XThreads Documentation - Undocumented Features

About Undocumented Features

This page will describe a number of things not documented in the interface, due to lack of space or otherwise.

Thread Filtering

XThreads offers a `{$filters_set}` array variable on the forumdisplay page when thread filters are used. This section describes this array.

The general format to use in templates is something like this: `{$filters_set['key']['var']}`. *key* refers to the filter; for example, if you've enabled filtering on a thread field named "myfield", the corresponding *key* is *myfield*. As for *var*, the following are available:

- **value**: value of selected filter, for example, if the user is filtering "myfield" by "label_a", sticking `{$filters_set['myfield']['value']}` in the appropriate template will display `label_a`. Note, if an array of filters have been used, this will contain all values selected, separated by commas.
- **urlarg**: URL parameter with selected filter, for example, `filtertf_myfield=label_a`
- **urlarga**: same as above, but with a preceeding ampersand, eg `&amp;filtertf_myfield=label_a`
- **urlargq**: same as above, but with a preceeding question mark, eg `?filtertf_myfield=label_a`
- **forminput**: like *urlarg*, but as a hidden form parameter, eg `<input type="hidden" name="filtertf_myfield" value="label_a" />`
- **selected**: an array, containing the selected value as a key and ` selected="selected"` as the value (that's a preceeding space before 'selected' there). Example usage - the appropriate option in the listbox will be selected depending on what filter is being used:
  ```
  <select name="filtertf_myfield">
  <option value="label_a"{$filters_set['myfield']['selected']['label_a']}>label_a</option>
  <option value="label_b"{$filters_set['myfield']['selected']['label_b']}>label_b</option>
  ...
  </select>
  ```
- **checked**: same as above, but sets value to ` checked="checked"`
- **active**: same as above, but sets value to `filtertf_active` (may be useful for applying a CSS class)
- **nullselected**: if filter is NOT set, will set to ` selected="selected"`, otherwise will be blank
- **nullchecked**: if filter is NOT set, will set to ` checked="checked"`, otherwise will be blank
- **nullactive**: if filter is NOT set, will set to `filtertf_active`, otherwise will be blank
- **hiddencss**: puts hidden CSS if filter is set, is blank otherwise; in other words, if this filter is being used at all, this will be set to `display: none;`
- **visiblecss**: opposite of the above; set to `display: none;` if filter is NOT set, or blank otherwise

There are also some special values for *key*, these are:

- **__xt_uid**, **__xt_lastposteruid**, **__xt_prefix** and **__xt_icon**: for filtering by uid, lastposteruid, prefix and icon, respectively
- **__all**: aggregation of all thread field filters, and uid/lastposteruid/prefix/icon filters (does not include inline search); this doesn't support *selected*, *checked* and *active* as *var*
- **__search**: for inline forum searching, if enabled

Display Format Fields

The *Display Format*, *Display Item Format*, *Blank Replacement Value*, *Unviewable Value* and replacement values for *Formatting Map List* have fairly complex display formatting abilities - this section will try to explain how it works.

Special Tokens

- `{VALUE}` - represents the value; not available for *Blank Replacement Value* as the value is blank here
- `{RAWVALUE}` - as above, but without the sanitisation step (step 5 below) - use with caution! Only available in *Display*

Item Format and *Display Format* if multiple values are not allowed.
- `{VALUE$1}`, `{VALUE$2}`, `{VALUE$3}` etc - represents captured value components if a capturing *Text Mask Filter* is defined
- `{RAWVALUE$1}`, `{RAWVALUE$2}`, `{RAWVALUE$3}` etc - as above, but values not gone through the sanitiser
- Variables - similar to how they work in MyBB templates, eg `{$mybb->user['username']}`; note, these must be surrounded by curly braces as in the preceeding example. Variables are always in the global scope.
- Conditionals - see the following section

Conditionals

XThreads supports conditionals in many fields, which are in the same syntax as the Template Conditionals plugin. There are a few differences/notes with the plugin however:

- `<template ...>` calls are not allowed
- `{VALUE}` and similar tokens above are treated as quoted string values, eg `<?='This is '.{VALUE}?>` is valid syntax
- All variables are auto-global'd - no need to do this explicitly. Note that, because of this, it won't work correctly with some PHP syntax, such as `$$variable`
- The two functions added in v1.4 of *Template Conditionals* are named `xthreads_phptpl_eval_expr` and `xthreads_phptpl_eval_text`

PHP support is also available, but only if the PHP in Templates plugin is installed and activated (and/or appropriate setting set in *cache/xthreads.php*); accepts PHP within `<?php ... ?>` tags, similar to how the plugin works.

Parse Process

1. If the value is blank, the *Blank Replacement Value* will be displayed
2. If the user is in a group which cannot view the value, the process will continue, however *Unviewable Value* will be used in place of *Display Format*
3. At this point, file thread fields will go off and do its own thing; other thread fields continue
4. If the value matches one defined in the *Formatting Value List*, the defined replacement will be used here and the following step will be skipped
5. The value will be sanitised according to the *Display Parsing* option chosen, at this point (eg, MyCode parsed)
6. If a capturing *Text Mask Filter* is used, components are captured and sent through the sanitiser (as above)
7. If multiple values are supported, the above goes into *Display Item Format*, and the above two steps are repeated for each value. Once done, all these values are aggregated together, separated by the *Multiple Value Delimiter*
8. Finally, the value goes into the *Display Format*, for display

Additional Custom Thread Field Variables

In addition to the `{$GLOBALS['threadfields']['`*`key`*`']}` variable (which displays the value of the thread field), there are other values which can be displayed via the additional variable, `{$GLOBALS['threadfields_x']['`*`key`*`']['`*`valuename`*`']}` variable, where *valuename* can be one of the following:

- **title** - displays the title given to the thread field
- **desc** - displays the description given to the thread field
- **num_values** - displays the number of values stored in this thread - will always be `1` for non-multivalued fields, or `0` if no value entered
- **num_values_friendly** - formatted version of the above
- **value** - reference back to `$threadfields['`*`key`*`']`, unless the field is a multivalued field, in which case, this is an array of each individual value

Additional Template Variables

XThreads adds some variables which can be used in templates for convenience.

- Thread/forum URL variables through `{$forumurl}`, `{$forumurl_q}`, `{$threadurl}` and `{$threadurl_q}` (the `_q` variants append a `?` or `&amp;` depending on the context of the URL). Example: `<a href="{$forumurl?}filtertf_myfield=something">filter by something</a>`

- Thread start date is evaluated in a variable through `{$thread['threaddate']}` and `{$thread['threadtime']}`; available in forumdisplay/showthread/search

Other

- Admins can add XThreads attachments by placing the file in the *uploads/xthreads_ul/admindrop* folder (and not a subfolder of that), and specify the filename (through URL box) using *file://* prefix. For example, if an admin tells the system to retrieve the URL *file://bigfile.iso*, XThreads will **move** the file in *uploads/xthreads_ul/admindrop/bigfile.iso* to where xtattachments are stored.
- You can force xthreads_attach.php to download a file (as opposed to viewing the file in the browser) by appending *?download=1* (or *&download=1* if you're using query strings) to the URL
- A number of "internal variables" can be modified by editing the *cache/xthreads.php* file. XThreads needs to be installed for this file to exist. Note that this file is auto-generated and may be overwritten during an upgrade, however values will be retained.
- Template modifications performed by XThreads can be redone (ie, after you install a new theme) by deactivating and reactivating the plugin (without un/installing)
- XThreads can sort any non-textarea custom thread field in forumdisplay. This reuses MyBB's sorting system - just that the URL parameter is a little different.
  To sort by a thread field, prefix its key with `tf_`, for example, *forumdisplay.php?fid=2&sortby=tf_myfield* will sort the threads by *myfield*. File input fields are a bit different, they need to be prefixed with `tfa_component_` where *component* is either `filename`, `filesize`, `uploadtime`, `updatetime` or `downloads`.
  Although XThreads adds this ability transparently (and to the list of default sort bys in the AdminCP), it is up to you to add appropriate options to the forumdisplay sorter listbox, if you want to.
  As this uses MyBB's system, template variables will be the same, eg `{$sortsel['tf_myfield']}` will be `selected="selected"` if sorting by *myfield*.
  Note that this is not indexed, like MyBB's sorting options aren't.
- XThreads can filter forumdisplay by thread starter, last poster, thread icon and thread prefix (MyBB 1.6 only); to do this, supply *filterxt_field=value* to the URL, where *field* is either *uid*, *lastposteruid*, *icon*, *prefix*. For example, *forumdisplay.php?fid=2&filterxt_uid=1*
- In a thread field's *Display Format Fields* (includes *Display Format*, *Display Item Format*, *Blank Replacement Value*, *Unviewable Value* and replacement values for *Formatting Map List*), references to other thread fields are possible, only if the *Display Order* is set lower than the thread field making the reference (for those with the same *Display Order*, whether or not it'll work will depend on your luck). For example, if we have two thread fields, the first having a *Display Order* of 1 and the second having a *Display Format* of 2, then this second field can refer to the first field via `{$GLOBALS['threadfields']['key']}` in any of its *Display Format Fields* (where *key* is the *Key* of the first thread field, which also isn't a File input field).

XThreads Additions

Just a list of the stuff XThreads adds:

AdminCP UI Additions

- *Custom Thread Fields* section under the Configuration tab - this is where you can add custom thread fields
- *XThreads Options* section under forum add/edit - this allows you to customise various XThreads options on a per forum basis; XThreads also adds to the list of sorting options, if you have sortable thread fields assigned to the forum
- *Modify Custom Thread Fields* textbox under add/edit Moderator Tools - this allows moderator tools to modify custom thread fields; note that values here are NOT validated and permissions are NOT checked!
- *Rebuild `threadfields` cache* under *Cache Manager* for rebuilding XThreads caches
- *Rebuild XThreads Attachment Thumbnails* under *Recount & Rebuild* for rebuilding thumbnails associated with uploaded XThreads images (also required if you change thumbnail sizes)
- *Prune XThreads Orphaned Attachments* task, which periodically cleans up orphaned XThreads attachments (as well as performs deferred MD5 hashing), similar to MyBB's *Find Orphaned Attachments*
- Folder chmod check under *Tools & Maintenance*, *System Health*

Modified Templates

- **editpost** and **newthread** - adds `{$extra_threadfields}` to the template, which just displays any added thread fields

- **showthread** - inserts `{$first_post}` before `{$posts}`; in XThreads, it's possible to customise the first post differently from other posts
- **forumdisplay_threadlist**
  - insert `{$nullthreads}` after `{$threads}`; this is just for display "null threads" if thread grouping is used
  - insert `{$sort_by_prefix}` before the first option in the sorter listbox - this is to give the option to sort threads by thread prefix
  - from the above, some other sorting options are added below the Sort By: Views option in the listbox
- **forumdisplay_threadlist_sortrating** - similar to the above, inserts a sort by number of ratings option to the sorter listbox

Added Templates

- **editpost_first** - allows you to have a different *editpost* template for editing the first post of a thread; defaults to `{$editpost}` which uses the default *editpost* template
- **forumdisplay_group_sep** and **forumdisplay_thread_null** - for use with thread grouping (is explained if you enable thread grouping); defaults to nothing
- **showthread_noreplies** - displayed in the `{$posts}` variable for *showthread* only if the thread has no replies; see the comment in the template for more information
- **forumdisplay_searchforum_inline** - inline forum searching - you don't need to bother with this unless you want to...
- **threadfields_inputrow** - how extra fields are displayed in newthread/editpost (of first post) - don't need to worry about this one either

Although not an added template, you can create *postbit_first** templates to override the various *postbit** templates for the first post of a thread.