by post date in descending order with a limit of 1 post). Filter function arguments: ORDER BY clause.

get_previous_post_where : in function `get_previous_post` (which finds the post before the currently-displayed post), applied to the SQL WHERE clause (which normally looks for the previous dated published post). Filter function arguments: WHERE clause, stay in same category (true/false), list of excluded categories.

gettext : applied to the translated text by the `translation()` function (which is called by functions like the `__()` and `_e()` internationalization functions). Filter function arguments: translated text, untranslated text and the text domain. Gets applied even if internationalization is not in effect or if the text domain has not been loaded.

override_load_textdomain

get_meta_sql : in function `WP_Meta_Query::get_sql` (which generates SQL clauses to be appended to a main query for advanced meta queries.), applied to the SQL JOIN and WHERE clause generated by the advanced meta query. Filter function arguments: array(compact('join', 'where'), $this->queries, $type, $primary_table, $primary_id_column, $context)

get_others_drafts : applied to the query that selects the other users' drafts for display in the admin menus.

get_users_drafts : applied to the query that selects the users' drafts for display in the admin menus.

locale : applied to the locale by the `get_locale` function.

query : applied to all queries (at least all queries run after plugins are loaded).

query_string : deprecated – use `query_vars` or `request` instead.

query_vars : applied to the list of public WordPress query variables before the SQL query is formed. Useful for removing extra permalink information the plugin has dealt with in some other manner.

request : like `query_vars`, but applied after “extra” and private query variables have been added.

excerpt_length : Defines the length of a single-post excerpt.

excerpt_more : Defines the more string at the end of the excerpt.

post_edit_form_tag : Allows you to append code to the form tag in the default post/page editor.

update_user_query : applied to the update query used to update user information, prior to running the query.

uploading_iframe_src (removed since WP 2.5): applied to the HTML src tag for the uploading iframe on the post and page editing screens.

xmlrpc_methods : applied to list of defined XMLRPC methods for the XMLRPC server.

wp_mail_from : applied before any mail is sent by the wp_mail function. Supplied value is the calculated from address which is wordpress at the current hostname (set by $_SERVER[‘SERVER_NAME’]). The filter should return an email address or name/email combo in the form “user@example.com” or “Name <user@example.com>” (without the quotes!).

wp_mail_from_name : applied before any mail is sent by the wp_mail function. The filter should return a name string to be used as the email from name.

update_(meta_type)_metadata : applied before a metadata gets updated. For example if a user metadata gets updated the hook would be ‘update_user_metadata’

Widgets

This section contains filters added by widgets present in WordPress core.

dynamic_sidebar_params : applied to the arguments passed to the widgets_init function in the WordPress widgets.

widget_archives_dropdown_args : applied to the arguments passed to the `wp_get_archives()` function in the WordPress Archives widget.

widget_categories_args : applied to the arguments passed to the `wp_list_categories()` function in the WordPress Categories widget.

widget_links_args : applied to the arguments passed to the `wp_list_bookmarks()` function in the WordPress Links widget.

widget_nav_menu_args : applied to the arguments passed to the `wp_nav_menu()` function in the WordPress Custom Menu widget.

widget_pages_args : applied to the arguments passed to the `wp_list_pages()` function in the WordPress Pages widget.

widget_tag_cloud_args : applied to the arguments passed to the `wp_tag_cloud()` function in the WordPress Pages widget.

widget_text : applied to the widget text of the WordPress Text widget. May also apply to some third party widgets as well.

widget_title : applied to the widget title of any user editable WordPress Widget. May also apply to some third party widgets as well.

Admin Bar

This section contains filters added by the Admin Bar added in WordPress 3.1.0.

wp_admin_bar_class : allows changing the default 'WP_Admin_Bar' class in the `_wp_admin_bar_init()` function in .

Further Reading

- Hooks Database – documentation for all hooks in the official WordPress Code Reference.

□□□ #2
□ □□□ □□□ 2026-08-29 15:36:33 UTC
□ □□□ □□□ 2026-08-29 15:36:33 UTC