Taglist manual

```
*taglist.txt*   Plugin for browsing source code

Author: Yegappan Lakshmanan  (yegappan AT yahoo DOT com)
For Vim version 6.0 and above
Last change: 2013 Feburary 26
```

```
                                                    *taglist-intro*
1. Overview

The "Tag List" plugin is a source code browser plugin for Vim. This plugin
allows you to efficiently browse through source code files for different
programming languages. The "Tag List" plugin provides the following features:

    * Displays the tags (functions, classes, structures, variables, etc.)
      defined in a file in a vertically or horizontally split Vim window.
    * In GUI Vim, optionally displays the tags in the Tags drop-down menu and
      in the popup menu.
    * Automatically updates the taglist window as you switch between
      files/buffers. As you open new files, the tags defined in the new files
      are added to the existing file list and the tags defined in all the
      files are displayed grouped by the filename.
    * When a tag name is selected from the taglist window, positions the
      cursor at the definition of the tag in the source file.
    * Automatically highlights the current tag name.
    * Groups the tags by their type and displays them in a foldable tree.
    * Can display the prototype and scope of a tag.
    * Can optionally display the tag prototype instead of the tag name in the
      taglist window.
    * The tag list can be sorted either by name or by chronological order.
    * Supports the following language files: Assembly, ASP, Awk, Beta, C,
      C++, C#, Cobol, Eiffel, Erlang, Fortran, HTML, Java, Javascript, Lisp,
      Lua, Make, Pascal, Perl, PHP, Python, Rexx, Ruby, Scheme, Shell, Slang,
      SML, Sql, TCL, Verilog, Vim and Yacc.
    * Can be easily extended to support new languages. Support for
      existing languages can be modified easily.
    * Provides functions to display the current tag name in the Vim status
      line or the window title bar.
    * The list of tags and files in the taglist can be saved and
      restored across Vim sessions.
    * Provides commands to get the name and prototype of the current tag.
    * Runs in both console/terminal and GUI versions of Vim.
    * Works with the winmanager plugin. Using the winmanager plugin, you
      can use Vim plugins like the file explorer, buffer explorer and the
      taglist plugin at the same time like an IDE.
```

    * Can be used in both Unix and MS-Windows systems.

---

                                                taglist-internet
2. Taglist on the internet

The home page of the taglist plugin is at:

        http://vim-taglist.sourceforge.net/

You can subscribe to the taglist mailing list to post your questions or suggestions for improvement or to send bug reports. Visit the following page for subscribing to the mailing list:

        http://groups.yahoo.com/group/taglist

---

                                                taglist-requirements
3. Requirements

The taglist plugin requires the following:

    * Vim version 6.0 and above
    * Exuberant ctags 5.0 and above

The taglist plugin will work on all the platforms where the exuberant ctags utility and Vim are supported (this includes MS-Windows and Unix based systems).

The taglist plugin relies on the exuberant ctags utility to dynamically generate the tag listing. The exuberant ctags utility must be installed in your system to use this plugin. The exuberant ctags utility is shipped with most of the Linux distributions. You can download the exuberant ctags utility from

        http://ctags.sourceforge.net

The taglist plugin doesn't use or create a tags file and there is no need to create a tags file to use this plugin. The taglist plugin will not work with the GNU ctags or the Unix ctags utility.

This plugin relies on the Vim "filetype" detection mechanism to determine the type of the current file. You have to turn on the Vim filetype detection by adding the following line to your .vimrc file:

```
        filetype on
```

The taglist plugin will not work if you run Vim in the restricted mode (using the -Z command-line argument).

The taglist plugin uses the Vim system() function to invoke the exuberant ctags utility. If Vim is compiled without the system() function then you cannot use the taglist plugin. Some of the Linux distributions (Suse) compile Vim without the system() function for security reasons.

---

                                                taglist-install
4. Installation

1. Download the taglist.zip file and unzip the files to the $HOME/.vim or the $HOME/vimfiles or the $VIM/vimfiles directory. After this step, you should have the following two files (the directory structure should be preserved):

```
        plugin/taglist.vim - main taglist plugin file
        doc/taglist.txt    - documentation (help) file
```

   Refer to the |add-plugin|and |'runtimepath'| Vim help pages for more
   details about installing Vim plugins.
2. Change to the $HOME/.vim/doc or $HOME/vimfiles/doc or $VIM/vimfiles/doc
   directory, start Vim and run the ":helptags ." command to process the
   taglist help file. Without this step, you cannot jump to the taglist help
   topics.
3. If the exuberant ctags utility is not present in one of the directories in
   the PATH environment variable, then set the 'Tlist_Ctags_Cmd' variable to
   point to the location of the exuberant ctags utility (not to the directory)
   in the .vimrc file.
4. If you are running a terminal/console version of Vim and the terminal
   doesn't support changing the window width then set the
   'Tlist_Inc_Winwidth' variable to 0 in the .vimrc file.
5. Restart Vim.
6. You can now use the ":TlistToggle" command to open/close the taglist
   window. You can use the ":help taglist" command to get more information
   about using the taglist plugin.

To uninstall the taglist plugin, remove the plugin/taglist.vim and
doc/taglist.txt files from the $HOME/.vim or $HOME/vimfiles directory.

---

                                                taglist-using

5. Usage

The taglist plugin can be used in several different ways.

1. You can keep the taglist window open during the entire editing session. On
   opening the taglist window, the tags defined in all the files in the Vim
   buffer list will be displayed in the taglist window. As you edit files, the
   tags defined in them will be added to the taglist window. You can select a
   tag from the taglist window and jump to it. The current tag will be
   highlighted in the taglist window. You can close the taglist window when
   you no longer need the window.
2. You can configure the taglist plugin to process the tags defined in all the
   edited files always. In this configuration, even if the taglist window is
   closed and the taglist menu is not displayed, the taglist plugin will
   processes the tags defined in newly edited files. You can then open the
   taglist window only when you need to select a tag and then automatically
   close the taglist window after selecting the tag.
3. You can configure the taglist plugin to display only the tags defined in
   the current file in the taglist window. By default, the taglist plugin
   displays the tags defined in all the files in the Vim buffer list. As you
   switch between files, the taglist window will be refreshed to display only
   the tags defined in the current file.
4. In GUI Vim, you can use the Tags pull-down and popup menu created by the
   taglist plugin to display the tags defined in the current file and select a
   tag to jump to it. You can use the menu without opening the taglist window.
   By default, the Tags menu is disabled.
5. You can configure the taglist plugin to display the name of the current tag
   in the Vim window status line or in the Vim window title bar. For this to
   work without the taglist window or menu, you need to configure the taglist
   plugin to process the tags defined in a file always.
6. You can save the tags defined in multiple files to a taglist session file
   and load it when needed. You can also configure the taglist plugin to not
   update the taglist window when editing new files. You can then manually add
   files to the taglist window.

Opening the taglist window
You can open the taglist window using the ":TlistOpen" or the ":TlistToggle"

commands. The ":TlistOpen" command opens the taglist window and jumps to it. The ":TlistToggle" command opens or closes (toggle) the taglist window and the cursor remains in the current window. If the 'Tlist_GainFocus_On_ToggleOpen' variable is set to 1, then the ":TlistToggle" command opens the taglist window and moves the cursor to the taglist window.

You can map a key to invoke these commands. For example, the following command creates a normal mode mapping for the <F8> key to toggle the taglist window.

```
        nnoremap <silent> <F8> :TlistToggle<CR>
```

Add the above mapping to your ~/.vimrc or $HOME/_vimrc file.

To automatically open the taglist window on Vim startup, set the 'Tlist_Auto_Open' variable to 1.

You can also open the taglist window on startup using the following command line:

```
        $ vim +TlistOpen
```

Closing the taglist window
You can close the taglist window from the taglist window by pressing 'q' or using the Vim ":q" command. You can also use any of the Vim window commands to close the taglist window. Invoking the ":TlistToggle" command when the taglist window is opened, closes the taglist window. You can also use the ":TlistClose" command to close the taglist window.

To automatically close the taglist window when a tag or file is selected, you can set the 'Tlist_Close_On_Select' variable to 1. To exit Vim when only the taglist window is present, set the 'Tlist_Exit_OnlyWindow' variable to 1.

Jumping to a tag or a file
You can select a tag in the taglist window either by pressing the <Enter> key or by double clicking the tag name using the mouse. To jump to a tag on a single mouse click set the 'Tlist_Use_SingleClick' variable to 1.

If the selected file is already opened in a window, then the cursor is moved to that window. If the file is not currently opened in a window then the file is opened in the window used by the taglist plugin to show the previously selected file. If there are no usable windows, then the file is opened in a new window. The file is not opened in special windows like the quickfix window, preview window and windows containing buffer with the 'buftype' option set.

To jump to the tag in a new window, press the 'o' key. To open the file in the previous window (Ctrl-W_p) use the 'P' key. You can press the 'p' key to jump to the tag but still keep the cursor in the taglist window (preview).

To open the selected file in a tab, use the 't' key. If the file is already present in a tab then the cursor is moved to that tab otherwise the file is opened in a new tab. To jump to a tag in a new tab press Ctrl-t. The taglist window is automatically opened in the newly created tab.

Instead of jumping to a tag, you can open a file by pressing the <Enter> key or by double clicking the file name using the mouse.

In the taglist window, you can use the [[or <Backspace> key to jump to the beginning of the previous file. You can use the]] or <Tab> key to jump to the beginning of the next file. When you reach the first or last file, the search wraps around and the jumps to the next/previous file.

Highlighting the current tag
The taglist plugin automatically highlights the name of the current tag in the taglist window. The Vim |CursorHold| autocmd event is used for this. If the

current tag name is not visible in the taglist window, then the taglist window contents are scrolled to make that tag name visible. You can also use the ":TlistHighlightTag" command to force the highlighting of the current tag.

The tag name is highlighted if no activity is performed for |'updatetime'| milliseconds. The default value for this Vim option is 4 seconds. To avoid unexpected problems, you should not set the |'updatetime'| option to a very low value.

To disable the automatic highlighting of the current tag name in the taglist window, set the 'Tlist_Auto_Highlight_Tag' variable to zero.

When entering a Vim buffer/window, the taglist plugin automatically highlights the current tag in that buffer/window. If you like to disable the automatic highlighting of the current tag when entering a buffer, set the 'Tlist_Highlight_Tag_On_BufEnter' variable to zero.

Adding files to the taglist
When the taglist window is opened, all the files in the Vim buffer list are processed and the supported files are added to the taglist. When you edit a file in Vim, the taglist plugin automatically processes this file and adds it to the taglist. If you close the taglist window, the tag information in the taglist is retained.

To process files even when the taglist window is not open, set the 'Tlist_Process_File_Always' variable to 1.

You can manually add multiple files to the taglist without opening them using the ":TlistAddFiles" and the ":TlistAddFilesRecursive" commands.

For example, to add all the C files in the /my/project/dir directory to the taglist, you can use the following command:

```
        :TlistAddFiles /my/project/dir/*.c
```

Note that when adding several files with a large number of tags or a large number of files, it will take several seconds to several minutes for the taglist plugin to process all the files. You should not interrupt the taglist plugin by pressing <CTRL-C>.

You can recursively add multiple files from a directory tree using the ":TlistAddFilesRecursive" command:

```
        :TlistAddFilesRecursive /my/project/dir *.c
```

This command takes two arguments. The first argument specifies the directory from which to recursively add the files. The second optional argument specifies the wildcard matching pattern for selecting the files to add. The default pattern is * and all the files are added.

Displaying tags for only one file
The taglist window displays the tags for all the files in the Vim buffer list and all the manually added files. To display the tags for only the current active buffer, set the 'Tlist_Show_One_File' variable to 1.

Removing files from the taglist
You can remove a file from the taglist window, by pressing the 'd' key when the cursor is on one of the tags listed for the file in the taglist window. The removed file will no longer be displayed in the taglist window in the current Vim session. To again display the tags for the file, open the file in a Vim window and then use the ":TlistUpdate" command or use ":TlistAddFiles" command to add the file to the taglist.

When a buffer is removed from the Vim buffer list using the ":bdelete" or the ":bwipeout" command, the taglist is updated to remove the stored information for this buffer.

Updating the tags displayed for a file
The taglist plugin keeps track of the modification time of a file. When the modification time changes (the file is modified), the taglist plugin automatically updates the tags listed for that file. The modification time of a file is checked when you enter a window containing that file or when you load that file.

You can also update or refresh the tags displayed for a file by pressing the "u" key in the taglist window. If an existing file is modified, after the file is saved, the taglist plugin automatically updates the tags displayed for the file.

You can also use the ":TlistUpdate" command to update the tags for the current buffer after you made some changes to it. You should save the modified buffer before you update the taglist window. Otherwise the listed tags will not include the new tags created in the buffer.

If you have deleted the tags displayed for a file in the taglist window using the 'd' key, you can again display the tags for that file using the ":TlistUpdate" command.

Controlling the taglist updates
To disable the automatic processing of new files or modified files, you can set the 'Tlist_Auto_Update' variable to zero. When this variable is set to zero, the taglist is updated only when you use the ":TlistUpdate" command or the ":TlistAddFiles" or the ":TlistAddFilesRecursive" commands. You can use this option to control which files are added to the taglist.

You can use the ":TlistLock" command to lock the taglist contents. After this command is executed, new files are not automatically added to the taglist. When the taglist is locked, you can use the ":TlistUpdate" command to add the current file or the ":TlistAddFiles" or ":TlistAddFilesRecursive" commands to add new files to the taglist. To unlock the taglist, use the ":TlistUnlock" command.

Displaying the tag prototype
To display the prototype of the tag under the cursor in the taglist window, press the space bar. If you place the cursor on a tag name in the taglist window, then the tag prototype is displayed at the Vim status line after |'updatetime'| milliseconds. The default value for the |'updatetime'| Vim option is 4 seconds.

You can get the name and prototype of a tag without opening the taglist window and the taglist menu using the ":TlistShowTag" and the ":TlistShowPrototype" commands. These commands will work only if the current file is already present in the taglist. To use these commands without opening the taglist window, set the 'Tlist_Process_File_Always' variable to 1.

You can use the ":TlistShowTag" command to display the name of the tag at or before the specified line number in the specified file. If the file name and line number are not supplied, then this command will display the name of the current tag. For example,

```
        :TlistShowTag
        :TlistShowTag myfile.java 100
```

You can use the ":TlistShowPrototype" command to display the prototype of the tag at or before the specified line number in the specified file. If the file name and the line number are not supplied, then this command will display the prototype of the current tag. For example,

```
        :TlistShowPrototype
        :TlistShowPrototype myfile.c 50
```

In the taglist window, when the mouse is moved over a tag name, the tag prototype is displayed in a balloon. This works only in GUI versions where balloon evaluation is supported.

Taglist window contents
The taglist window contains the tags defined in various files in the taglist grouped by the filename and by the tag type (variable, function, class, etc.). For tags with scope information (like class members, structures inside structures, etc.), the scope information is displayed in square brackets "[]" after the tag name.

The contents of the taglist buffer/window are managed by the taglist plugin. The |'filetype'| for the taglist buffer is set to 'taglist'. The Vim |'modifiable'| option is turned off for the taglist buffer. You should not manually edit the taglist buffer, by setting the |'modifiable'| flag. If you manually edit the taglist buffer contents, then the taglist plugin will be out of sync with the taglist buffer contents and the plugin will no longer work correctly. To redisplay the taglist buffer contents again, close the taglist window and reopen it.

Opening and closing the tag and file tree
In the taglist window, the tag names are displayed as a foldable tree using the Vim folding support. You can collapse the tree using the '-' key or using the Vim |zc| fold command. You can open the tree using the '+' key or using the Vim |zo| fold command. You can open all the folds using the '*' key or using the Vim |zR| fold command. You can also use the mouse to open/close the folds. You can close all the folds using the '=' key. You should not manually create or delete the folds in the taglist window.

To automatically close the fold for the inactive files/buffers and open only the fold for the current buffer in the taglist window, set the 'Tlist_File_Fold_Auto_Close' variable to 1.

Sorting the tags for a file
The tags displayed in the taglist window can be sorted either by their name or by their chronological order. The default sorting method is by the order in which the tags appear in a file. You can change the default sort method by setting the 'Tlist_Sort_Type' variable to either "name" or "order". You can sort the tags by their name by pressing the "s" key in the taglist window. You can again sort the tags by their chronological order using the "s" key. Each file in the taglist window can be sorted using different order.

Zooming in and out of the taglist window
You can press the 'x' key in the taglist window to maximize the taglist window width/height. The window will be maximized to the maximum possible width/height without closing the other existing windows. You can again press 'x' to restore the taglist window to the default width/height.

                                                taglist-session
Taglist Session
A taglist session refers to the group of files and their tags stored in the taglist in a Vim session.

You can save and restore a taglist session (and all the displayed tags) using the ":TlistSessionSave" and ":TlistSessionLoad" commands.

To save the information about the tags and files in the taglist to a file, use the ":TlistSessionSave" command and specify the filename:

```
        :TlistSessionSave <file name>
```

To load a saved taglist session, use the ":TlistSessionLoad" command:

```
        :TlistSessionLoad <file name>
```

When you load a taglist session file, the tags stored in the file will be added to the tags already stored in the taglist.

The taglist session feature can be used to save the tags for large files or a group of frequently used files (like a project). By using the taglist session file, you can minimize the amount to time it takes to load/refresh the taglist for multiple files.

You can create more than one taglist session file for multiple groups of files.

Displaying the tag name in the Vim status line or the window title bar
You can use the Tlist_Get_Tagname_By_Line() function provided by the taglist plugin to display the current tag name in the Vim status line or the window title bar. Similarly, you can use the Tlist_Get_Tag_Prototype_By_Line() function to display the current tag prototype in the Vim status line or the window title bar.

For example, the following command can be used to display the current tag name in the status line:

```
        :set statusline=%<%f%=%([%{Tlist_Get_Tagname_By_Line()}]%)
```

The following command can be used to display the current tag name in the window title bar:

```
        :set title titlestring=%<%f\ %([%{Tlist_Get_Tagname_By_Line()}]%)
```

Note that the current tag name can be displayed only after the file is processed by the taglist plugin. For this, you have to either set the 'Tlist_Process_File_Always' variable to 1 or open the taglist window or use the taglist menu. For more information about configuring the Vim status line, refer to the documentation for the Vim |'statusline'| option.

Changing the taglist window highlighting
The following Vim highlight groups are defined and used to highlight the various entities in the taglist window:

```
    TagListTagName  - Used for tag names
    TagListTagScope - Used for tag scope
    TagListTitle    - Used for tag titles
    TagListComment  - Used for comments
    TagListFileName - Used for filenames
```

By default, these highlight groups are linked to the standard Vim highlight groups. If you want to change the colors used for these highlight groups, prefix the highlight group name with 'My' and define it in your .vimrc or .gvimrc file: MyTagListTagName, MyTagListTagScope, MyTagListTitle, MyTagListComment and MyTagListFileName. For example, to change the colors used for tag names, you can use the following command:

```
    :highlight MyTagListTagName guifg=blue ctermfg=blue
```

Controlling the taglist window
To use a horizontally split taglist window, instead of a vertically split window, set the 'Tlist_Use_Horiz_Window' variable to 1.

To use a vertically split taglist window on the rightmost side of the Vim window, set the 'Tlist_Use_Right_Window' variable to 1.

You can specify the width of the vertically split taglist window, by setting the 'Tlist_WinWidth' variable. You can specify the height of the horizontally split taglist window, by setting the 'Tlist_WinHeight' variable.

When opening a vertically split taglist window, the Vim window width is increased to accommodate the new taglist window. When the taglist window is closed, the Vim window is reduced. To disable this, set the 'Tlist_Inc_Winwidth' variable to zero.

To reduce the number of empty lines in the taglist window, set the 'Tlist_Compact_Format' variable to 1.

To not display the Vim fold column in the taglist window, set the 'Tlist_Enable_Fold_Column' variable to zero.

To display the tag prototypes instead of the tag names in the taglist window, set the 'Tlist_Display_Prototype' variable to 1.

To not display the scope of the tags next to the tag names, set the 'Tlist_Display_Tag_Scope' variable to zero.

                                                taglist-keys
Taglist window key list
The following table lists the description of the keys that can be used in the taglist window.

Key	Description
<CR>	Jump to the location where the tag under cursor is defined.
o	Jump to the location where the tag under cursor is defined in a new window.
P	Jump to the tag in the previous (Ctrl-W_p) window.
p	Display the tag definition in the file window and keep the cursor in the taglist window itself.
t	Jump to the tag in a new tab. If the file is already opened in a tab, move to that tab.
Ctrl-t	Jump to the tag in a new tab.
<Space>	Display the prototype of the tag under the cursor. For file names, display the full path to the file, file type and the number of tags. For tag types, display the tag type and the number of tags.
u	Update the tags listed in the taglist window
s	Change the sort order of the tags (by name or by order)
d	Remove the tags for the file under the cursor
x	Zoom-in or Zoom-out the taglist window
+	Open a fold
-	Close a fold
*	Open all folds
=	Close all folds
[[	Jump to the beginning of the previous file
<Backspace>	Jump to the beginning of the previous file
]]	Jump to the beginning of the next file
<Tab>	Jump to the beginning of the next file
q	Close the taglist window
<F1>	Display help

The above keys will work in both the normal mode and the insert mode.

                                                taglist-menu
Taglist menu
When using GUI Vim, the taglist plugin can display the tags defined in the current file in the drop-down menu and the popup menu. By default, this feature is turned off. To turn on this feature, set the 'Tlist_Show_Menu' variable to 1.

You can jump to a tag by selecting the tag name from the menu. You can use the taglist menu independent of the taglist window i.e. you don't need to open the taglist window to get the taglist menu.

When you switch between files/buffers, the taglist menu is automatically
updated to display the tags defined in the current file/buffer.

The tags are grouped by their type (variables, functions, classes, methods,
etc.) and displayed as a separate sub-menu for each type. If all the tags
defined in a file are of the same type (e.g. functions), then the sub-menu is
not used.

If the number of items in a tag type submenu exceeds the value specified by
the 'Tlist_Max_Submenu_Items' variable, then the submenu will be split into
multiple submenus. The default setting for 'Tlist_Max_Submenu_Items' is 25.
The first and last tag names in the submenu are used to form the submenu name.
The menu items are prefixed by alpha-numeric characters for easy selection by
keyboard.

If the popup menu support is enabled (the |'mousemodel'| option contains
"popup"), then the tags menu is added to the popup menu. You can access
the popup menu by right clicking on the GUI window.

You can regenerate the tags menu by selecting the 'Tags->Refresh menu' entry.
You can sort the tags listed in the menu either by name or by order by
selecting the 'Tags->Sort menu by->Name/Order' menu entry.

You can tear-off the Tags menu and keep it on the side of the Vim window
for quickly locating the tags.

Using the taglist plugin with the winmanager plugin
You can use the taglist plugin with the winmanager plugin. This will allow you
to use the file explorer, buffer explorer and the taglist plugin at the same
time in different windows. To use the taglist plugin with the winmanager
plugin, set 'TagList' in the 'winManagerWindowLayout' variable. For example,
to use the file explorer plugin and the taglist plugin at the same time, use
the following setting:

```
        let winManagerWindowLayout = 'FileExplorer|TagList'
```

Getting help
If you have installed the taglist help file (this file), then you can use the
Vim ":help taglist-<keyword>" command to get help on the various taglist
topics.

You can press the <F1> key in the taglist window to display the help
information about using the taglist window. If you again press the <F1> key,
the help information is removed from the taglist window.

                                                taglist-debug
Debugging the taglist plugin
You can use the ":TlistDebug" command to enable logging of the debug messages
from the taglist plugin. To display the logged debug messages, you can use the
":TlistMessages" command. To disable the logging of the debug messages, use
the ":TlistUndebug" command.

You can specify a file name to the ":TlistDebug" command to log the debug
messages to a file. Otherwise, the debug messages are stored in a script-local
variable. In the later case, to minimize memory usage, only the last 3000
characters from the debug messages are stored.

---

                                                taglist-options
6. Options

A number of Vim variables control the behavior of the taglist plugin. These
variables are initialized to a default value. By changing these variables you

can change the behavior of the taglist plugin. You need to change these settings only if you want to change the behavior of the taglist plugin. You should use the |:let| command in your .vimrc file to change the setting of any of these variables.

The configurable taglist variables are listed below. For a detailed description of these variables refer to the text below this table.

|'Tlist_Auto_Highlight_Tag'|	Automatically highlight the current tag in the taglist.
|'Tlist_Auto_Open'|	Open the taglist window when Vim starts.
|'Tlist_Auto_Update'|	Automatically update the taglist to include newly edited files.
|'Tlist_Close_On_Select'|	Close the taglist window when a file or tag is selected.
|'Tlist_Compact_Format'|	Remove extra information and blank lines from the taglist window.
|'Tlist_Ctags_Cmd'|	Specifies the path to the ctags utility.
|'Tlist_Display_Prototype'|	Show prototypes and not tags in the taglist window.
|'Tlist_Display_Tag_Scope'|	Show tag scope next to the tag name.
|'Tlist_Enable_Fold_Column'|	Show the fold indicator column in the taglist window.
|'Tlist_Exit_OnlyWindow'|	Close Vim if the taglist is the only window.
|'Tlist_File_Fold_Auto_Close'|	Close tag folds for inactive buffers.
|'Tlist_GainFocus_On_ToggleOpen'|	Jump to taglist window on open.
|'Tlist_Highlight_Tag_On_BufEnter'|	On entering a buffer, automatically highlight the current tag.
|'Tlist_Inc_Winwidth'|	Increase the Vim window width to accommodate the taglist window.
|'Tlist_Max_Submenu_Items'|	Maximum number of items in a tags sub-menu.
|'Tlist_Max_Tag_Length'|	Maximum tag length used in a tag menu entry.
|'Tlist_Process_File_Always'|	Process files even when the taglist window is closed.
|'Tlist_Show_Menu'|	Display the tags menu.
|'Tlist_Show_One_File'|	Show tags for the current buffer only.
|'Tlist_Sort_Type'|	Sort method used for arranging the tags.
|'Tlist_Use_Horiz_Window'|	Use a horizontally split window for the taglist window.
|'Tlist_Use_Right_Window'|	Place the taglist window on the right side.
|'Tlist_Use_SingleClick'|	Single click on a tag jumps to it.
|'Tlist_WinHeight'|	Horizontally split taglist window height.
|'Tlist_WinWidth'|	Vertically split taglist window width.

'Tlist_Auto_Highlight_Tag'

Tlist_Auto_Highlight_Tag
The taglist plugin will automatically highlight the current tag in the taglist window. If you want to disable this, then you can set the 'Tlist_Auto_Highlight_Tag' variable to zero. Note that even though the current tag highlighting is disabled, the tags for a new file will still be added to the taglist window.

```
        let Tlist_Auto_Highlight_Tag = 0
```

With the above variable set to 1, you can use the ":TlistHighlightTag" command to highlight the current tag.

'Tlist_Auto_Open'

Tlist_Auto_Open
To automatically open the taglist window, when you start Vim, you can set the 'Tlist_Auto_Open' variable to 1. By default, this variable is set to zero and the taglist window will not be opened automatically on Vim startup.

```
        let Tlist_Auto_Open = 1

The taglist window is opened only when a supported type of file is opened on
Vim startup. For example, if you open text files, then the taglist window will
not be opened.

                                                *'Tlist_Auto_Update'*
Tlist_Auto_Update
When a new file is edited, the tags defined in the file are automatically
processed and added to the taglist. To stop adding new files to the taglist,
set the 'Tlist_Auto_Update' variable to zero. By default, this variable is set
to 1.

        let Tlist_Auto_Update = 0

With the above variable set to 1, you can use the ":TlistUpdate" command to
add the tags defined in the current file to the taglist.

                                                *'Tlist_Close_On_Select'*
Tlist_Close_On_Select
If you want to close the taglist window when a file or tag is selected, then
set the 'Tlist_Close_On_Select' variable to 1. By default, this variable is
set zero and when you select a tag or file from the taglist window, the window
is not closed.

        let Tlist_Close_On_Select = 1

                                                *'Tlist_Compact_Format'*
Tlist_Compact_Format
By default, empty lines are used to separate different tag types displayed for
a file and the tags displayed for different files in the taglist window. If
you want to display as many tags as possible in the taglist window, you can
set the 'Tlist_Compact_Format' variable to 1 to get a compact display.

        let Tlist_Compact_Format = 1

                                                *'Tlist_Ctags_Cmd'*
Tlist_Ctags_Cmd
The 'Tlist_Ctags_Cmd' variable specifies the location (path) of the exuberant
ctags utility. If exuberant ctags is present in any one of the directories in
the PATH environment variable, then there is no need to set this variable.

The exuberant ctags tool can be installed under different names.  When the
taglist plugin starts up, if the 'Tlist_Ctags_Cmd' variable is not set, it
checks for the names exuberant-ctags, exctags, ctags, ctags.exe and tags in
the PATH environment variable.  If any one of the named executable is found,
then the Tlist_Ctags_Cmd variable is set to that name.

If exuberant ctags is not present in one of the directories specified in the
PATH environment variable, then set this variable to point to the location of
the ctags utility in your system. Note that this variable should point to the
fully qualified exuberant ctags location and NOT to the directory in which
exuberant ctags is installed. If the exuberant ctags tool is not found in
either PATH or in the specified location, then the taglist plugin will not be
loaded. Examples:

        let Tlist_Ctags_Cmd = 'd:\tools\ctags.exe'
        let Tlist_Ctags_Cmd = '/usr/local/bin/ctags'

On Microsoft Windows, if ctags.exe is installed in a directory with space
characters in the name (e.g. C:\Program Files\ctags\ctags.exe), then you need
to set the Tlist_Ctags_Cmd variable like this:

        let Tlist_Ctags_Cmd = '"C:\Program Files\ctags\ctags.exe"'
```

```
                                                *'Tlist_Display_Prototype'*
Tlist_Display_Prototype
By default, only the tag name will be displayed in the taglist window. If you
like to see tag prototypes instead of names, set the 'Tlist_Display_Prototype'
variable to 1. By default, this variable is set to zero and only tag names
will be displayed.

        let Tlist_Display_Prototype = 1

                                                *'Tlist_Display_Tag_Scope'*
Tlist_Display_Tag_Scope
By default, the scope of a tag (like a C++ class) will be displayed in
square brackets next to the tag name. If you don't want the tag scopes
to be displayed, then set the 'Tlist_Display_Tag_Scope' to zero. By default,
this variable is set to 1 and the tag scopes will be displayed.

        let Tlist_Display_Tag_Scope = 0

                                                *'Tlist_Enable_Fold_Column'*
Tlist_Enable_Fold_Column
By default, the Vim fold column is enabled and displayed in the taglist
window. If you wish to disable this (for example, when you are working with a
narrow Vim window or terminal), you can set the 'Tlist_Enable_Fold_Column'
variable to zero.

        let Tlist_Enable_Fold_Column = 1

                                                *'Tlist_Exit_OnlyWindow'*
Tlist_Exit_OnlyWindow
If you want to exit Vim if only the taglist window is currently opened, then
set the 'Tlist_Exit_OnlyWindow' variable to 1. By default, this variable is
set to zero and the Vim instance will not be closed if only the taglist window
is present.

        let Tlist_Exit_OnlyWindow = 1

                                                *'Tlist_File_Fold_Auto_Close'*
Tlist_File_Fold_Auto_Close
By default, the tags tree displayed in the taglist window for all the files is
opened. You can close/fold the tags tree for the files manually. To
automatically close the tags tree for inactive files, you can set the
'Tlist_File_Fold_Auto_Close' variable to 1. When this variable is set to 1,
the tags tree for the current buffer is automatically opened and for all the
other buffers is closed.

        let Tlist_File_Fold_Auto_Close = 1

                                            *'Tlist_GainFocus_On_ToggleOpen'*
Tlist_GainFocus_On_ToggleOpen
When the taglist window is opened using the ':TlistToggle' command, this
option controls whether the cursor is moved to the taglist window or remains
in the current window. By default, this option is set to 0 and the cursor
remains in the current window. When this variable is set to 1, the cursor
moves to the taglist window after opening the taglist window.

        let Tlist_GainFocus_On_ToggleOpen = 1

                                            *'Tlist_Highlight_Tag_On_BufEnter'*
Tlist_Highlight_Tag_On_BufEnter
When you enter a Vim buffer/window, the current tag in that buffer/window is
automatically highlighted in the taglist window. If the current tag name is
not visible in the taglist window, then the taglist window contents are
scrolled to make that tag name visible. If you like to disable the automatic
highlighting of the current tag when entering a buffer, you can set the
'Tlist_Highlight_Tag_On_BufEnter' variable to zero. The default setting for
```

this variable is 1.

```
        let Tlist_Highlight_Tag_On_BufEnter = 0
```

                                                'Tlist_Inc_Winwidth'

Tlist_Inc_Winwidth
By default, when the width of the window is less than 100 and a new taglist window is opened vertically, then the window width is increased by the value set in the 'Tlist_WinWidth' variable to accommodate the new window. The value of this variable is used only if you are using a vertically split taglist window.

If your terminal doesn't support changing the window width from Vim (older version of xterm running in a Unix system) or if you see any weird problems in the screen due to the change in the window width or if you prefer not to adjust the window width then set the 'Tlist_Inc_Winwidth' variable to zero. If you are using GNU Screen, you may want to set this variable to zero. CAUTION: If you are using the MS-Windows version of Vim in a MS-DOS command window then you must set this variable to zero, otherwise the system may hang due to a Vim limitation (explained in :help win32-problems)

```
        let Tlist_Inc_Winwidth = 0
```

                                                'Tlist_Max_Submenu_Items'

Tlist_Max_Submenu_Items
If a file contains too many tags of a particular type (function, variable, class, etc.), greater than that specified by the 'Tlist_Max_Submenu_Items' variable, then the menu for that tag type will be split into multiple sub-menus. The default setting for the 'Tlist_Max_Submenu_Items' variable is 25. This can be changed by setting the 'Tlist_Max_Submenu_Items' variable:

```
        let Tlist_Max_Submenu_Items = 20
```

The name of the submenu is formed using the names of the first and the last tag entries in that submenu.

                                                'Tlist_Max_Tag_Length'

Tlist_Max_Tag_Length
Only the first 'Tlist_Max_Tag_Length' characters from the tag names will be used to form the tag type submenu name. The default value for this variable is 10. Change the 'Tlist_Max_Tag_Length' setting if you want to include more or less characters:

```
        let Tlist_Max_Tag_Length = 10
```

                                                'Tlist_Process_File_Always'

Tlist_Process_File_Always
By default, the taglist plugin will generate and process the tags defined in the newly opened files only when the taglist window is opened or when the taglist menu is enabled. When the taglist window is closed, the taglist plugin will stop processing the tags for newly opened files.

You can set the 'Tlist_Process_File_Always' variable to 1 to generate the list of tags for new files even when the taglist window is closed and the taglist menu is disabled.

```
        let Tlist_Process_File_Always = 1
```

To use the ":TlistShowTag" and the ":TlistShowPrototype" commands without the taglist window and the taglist menu, you should set this variable to 1.

                                                'Tlist_Show_Menu'

Tlist_Show_Menu
When using GUI Vim, you can display the tags defined in the current file in a menu named "Tags". By default, this feature is turned off. To turn on this

feature, set the 'Tlist_Show_Menu' variable to 1:

```
        let Tlist_Show_Menu = 1
```

                                                'Tlist_Show_One_File'
Tlist_Show_One_File
By default, the taglist plugin will display the tags defined in all the loaded buffers in the taglist window. If you prefer to display the tags defined only in the current buffer, then you can set the 'Tlist_Show_One_File' to 1. When this variable is set to 1, as you switch between buffers, the taglist window will be refreshed to display the tags for the current buffer and the tags for the previous buffer will be removed.

```
        let Tlist_Show_One_File = 1
```

                                                'Tlist_Sort_Type'
Tlist_Sort_Type
The 'Tlist_Sort_Type' variable specifies the sort order for the tags in the taglist window. The tags can be sorted either alphabetically by their name or by the order of their appearance in the file (chronological order). By default, the tag names will be listed by the order in which they are defined in the file. You can change the sort type (from name to order or from order to name) by pressing the "s" key in the taglist window. You can also change the default sort order by setting 'Tlist_Sort_Type' to "name" or "order":

```
        let Tlist_Sort_Type = "name"
```

                                                'Tlist_Use_Horiz_Window'
Tlist_Use_Horiz_Window
Be default, the tag names are displayed in a vertically split window. If you prefer a horizontally split window, then set the 'Tlist_Use_Horiz_Window' variable to 1. If you are running MS-Windows version of Vim in a MS-DOS command window, then you should use a horizontally split window instead of a vertically split window. Also, if you are using an older version of xterm in a Unix system that doesn't support changing the xterm window width, you should use a horizontally split window.

```
        let Tlist_Use_Horiz_Window = 1
```

                                                'Tlist_Use_Right_Window'
Tlist_Use_Right_Window
By default, the vertically split taglist window will appear on the left hand side. If you prefer to open the window on the right hand side, you can set the 'Tlist_Use_Right_Window' variable to 1:

```
        let Tlist_Use_Right_Window = 1
```

                                                'Tlist_Use_SingleClick'
Tlist_Use_SingleClick
By default, when you double click on the tag name using the left mouse button, the cursor will be positioned at the definition of the tag. You can set the 'Tlist_Use_SingleClick' variable to 1 to jump to a tag when you single click on the tag name using the mouse. By default this variable is set to zero.

```
        let Tlist_Use_SingleClick = 1
```

Due to a bug in Vim, if you set 'Tlist_Use_SingleClick' to 1 and try to resize the taglist window using the mouse, then Vim will crash. This problem is fixed in Vim 6.3 and above. In the meantime, instead of resizing the taglist window using the mouse, you can use normal Vim window resizing commands to resize the taglist window.

                                                'Tlist_WinHeight'
Tlist_WinHeight

The default height of the horizontally split taglist window is 10. This can be changed by modifying the 'Tlist_WinHeight' variable:

```
        let Tlist_WinHeight = 20
```

The |'winfixheight'| option is set for the taglist window, to maintain the height of the taglist window, when new Vim windows are opened and existing windows are closed.

'Tlist_WinWidth'

Tlist_WinWidth
The default width of the vertically split taglist window is 30. This can be changed by modifying the 'Tlist_WinWidth' variable:

```
        let Tlist_WinWidth = 20
```

Note that the value of the |'winwidth'| option setting determines the minimum width of the current window. If you set the 'Tlist_WinWidth' variable to a value less than that of the |'winwidth'| option setting, then Vim will use the value of the |'winwidth'| option.

When new Vim windows are opened and existing windows are closed, the taglist plugin will try to maintain the width of the taglist window to the size specified by the 'Tlist_WinWidth' variable.

---

taglist-commands

7. Commands

The taglist plugin provides the following ex-mode commands:

|:TlistAddFiles|	Add multiple files to the taglist.
|:TlistAddFilesRecursive|	Add files recursively to the taglist.
|:TlistClose|	Close the taglist window.
|:TlistDebug|	Start logging of taglist debug messages.
|:TlistLock|	Stop adding new files to the taglist.
|:TlistMessages|	Display the logged taglist plugin debug messages.
|:TlistOpen|	Open and jump to the taglist window.
|:TlistSessionSave|	Save the information about files and tags in the taglist to a session file.
|:TlistSessionLoad|	Load the information about files and tags stored in a session file to taglist.
|:TlistShowPrototype|	Display the prototype of the tag at or before the specified line number.
|:TlistShowTag|	Display the name of the tag defined at or before the specified line number.
|:TlistHighlightTag|	Highlight the current tag in the taglist window.
|:TlistToggle|	Open or close (toggle) the taglist window.
|:TlistUndebug|	Stop logging of taglist debug messages.
|:TlistUnlock|	Start adding new files to the taglist.
|:TlistUpdate|	Update the tags for the current buffer.

:TlistAddFiles

:TlistAddFiles {file(s)} [file(s) ...]
Add one or more specified files to the taglist. You can specify multiple filenames using wildcards. To specify a file name with space character, you should escape the space character with a backslash.
Examples:

```
    :TlistAddFiles *.c *.cpp
    :TlistAddFiles file1.html file2.html
```

```
                If you specify a large number of files, then it will take some
                time for the taglist plugin to process all of them. The
                specified files will not be edited in a Vim window and will
                not be added to the Vim buffer list.

                                                *:TlistAddFilesRecursive*
:TlistAddFilesRecursive {directory} [ {pattern} ]
                Add files matching {pattern} recursively from the specified
                {directory} to the taglist. If {pattern} is not specified,
                then '*' is assumed. To specify the current directory, use "."
                for {directory}. To specify a directory name with space
                character, you should escape the space character with a
                backslash.
                Examples:

                    :TlistAddFilesRecursive myproject *.java
                    :TlistAddFilesRecursive smallproject

                If large number of files are present in the specified
                directory tree, then it will take some time for the taglist
                plugin to process all of them.

                                                *:TlistClose*
:TlistClose     Close the taglist window. This command can be used from any
                one of the Vim windows.

                                                *:TlistDebug*
:TlistDebug [filename]
                Start logging of debug messages from the taglist plugin.
                If {filename} is specified, then the debug messages are stored
                in the specified file. Otherwise, the debug messages are
                stored in a script local variable. If the file {filename} is
                already present, then it is overwritten.

                                                *:TlistLock*
:TlistLock
                Lock the taglist and don't process new files. After this
                command is executed, newly edited files will not be added to
                the taglist.

                                                *:TlistMessages*
:TlistMessages
                Display the logged debug messages from the taglist plugin
                in a window.  This command works only when logging to a
                script-local variable.

                                                *:TlistOpen*
:TlistOpen      Open and jump to the taglist window. Creates the taglist
                window, if the window is not opened currently. After executing
                this command, the cursor is moved to the taglist window.  When
                the taglist window is opened for the first time, all the files
                in the buffer list are processed and the tags defined in them
                are displayed in the taglist window.

                                                *:TlistSessionSave*
:TlistSessionSave {filename}
                Saves the information about files and tags in the taglist to
                the specified file. This command can be used to save and
                restore the taglist contents across Vim sessions.

                                                *:TlistSessionLoad*
:TlistSessionLoad {filename}
                Load the information about files and tags stored in the
                specified session file to the taglist.
```

```
                                                *:TlistShowPrototype*
:TlistShowPrototype [filename] [linenumber]
                Display the prototype of the tag at or before the specified
                line number. If the file name and the line number are not
                specified, then the current file name and line number are
                used. A tag spans multiple lines starting from the line where
                it is defined to the line before the next tag. This command
                displays the prototype for the tag for any line number in this
                range.

                                                *:TlistShowTag*
:TlistShowTag [filename] [linenumber]
                Display the name of the tag defined at or before the specified
                line number. If the file name and the line number are not
                specified, then the current file name and line number are
                used. A tag spans multiple lines starting from the line where
                it is defined to the line before the next tag. This command
                displays the tag name for any line number in this range.

                                                *:TlistHighlightTag*
:TlistHighlightTag
                Highlight the current tag in the taglist window. By default,
                the taglist plugin periodically updates the taglist window to
                highlight the current tag. This command can be used to force
                the taglist plugin to highlight the current tag.

                                                *:TlistToggle*
:TlistToggle    Open or close (toggle) the taglist window. Opens the taglist
                window, if the window is not opened currently. Closes the
                taglist window, if the taglist window is already opened. When
                the taglist window is opened for the first time, all the files
                in the buffer list are processed and the tags are displayed in
                the taglist window. After executing this command, the cursor
                is not moved from the current window to the taglist window.

                                                *:TlistUndebug*
:TlistUndebug
                Stop logging of debug messages from the taglist plugin.

                                                *:TlistUnlock*
:TlistUnlock
                Unlock the taglist and start processing newly edited files.

                                                *:TlistUpdate*
:TlistUpdate    Update the tags information for the current buffer. This
                command can be used to re-process the current file/buffer and
                get the tags information. As the taglist plugin uses the file
                saved in the disk (instead of the file displayed in a Vim
                buffer), you should save a modified buffer before you update
                the taglist. Otherwise the listed tags will not include the
                new tags created in the buffer. You can use this command even
                when the taglist window is not opened.
```

```
                                                *taglist-functions*
8. Global functions

The taglist plugin provides several global functions that can be used from
other Vim plugins to interact with the taglist plugin. These functions are
described below.

|Tlist_Get_Filenames()|                 Return filenames in the taglist
|Tlist_Update_File_Tags()|              Update the tags for the specified file
|Tlist_Get_Tag_Prototype_By_Line()|     Return the prototype of the tag at or
```

	before the specified line number in the specified file.
|Tlist_Get_Tagname_By_Line()|	Return the name of the tag at or before the specified line number in the specified file.
|Tlist_Set_App()|	Set the name of the application controlling the taglist window.

Tlist_Get_Filenames()

Tlist_Get_Filenames()

Returns a list of filenames in the taglist. Each filename is separated by a newline (\n) character. If the taglist is empty an empty string is returned.

Tlist_Update_File_Tags()

Tlist_Update_File_Tags({filename}, {filetype})

Update the tags for the file {filename}. The second argument specifies the Vim filetype for the file. If the taglist plugin has not processed the file previously, then the exuberant ctags tool is invoked to generate the tags for the file.

Tlist_Get_Tag_Prototype_By_Line()

Tlist_Get_Tag_Prototype_By_Line([{filename}, {linenumber}])

Return the prototype of the tag at or before the specified line number in the specified file. If the filename and line number are not specified, then the current buffer name and the current line number are used.

Tlist_Get_Tagname_By_Line()

Tlist_Get_Tagname_By_Line([{filename}, {linenumber}])

Return the name of the tag at or before the specified line number in the specified file. If the filename and line number are not specified, then the current buffer name and the current line number are used.

Tlist_Set_App()

Tlist_Set_App({appname})

Set the name of the plugin that controls the taglist plugin window and buffer. This can be used to integrate the taglist plugin with other Vim plugins.

For example, the winmanager plugin and the Cream package use this function and specify the appname as "winmanager" and "cream" respectively.

By default, the taglist plugin is a stand-alone plugin and controls the taglist window and buffer. If the taglist window is controlled by an external plugin, then the appname should be set appropriately.

---

taglist-extend

9. Extending

The taglist plugin supports all the languages supported by the exuberant ctags tool, which includes the following languages: Assembly, ASP, Awk, Beta, C, C++, C#, Cobol, Eiffel, Erlang, Fortran, HTML, Java, Javascript, Lisp, Lua, Make, Pascal, Perl, PHP, Python, Rexx, Ruby, Scheme, Shell, Slang, SML, Sql, TCL, Verilog, Vim and Yacc.

You can extend the taglist plugin to add support for new languages and also modify the support for the above listed languages.

You should NOT make modifications to the taglist plugin script file to add support for new languages. You will lose these changes when you upgrade to the next version of the taglist plugin. Instead you should follow the below described instructions to extend the taglist plugin.

You can extend the taglist plugin by setting variables in the .vimrc or _vimrc file. The name of these variables depends on the language name and is described below.

Modifying support for an existing language
To modify the support for an already supported language, you have to set the tlist_xxx_settings variable in the ~/.vimrc or $HOME/_vimrc file. Replace xxx with the Vim filetype name for the language file. For example, to modify the support for the perl language files, you have to set the tlist_perl_settings variable. To modify the support for java files, you have to set the tlist_java_settings variable.

To determine the filetype name used by Vim for a file, use the following command in the buffer containing the file:

```
        :set filetype
```

The above command will display the Vim filetype for the current buffer.

The format of the value set in the tlist_xxx_settings variable is

```
    <language_name>;flag1:name1;flag2:name2;flag3:name3
```

The different fields in the value are separated by the ';' character.

The first field 'language_name' is the name used by exuberant ctags to refer to this language file. This name can be different from the file type name used by Vim. For example, for C++, the language name used by ctags is 'c++' but the filetype name used by Vim is 'cpp'. To get the list of language names supported by exuberant ctags, use the following command:

```
        $ ctags --list-maps=all
```

The remaining fields follow the format "flag:name". The sub-field 'flag' is the language specific flag used by exuberant ctags to generate the corresponding tags. For example, for the C language, to list only the functions, the 'f' flag is used. To get the list of flags supported by exuberant ctags for the various languages use the following command:

```
        $ ctags --list-kinds=all
```

The sub-field 'name' specifies the title text to use for displaying the tags of a particular type. For example, 'name' can be set to 'functions'. This field can be set to any text string name.

For example, to list only the classes and functions defined in a C++ language file, add the following line to your .vimrc file:

```
        let tlist_cpp_settings = 'c++;c:class;f:function'
```

In the above setting, 'cpp' is the Vim filetype name and 'c++' is the name used by the exuberant ctags tool. 'c' and 'f' are the flags passed to exuberant ctags to list C++ classes and functions and 'class' is the title used for the class tags and 'function' is the title used for the function tags in the taglist window.

For example, to display only functions defined in a C file and to use "My Functions" as the title for the function tags, use

```
        let tlist_c_settings = 'c;f:My Functions'
```

When you set the tlist_xxx_settings variable, you will override the default setting used by the taglist plugin for the 'xxx' language. You cannot add to the default options used by the taglist plugin for a particular file type. To add to the options used by the taglist plugin for a language, copy the option values from the taglist plugin file to your .vimrc file and modify it.

Adding support for a new language
If you want to add support for a new language to the taglist plugin, you need to first extend the exuberant ctags tool. For more information about extending exuberant ctags, visit the following page:

```
    http://ctags.sourceforge.net/EXTENDING.html
```

To add support for a new language, set the tlist_xxx_settings variable in the ~/.vimrc file appropriately as described above. Replace 'xxx' in the variable name with the Vim filetype name for the new language.

For example, to extend the taglist plugin to support the latex language, you can use the following line (assuming, you have already extended exuberant ctags to support the latex language):

```
        let tlist_tex_settings='latex;b:bibitem;c:command;l:label'
```

With the above line, when you edit files of filetype "tex" in Vim, the taglist plugin will invoke the exuberant ctags tool passing the "latex" filetype and the flags b, c and l to generate the tags. The text heading 'bibitem', 'command' and 'label' will be used in the taglist window for the tags which are generated for the flags b, c and l respectively.

---

                                                taglist-faq
10. Frequently Asked Questions

Q. The taglist plugin doesn't work. The taglist window is empty and the tags defined in a file are not displayed.
A. Are you using Vim version 6.0 and above? The taglist plugin relies on the features supported by Vim version 6.0 and above. You can use the following command to get the Vim version:

```
      $ vim --version
```

Are you using exuberant ctags version 5.0 and above? The taglist plugin relies on the features supported by exuberant ctags and will not work with GNU ctags or the Unix ctags utility. You can use the following command to determine whether the ctags installed in your system is exuberant ctags:

```
      $ ctags --version
```

Is exuberant ctags present in one of the directories in your PATH? If not, you need to set the Tlist_Ctags_Cmd variable to point to the location of exuberant ctags. Use the following Vim command to verify that this is setup correctly:

```
      :echo system(Tlist_Ctags_Cmd . ' --version')
```

The above command should display the version information for exuberant ctags.

Did you turn on the Vim filetype detection? The taglist plugin relies on the filetype detected by Vim and passes the filetype to the exuberant ctags utility to parse the tags. Check the output of the following Vim command:

```
      :filetype
```

The output of the above command should contain "filetype detection:ON".
To turn on the filetype detection, add the following line to the .vimrc or
_vimrc file:

```
     filetype on
```

Is your version of Vim compiled with the support for the system() function?
The following Vim command should display 1:

```
     :echo exists('*system')
```

In some Linux distributions (particularly Suse Linux), the default Vim
installation is built without the support for the system() function. The
taglist plugin uses the system() function to invoke the exuberant ctags
utility. You need to rebuild Vim after enabling the support for the
system() function. If you use the default build options, the system()
function will be supported.

Do you have the |'shellslash'| option set? You can try disabling the
|'shellslash'| option. When the taglist plugin invokes the exuberant ctags
utility with the path to the file, if the incorrect slashes are used, then
you will see errors.

Check the shell related Vim options values using the following command:

```
     :set shell? shellcmdflag? shellpipe?
     :set shellquote? shellredir? shellxquote?
```

If these options are set in your .vimrc or _vimrc file, try removing those
lines.

Are you using a Unix shell in a MS-Windows environment? For example,
the Unix shell from the MKS-toolkit. Do you have the SHELL environment
set to point to this shell? You can try resetting the SHELL environment
variable.

If you are using a Unix shell on MS-Windows, you should try to use
exuberant ctags that is compiled for Unix-like environments so that
exuberant ctags will understand path names with forward slash characters.

Is your filetype supported by the exuberant ctags utility? The file types
supported by the exuberant ctags utility are listed in the ctags help. If a
file type is not supported, you have to extend exuberant ctags. You can use
the following command to list the filetypes supported by exuberant ctags:

```
     ctags --list-languages
```

Run the following command from the shell prompt and check whether the tags
defined in your file are listed in the output from exuberant ctags:

```
     ctags -f - --format=2 --excmd=pattern --fields=nks <filename>
```

If you see your tags in the output from the above command, then the
exuberant ctags utility is properly parsing your file.

Do you have the .ctags or _ctags or the ctags.cnf file in your home
directory for specifying default options or for extending exuberant ctags?
If you do have this file, check the options in this file and make sure
these options are not interfering with the operation of the taglist plugin.

If you are using MS-Windows, check the value of the TEMP and TMP
environment variables. If these environment variables are set to a path
with space characters in the name, then try using the DOS 8.3 short name
for the path or set them to a path without the space characters in the
name. For example, if the temporary directory name is "C:\Documents and

```
    Settings\xyz\Local Settings\Temp", then try setting the TEMP variable to
    the following:

         set TEMP=C:\DOCUMEN~1\xyz\LOCALS~1\Temp

    If exuberant ctags is installed in a directory with space characters in the
    name, then try adding the directory to the PATH environment variable or try
    setting the 'Tlist_Ctags_Cmd' variable to the shortest path name to ctags
    or try copying the exuberant ctags to a path without space characters in
    the name. For example, if exuberant ctags is installed in the directory
    "C:\Program Files\Ctags", then try setting the 'Tlist_Ctags_Cmd' variable
    as below:

         let Tlist_Ctags_Cmd='C:\Progra~1\Ctags\ctags.exe'

    If you are using a cygwin compiled version of exuberant ctags on MS-Windows,
    make sure that either you have the cygwin compiled sort utility installed
    and available in your PATH or compile exuberant ctags with internal sort
    support. Otherwise, when exuberant ctags sorts the tags output by invoking
    the sort utility, it may end up invoking the MS-Windows version of
    sort.exe, thereby resulting in failure.

Q. When I try to open the taglist window, I am seeing the following error
   message. How do I fix this problem?

   Taglist: Failed to generate tags for /my/path/to/file
   ctags: illegal option -- -^@usage: ctags [-BFadtuwvx] [-f tagsfile] file ...

A. The taglist plugin will work only with the exuberant ctags tool. You
   cannot use the GNU ctags or the Unix ctags program with the taglist plugin.
   You will see an error message similar to the one shown above, if you try
   use a non-exuberant ctags program with Vim. To fix this problem, either add
   the exuberant ctags tool location to the PATH environment variable or set
   the 'Tlist_Ctags_Cmd' variable.

Q. A file has more than one tag with the same name. When I select a tag name
   from the taglist window, the cursor is positioned at the incorrect tag
   location.
A. The taglist plugin uses the search pattern generated by the exuberant ctags
   utility to position the cursor at the location of a tag definition. If a
   file has more than one tag with the same name and same prototype, then the
   search pattern will be the same. In this case, when searching for the tag
   pattern, the cursor may be positioned at the incorrect location.

Q. I have made some modifications to my file and introduced new
   functions/classes/variables. I have not yet saved my file. The taglist
   plugin is not displaying the new tags when I update the taglist window.
A. The exuberant ctags utility will process only files that are present in the
   disk. To list the tags defined in a file, you have to save the file and
   then update the taglist window.

Q. I have created a ctags file using the exuberant ctags utility for my source
   tree. How do I configure the taglist plugin to use this tags file?
A. The taglist plugin doesn't use a tags file stored in disk. For every opened
   file, the taglist plugin invokes the exuberant ctags utility to get the
   list of tags dynamically. The Vim system() function is used to invoke
   exuberant ctags and get the ctags output. This function internally uses a
   temporary file to store the output. This file is deleted after the output
   from the command is read. So you will never see the file that contains the
   output of exuberant ctags.

Q. When I set the |'updatetime'| option to a low value (less than 1000) and if
   I keep pressing a key with the taglist window open, the current buffer
   contents are changed. Why is this?
A. The taglist plugin uses the |CursorHold| autocmd to highlight the current
```

```
   tag. The CursorHold autocmd triggers for every |'updatetime'| milliseconds.
   If the |'updatetime'| option is set to a low value, then the CursorHold
   autocmd will be triggered frequently. As the taglist plugin changes
   the focus to the taglist window to highlight the current tag, this could
   interfere with the key movement resulting in changing the contents of
   the current buffer. The workaround for this problem is to not set the
   |'updatetime'| option to a low value.
```

```
                                                *taglist-license*
11. License
Permission is hereby granted to use and distribute the taglist plugin, with or
without modifications, provided that this copyright notice is copied with it.
Like anything else that's free, taglist.vim is provided *as is* and comes with
no warranty of any kind, either expressed or implied. In no event will the
copyright holder be liable for any damamges resulting from the use of this
software.
```

```
                                                *taglist-todo*
12. Todo

1. Group tags according to the scope and display them. For example,
   group all the tags belonging to a C++/Java class
2. When opening a new Vim tab, automatically open the taglist window.
3. Support for displaying tags in a modified (not-yet-saved) file.
4. Automatically open the taglist window only for selected filetypes.
   For other filetypes, close the taglist window.
5. When using the shell from the MKS toolkit, the taglist plugin
   doesn't work.
6. The taglist plugin doesn't work with files edited remotely using the
   netrw plugin. The exuberant ctags utility cannot process files over
   scp/rcp/ftp, etc.
```
