

FL Package of BBEdit Extensions

The FL Package contains four Extensions

- **Function Lister**
- **Fast Search**
- **Word Completion**
- **Function Lookup**

Function Lister is a utility for programmers which displays information about functions defined in a source file;

Fast Search is an alternative to BBEdit's regular search, a fast incremental search which finds as the user types characters;

Word Completion enables the user to type a few letters of a variable name and have the extension type the rest;

Function Lookup is a convenient method of accessing Think Reference or MPW's 411.

Because **Function Lookup** requires the "Insert ToolBox Template" command of BBEdit, it is the only extension of the package that is not usable with BBEdit Lite. All four extensions work with BBEdit 2.5.

BBEdit 2.5 allows the assignment of cmd keys to extensions, and the four FL extensions come prebound to ⌘F11, ⌘F9, ⌘F12 and ⌘F10 respectively. For BBEdit Lite, I highly recommend the BBEdit extension BBXKeys, written by Jamie McCarthy, available at the usual sites.

The FL Package is distributed as ShareWare, and is Copyright © 1993, 1994, Paul van Mulbregt, All rights reserved.

If, after trying these extensions for 30 days, you decide to continue using them, you should register your copy by sending \$15 to

Paul van Mulbregt, 33 Pine Ridge Road, Wayland, MA 01778, USA.

A registration form is included as the last page of this documentation. Site licenses are available.

Registered Users receive automatic updates (via email)

Questions or Comments are best sent via email to:

Internet : paulvm@dragonsys.com

The FL Package may be distributed freely, provided that the software, including this documentation, is unchanged. The FL Package may not be sold or offered for sale, or included with another software product offered for sale, except with the express written permission of the author.

Companies that distribute public domain/freeware/shareware software for profit should contact me.

This restriction does not apply to bulletin boards, commercial on-line services such as America Online, CompuServe and GENie, and non-profit Macintosh user groups which hold regularly scheduled public meetings.

Legalities: The FL Package is distributed as is, with no guarantee that it is bug free or compatible with all hardware and software configurations. The author is not responsible for losses incurred through the uses or malfunction of this program.

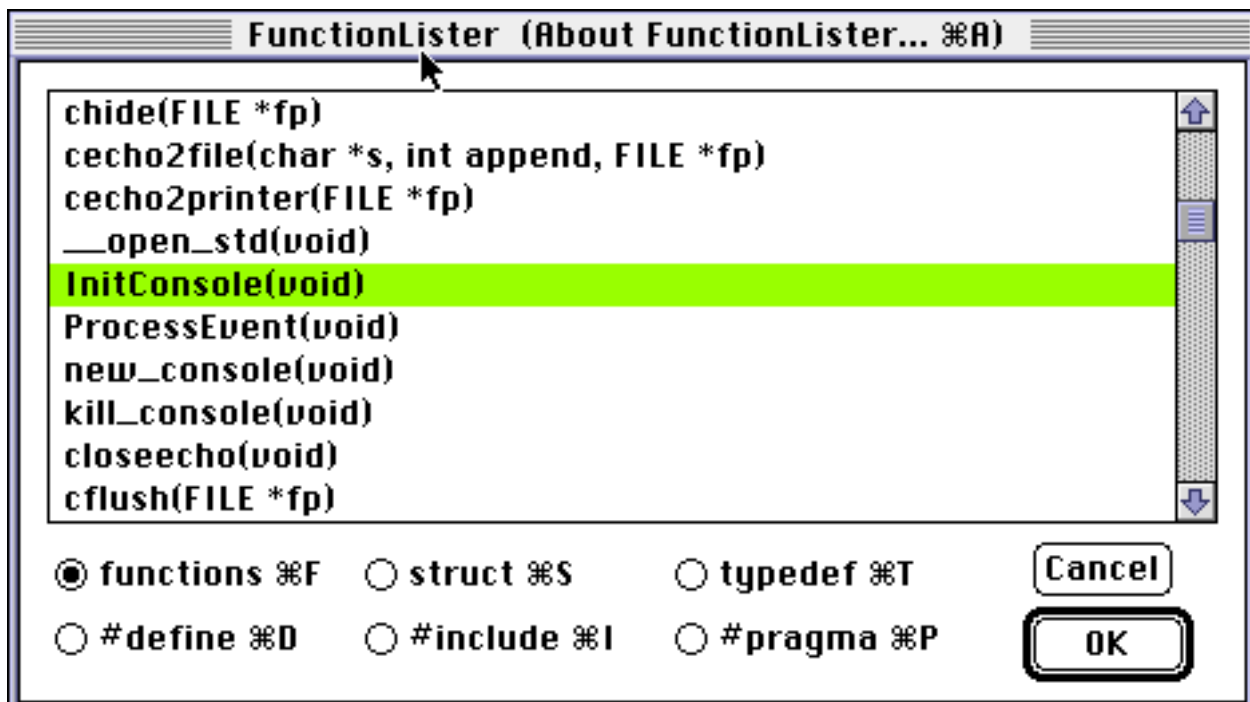
Function Lister

Function Lister displays information about functions defined in C, C++ and Pascal files, and information about resources in Rez files.

For C and C++ files, the functions, structs and typedefs are displayed by **Function Lister**, as well as any #include files, #defines and #pragmas.

For Rez files, the resources, data entries, #defines, #includes, include's and types are displayed. For Pascal files, the functions and procedures, consts, types, compiler directives, units used are displayed.

For example, opening <console.c> from Symantec's Standard C Libraries and choosing the **Function Lister** BBXT brings up a movable modal dialog box with a list of the functions defined in <console.c>.



To view the definition of a function, click once on the function name in the list. The file automatically scrolls to that function and highlights its name and argument list. Highlighting appears as a box drawn around the function, in the current highlighting colour. The file scrolls so that the selected function appears as the third line from the top of the window. (One exception to this rule occurs if the function is in the last page of the file — then BBEdit will not scroll the file all the way.)

To dismiss the dialog, click on the Cancel button, which leaves the file exactly as it was before the extension ran; or click on the OK button, which scrolls the file to that function, and leaves the selection point at the start of the function definition.

Double-clicking on a function in the list accomplishes the same as single-clicking and choosing OK.

The arrow keys also navigate the list, as do PageUp, PageDown, Home and End. The arrow keys change the selected item in the list, while PageUp, PageDown etc. scroll the list but leave the currently selected item unchanged.

To see the other items of interest in a C source file, click on one of the radio buttons.

Radio Button	Cmd-Key equivalent	Displays
functions	F	functions
struct	S	structs
typedef	T	typedefs
#include	I	#includes
#define	D	#defines, #undefs
#pragma	P	#pragma's
	A	About Box

Cmd-A will display the About Box for Function Lister. Any of the above Command key combinations will hide the About Item and return to displaying information.

All information above about scrolling, selecting for the functions list is also true for all the other lists, with one exception. If the dialog is dismissed and an element of the **#include** list has been selected, then the include file is actually opened, just as if the user had selected the text inside the <> or "" and chosen "Open Selection" from BBEEdit.

The dialog box is a movable modal dialog. Hence it can be dragged around the desktop and placed wherever it is desired.

Windows behind the dialog can be moved by holding down the Command key and dragging the window by its title bar.

For a C++ source file (ending in .cpp, .cp or .cxx), the extra constructs of C++ are considered as well. Classes appear together with the structs. Header files with extension .h are parsed as C++.

Rez files are those ending in .r. Two of the radio buttons in the dialog change titles — **functions** becomes **resources**, and **#pragma** becomes **other**.

Resources and data show up in the **resources** list, while **other** covers include, read, change delete etc. Types appear as structs, and "type xxx as yyy" will appear in the typedefs list.

The file extensions determining the type of file are stored in an `STRe' resource. While there is no way to change these from Function Lister, a Resorcerer® Template is provided, so adding or deleting from these file extensions is possible.

If the type of a file cannot be determined by its name, a dialog will appear asking the user to specify the type of file: C, C+, Pascal or Rez.

Location of the **Function Lister** Dialog Box

The dialog box will be positioned on the same monitor as the frontmost window. If there is room above or below or to the side of this window, then that is where it will be placed. Otherwise it will be at the bottom of the screen.

Known Anomalies:

- With nested structs or classes, BBEEdit's algorithm for highlighting text in a non-active window sometimes leaves some residue on the screen.
- With a split window, and the selection point in the lower half of the window, scrolling doesn't always occur in some versions of BBEEdit.
- **Function Lister** is not a C compiler nor a C preprocessor - nor is that desirable. One consequence of this is that by with "unusual" use of macros, it is possible to confuse **Function Lister**. Unusual here means, for example, a macro which looks like a function but in reality is declaring a struct. Without having access to all the defines in header files and at the make level (MPW makefile, Think C project etc) it is not possible to exactly determine the nature of the code. **Function Lister** does what can be done in the absence of this knowledge, and in the process is able to show all functions and structs, not just those which are being compiled with the current set of #defines.

How is it different from PopUpFuncs?

PopUpFuncs from SciComp Software is also a productivity tool. A demo of PopUpFuncs is shipped with BBEEdit.

PopUpFuncs also displays a list of functions and typedefs in a C file, but it does so by installing a WDEF in BBEEdit itself. This may cause a conflict with other utilities such as James Walker's Text Editor Patches.

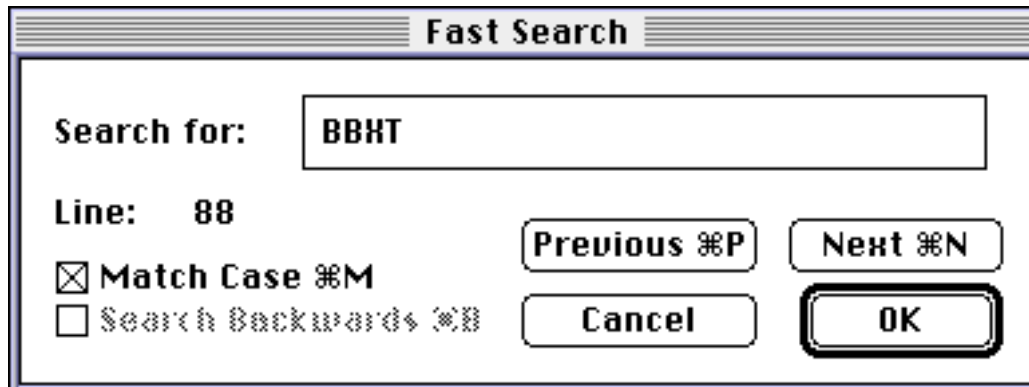
PopUpFuncs works with MPW and BBEEdit, parses Assembly files in addition to C, C++ Pascal and Rez files, and is quick for short files.

- What PopUpFuncs doesn't do is display information about structs, #includes, #defines, #pragmas.
- PopUpFuncs doesn't parse C++ files correctly.
- PopUpFuncs is less robust than **Function Lister** to code which makes uses of C macros.
- For long files, **Function Lister** is faster than PopUpFuncs, and more convenient to use than a very long drop-down list of functions.
- **Function Lister** enables the user to browse the file, viewing just the interesting functions, without having to traverse a list each time to get to the next function.
- **Function Lister** is also cheaper.

Fast Search

Fast Search is an alternative to the regular BBEdit search. By restricting the search to the immediate file, it is possible to search as the letters are typed in.

The **Fast Search** dialog is a movable modal dialog box and looks like



Entering a Search String:

Characters are appended to the Search String as they are typed — they appear in the “**Search for:**” item. After each character is entered the file is searched for the new string, the search starting at the current location. The file is then scrolled to display that match as the third line from the top of the file.

The line number of a match is displayed. In the above example, “BBXT” was found on line 88 of the window.

If a match is not found, then the file is scrolled to the end (or beginning, depending on the search direction).

Hitting the backspace key deletes the last letter, and returns to the previous match of the shorter string.

To find the next occurrence of the same search string, just select the **Next** button. For the previous match, select the **Previous** button.

Note that this search string is not an editText item, and cannot be edited in the usual manner for editText items.

Search Options

Case-sensitivity:

It is possible to search case-sensitively or case-insensitively, by appropriately checking the check-box labelled "**Match Case**". The default is to match case exactly.

Search Direction:

The default search direction is forward, although at the start of a search, the "**Search Backwards**" checkbox is active. Unless this checkbox is checked, typing any characters will initiate a forward search. The "**Search Backwards**" checkbox may only be checked if there are no characters in the search string. It is always possible however, to search in the other direction by using **Previous** and **Next**. (Note that **Previous** and **Next** refer to the search direction, so selecting **Previous** while searching backwards would actually move forwards in the file.)

What do I do now that I've found it?

To dismiss the dialog, select the OK button (or hit the return key), and the dialog will disappear, and the selection point in the file will be that of the last match.

Selecting Cancel (or Cmd-.) will dismiss the dialog and return the file in its previous state.

Location of the **Fast Search** Dialog Box

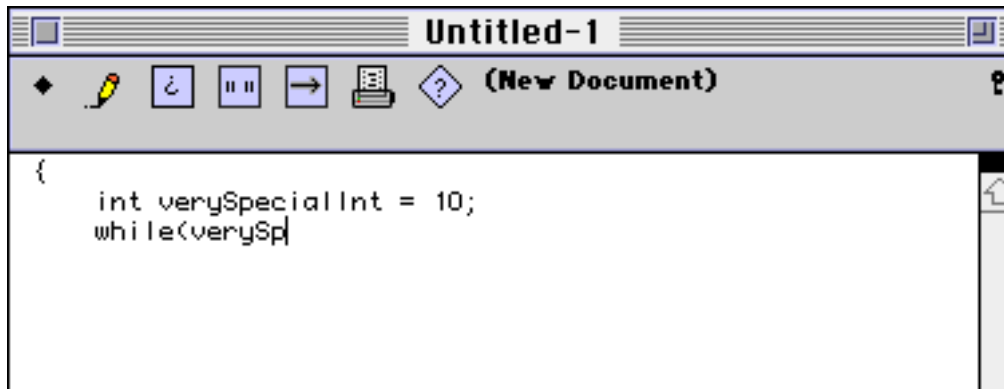
The dialog box will be positioned on the same monitor as the frontmost window. If there is room above or below or to the side of this window, then that is where it will be placed. Otherwise it will be at the bottom of the screen.

Known Anomalies:

- With a split window, and the selection point in the lower half of the window, scrolling doesn't always occur in some versions of BBEEdit.

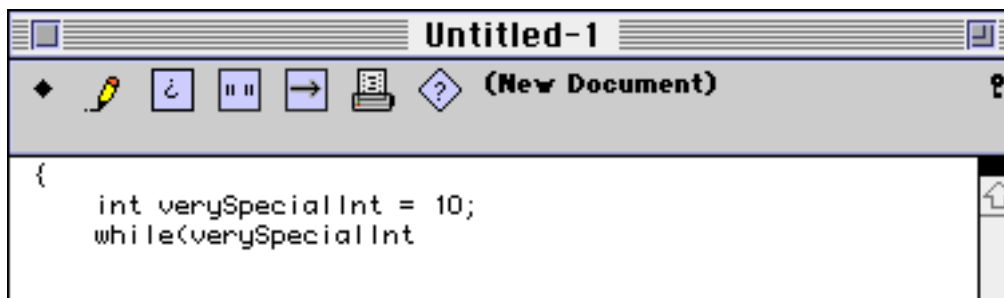
Word Completion

This extension saves typing, as well as making sure variable names are correct. While typing the following C code, there is no need to type all of the second occurrence of `verySpecialInt`. One could copy and paste, but another way is to type a few letters, and select the **Word Completion** Extension.



```
{
    int verySpecialInt = 10;
    while(verySp
```

becomes



```
{
    int verySpecialInt = 10;
    while(verySpecialInt
```

Word Completion will look back in the code to find the first match and then extend the current occurrence to match the previous occurrence. If a match is not found looking backwards, then it looks forwards. If not found forwards, the word is selected. This is a quick way to save on typing, good for helping prevent RSI, but perhaps more importantly, to make sure that variable names are spelt correctly.

There is no user interface for **Word Completion**.

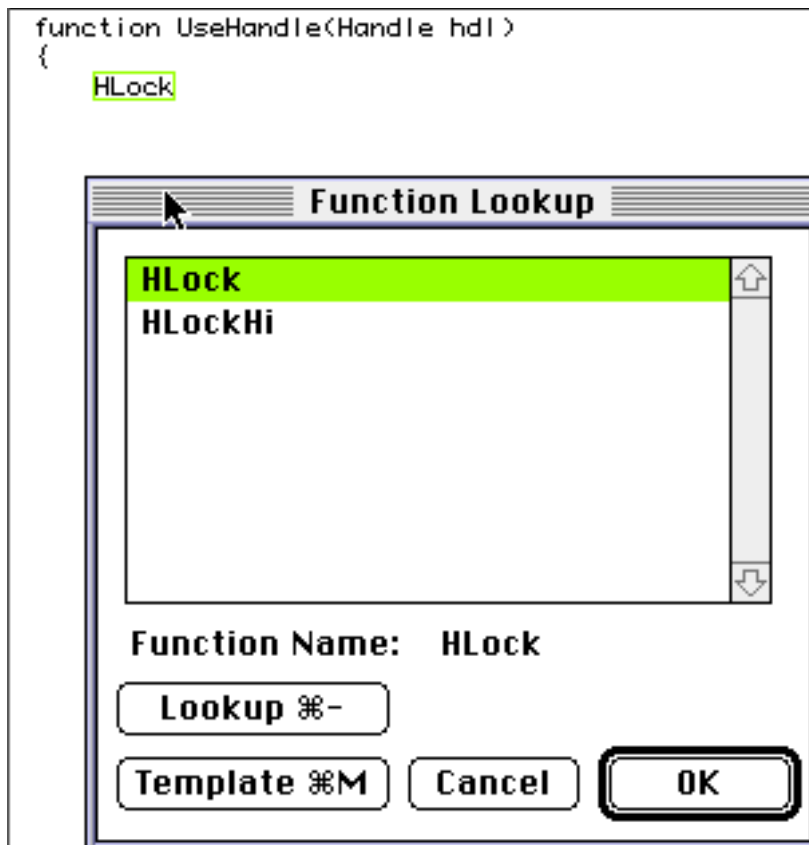
The usefulness of this extension is greatly enhanced if the extension is bound to a key!

Function Lookup

Function Lookup allows the user to insert a function name and/or template in a file without typing the whole name, or remembering the argument list

Type a few letters of a function. Select the **Function Lookup** extension -- it will try to match those first few letters against a list of ToolBox traps and functions from the Standard C library. A dialog box with a list of potential matches will then be displayed.

For example, typing HLock and then selecting **Function Lookup** brings up a dialog box and a list with two elements, HLock and HLockHi, since there are only two ToolBox traps which start HLock.



At this stage there are several choices.

If HLock, the highlighted element of the list, is the desired ToolBox call, then one can

1. Paste a template for HLock (Cmd M)
2. Lookup the entry for HLock (Cmd -)
3. Cancel, which leaves the file the way it was
4. OK, which adds () and leaves the selection in the middle of the parentheses

All ToolBox Lookups and Template Pastings are done through either MPW 411 or Think Reference, depending on which one the user has chosen in BBEdit. (Hence this extension requires either Think Reference or 411!)

For example, with Think Reference as the selected ToolBox reference, the above four options leave the text like

1. HLock(theHandle); /* IM II pg 41 */ (I-beam cursor at end of line)
2. HLock (switched to Think Reference)
3. HLock (no change)
4. HLock() (I-beam cursor in between '(' and ')')

If, on the other hand, HLock is not the desired ToolBox call, then one can

1. scroll through the list to find the correct call; or,
2. type more letters to narrow down the search; or,
3. press delete to remove letters and widen the search.

HLock is a short name and easy to remember trap and used above as an example only. The real use of this extension is with traps which have many letters in common with other traps. For example, there are about 60 AppleEvent calls which start with the letters AE. Using this extension is a quick way to trim the search for some AppleEvent call, and find an argument list.

All the functions from the C standard libraries are also available.

Not all of the functions known about by **Function Lookup** may be present in the database used by 411 or Think Reference.

(If I missed any out, please let me know!)

Lookup and Make Template have the same command key shortcuts as used in BBEdit 2.5

FL Package v1.1 registration form.

Please complete this form and mail it to: Paul van Mulbregt 33 Pine Ridge Road
Wayland, MA 01778
U.S.A

My internet address is paulvm@dragonsys.com

Date:

Your name:

Your Postal Address:

E-Mail:

Method of Payment:

(Check / Postal Money Order / Cash :)

Number of Copies:

Amount (US\$15 - or equivalent) :

Where did you get FL Package v1.0 ? :

Was the documentation accompanying the software ? : Yes

Your Mac is a: (e.g. IIfx, Quadra 650)

Version of System: (e.g. 6.0.5, 7.1)

Would you like to recieve automatic updates or just notification of updates?

(Automatic, Notification, Neither)

Have you any bugs to report or suggestions?