CODEHEAD SOFTWARE

Doug Harrison's

LOOKIT & POPIT

Two Powerful Utilities for the Price of One!

CodeHead Software

LOOKIT & POPIT

Reference Manual

Version 1.0

Software by **Doug Harrison**

Manual by **Doug Harrison**

Cover Art and Graphics by **Maurice Molyneaux**

Layout and Editing by **Charles F. Johnson**

1st Printing

LOOKIT & POPIT Contents

LOOKIT

POPIT

Introduction LOOKIT & POPIT

Thank you for purchasing LookIt! and PopIt! These two powerful programs will quickly become indispensable tools in your utility collection. Although both programs are simple and intuitive to operate, we recommend that you take at least a quick look through this manual to familiarize yourself with their basic features; you'll enjoy the programs much more if you do.

But even before reading the rest of the manual, you should first make a backup copy of your LookIt! & PopIt! master disk, either by copying the entire disk or by copying the individual files (and folders) on the disk to another disk. Once you've made the copy, put the original away in a safe place, and *work from the backup copy.*

Also, please take a moment to fill out and return the registration card included with this package. This will enable us to enter you properly in our registered user database.

• REGACC.PRG

To work properly, both LookIt! and PopIt! require that you install the program called REGACC.PRG (located on your master disk) in a folder called AUTO on the disk with which you boot up. If you don't already have such a folder, simply create one and copy REGACC.PRG to it. If you do have an existing AUTO folder with programs already in it, just copy REGACC.PRG to the folder (the order is not important). After placing it in the AUTO folder, REGACC will be installed the next time you boot your computer.

Essential Concepts

• The File List

The central LookIt! concept is the file list. The file list may contain up to 32 files at any given time. You can step forward and backward through the list, or you can choose a file directly from a scrollable list containing the filenames.

• The Display

LookIt! has two display screens; the "GEM screen" provides a menu bar, while one views files from the "View screen." Assuming use of the standard monitors, the View screen provides a full 80 columns, and in monochrome, it supports the 50 line mode. LookIt! requires a graphics mode resolution of at least 640x200 pixels, which is the ST medium resolution mode.

• View Mode

Since LookIt! supports viewing of both ASCII and binary files, each file has an associated type, which controls the view mode.

• Opening a File

To open a file, select File Open from the menu bar. You will be presented with a custom file selector, which is a bit different from the GEM item selector and other replacements. See "The File Selector" for details. Selecting a file works exactly as with the GEM item selector, except that you may select multiple files, adding them to the file list. Since the file list only has room for 32 entries, any files above this limit are ignored; that is, if you select 100 files, LookIt! will only acknowledge the first 32. However, if there isn't enough space in the file list for the files LookIt! does acknowledge, you will be asked to remove enough files from the list to make room for the new ones (this applies to the *number* of files, not the memory they require; LookIt! handles memory management itself).

LookIt! provides several standard file extensions, and it automatically opens files with these extensions in the proper view mode, Text or Binary. Files that do not match any of the extensions are opened in Text mode by default. You can override this as follows. To open any file in Text mode, press the right Shift key when you double-click on the file or press Return to close the file selector. To open a file in Binary mode, press the left Shift key. Be aware that if you use the Shift keys, all selected files will be assigned the indicated mode, even if they match the built-in extensions.

LookIt! also provides "Open Binary File" and "Open Text File" menu items. When you choose one of these, LookIt! opens files in the indicated mode, unless they match one of the built-in file extensions, in which case they are opened in the default mode for the extension. To override the default, use the Shift keys as described above.

The result of all this is that you can select several files, say, some .PRG, some .S, some .TXT, and some .ACC, and they will all be assigned the proper mode, unless you have chosen to override the default by pressing the shift keys.

Tip: you will notice the three File Open possibilities each provide a different set of extensions. For example, File Open Binary provides PRG, TTP, TOS, ACC, APP, O, OVL, and * in the dialog box. File Open itself is a "general" command, and it provides a mix of text and binary file extensions. The 'O' key is the keyboard equivalent for File Open. Left-Shift-O is equivalent to File Open Binary, and right-Shift-O maps to File Open Text.

Viewing Files

• File Viewing Quick Start

Assuming you've already opened a file, you will be in the View screen. There are several ways to move around the file. They are as follows:

Up arrow/right mouse button	line up
Down arrow/left mouse button	line down
Return/Shift up arrow	page up
Spacebar/Shift down arrow	page down
Left arrow	column left
Right arrow	column right
Shift left arrow	10 columns left
Shift right arrow	10 columns right
Control left arrow	to first column
Clr Home	to first line
Shift Clr Home	to last line
G	invoke the Go To dialog box
Esc/Undo/Left+Right Click	return to the GEM screen
Control-Q	quit LookIt! immediately

Except for the column movement commands, these methods apply to both Binary and Text files. When in Text mode, the Go To dialog box expects a decimal line number. While in Binary mode, it expects an offset into the file. This offset by default is a decimal number of bytes; you may enter a hexadecimal number, provided you preface it with "$" or "0x".

While in the View screen, press both mouse buttons simultaneously or press Esc or Undo to return to the GEM screen. To quit the program from either the GEM or the View screen, press Control-Q.

While in Text mode, LookIt! updates the Line Counter at the bottom of the screen. This counter always refers to the first displayed line on the current screen. Binary mode has no Line Counter for obvious reasons; the size of the file, in decimal, is displayed instead.

Other handy View Screen features:

+ or -......Increment/decrement the spaces per tab stop value. The range is 0-9, and if no tabs are present, or you are viewing a binary file, a bell sounds.

F1..........Selects a group of lines as a block. To begin the block, press F1. The mouse pointer appears, letting you drag the mouse over the lines you wish to form the block.

S-F1.......Shift+F1 allows you to extend a defined block.

F2..........Performs block deletion for text files only. The Delete key is identical to F2.

F3..........Allows you to save a block to a new file

F5..........Allows you to print the entire file, a block, or what's currently on screen.

F7..........Changes view mode from binary to text and vice versa.

F8..........In binary viewing mode, F8 changes the row/column offsets from decimal to hexadecimal and vice versa.

F9.........Exchanges foreground and background colors.

F10........In high resolution modes, changes between the small 8x8 font and the regular 8x16 system font.

E...........In binary mode, this allows you to edit the file.

S...........Invokes the Search dialog. When viewing text files, you are only allowed to enter an ASCII string or regular expression. However, when viewing binary files, you can enter a set of comma-separated decimal or hexadecimal bytes as well as an ASCII string. Be sure to select the proper Mode button before pressing Return.

F...........Invokes the File List dialog box.

O...........Invokes the File Selector. O is the same as File Open; use Left-Shift-O for File Open Binary and Right-Shift-O for File Open Text.

M..........Displays the BookMarks popup menu. This dialog box allows you to save and later jump to up to four positions within the file you are viewing.

1,2,3,4.....These numeric keys allow you to jump to the corresponding bookmark, providing you have saved it via the BookMarks menu (these keys don't invoke the popup menu).

N...........Advances to the next file in the list.

P...........Backs up to the previous file in the list.

Note that while scrolling with the mouse or selecting a block, you may press either Shift key to scroll by pages rather than lines. Releasing Shift reverts to line scrolling. Pressing Control slows the scroll rate, while pressing Alt gives maximum speed.

• Memory Management

In order to provide the fastest possible display in Text mode, LookIt! requires the file to fit completely in the ST's memory. The memory required by a text file is the size of the file plus an additional four bytes per line. The only hard limit for text files, besides the above, is the number of lines; only the first 65,536 lines of a text file will be displayed. Binary files require 16K bytes each; for files larger than 16K, LookIt! will load additional parts of the file as necessary, and for Binary mode, there is no practical limit to file size.

LookIt! keeps as many files as possible in memory, so that you can flip back and forth between files instantly without reloading them, losing your place, etc. When you attempt to load a file that is too large to fit in the remaining space, LookIt! will free memory occupied by files currently in the File List, beginning with the smallest files, until there is enough room to load the new file. Although memory used by these files is freed, the files remain in the File List, so that they still may be viewed without resorting to the File Selector; they will, of course, have to be reloaded from disk.

• Changing the View Mode

Once a file has been added to the list, you still have a chance to switch view modes. While viewing a file, press F7 to change to the complementary mode. While in the GEM screen, select View File List or press F. This leads you to the File List dialog box, described below. Changing the view mode "invalidates" the file, which means it must be reloaded from disk to be viewed in the new mode.

• Viewing Binary Files

LookIt! displays Binary files as a series of rows of 16 byte lines; the left side of the screen displays the bytes in hexadecimal, while the right side contains the ASCII representation of the bytes. The far left side of each line contains the offset into the file of the first byte in the line; this is the row offset value. This value will always be a multiple of 16. When viewed from top to bottom, the bytes form columns, and the top line of the screen displays the offsets into the line of each column; these are the column offset values. To determine the offset into the file of a given byte, add the column offset value for the byte, which always lies in the range 0 to 15, to the row offset value.

Note: the number 0 is shown as a decimal point in the ASCII display area. Also, the first byte in the file is at offset 0, not 1.

• Editing Binary Files

To edit a binary file, press 'E'. The mouse pointer will appear, and the status line will indicate that you are in Edit Mode. To begin editing, click on a byte in either the hexadecimal or ASCII display. A flashing cursor will appear, and you can begin typing in changes. Use the cursor keys to move around; you cannot scroll the file while in Edit Mode. You may edit either the hex or ASCII fields; to switch back and forth between the two fields, type the Tab key.

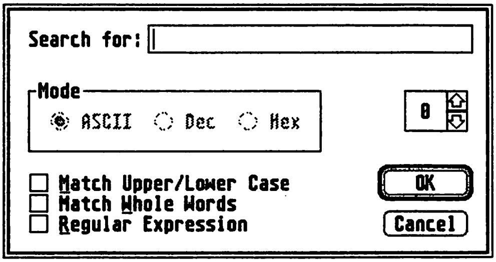
Editing is simple; whatever you type replaces whatever was there before. The only complication is entering zero in the ASCII display. You may accomplish this by pressing the Insert key.

To exit Edit Mode with the intent to update the disk file, press Return. If you've changed some bytes, an alert will appear asking you to confirm your changes. If you decide you don't want to alter the file, choose Cancel. If you press OK, LookIt! will attempt to update the file on disk to reflect the changes you have made. Press Undo to exit Edit Mode without saving any changes.

Updating a file in this manner never affects the file length. Such operations do not alter the time/date stamp of a file; this is the way TOS works. But depending on the File Options, LookIt! can be made to update the time/date stamp to the current system time, and it can be configured to set, clear, or keep the Archive bit setting. See File Options for more.

• Searching

To search for a string of ASCII characters (or decimal/hex bytes, if you are in Binary mode), press the 'S' key while in the View screen. You can specify a search string up to 59 characters or 20 bytes long. If you have exceeded this length, the Search Dialog will not accept the pattern. If you want to search for a set of bytes, be sure to select the proper Mode button in the Search dialog, and separate the bytes with commas. Hex bytes should not be prefaced with "$" or "0x;" instead, make sure Hex mode is set, otherwise, LookIt! won't accept the pattern. The search will begin with the line at the top of the screen and proceed to the end of the file. If a match is found, LookIt! will highlight the line containing the match, and it will place the flashing cursor at the first character/byte of the match. To continue searching, press Return or the Spacebar, or if you are searching a binary file, press 'E' to enter Edit mode directly. To cancel Search Mode, press Esc, Undo, or both Shift keys. The latter also works while LookIt! is searching for a match, as well as when it has found one and is waiting for your acknowledgement.



Search for: |

Mode
● ASCII ○ Dec ○ Hex

8

□ Match Upper/Lower Case
□ Match Whole Words
□ Regular Expression

• Case (In)sensitive Searching

When searching, it is often useful to ignore upper/lower case distinctions; that is, have "search" match "SeArCH." You can control this through the "Match Upper/Lower Case" button in the search dialog. Case-insensitive searching is slightly slower than case-sensitive searching.

• Whole Word Searching

The "Match Whole Word" option in the Search dialog box is available when searching Text files for both regular expressions and literal strings. A "word" is a sequence of characters satisfying the matching criteria such that the matching string is bounded on either side by "non-word characters." A non-word character is any non-alphanumeric character or the "invisible zero-length string" at the beginning and end of every line (that is, the beginning and end of a line always match successfully, if the rest of the search string matches).

• Regular Expressions

LookIt! supports not only exact-match searching but Unix-style regular expressions (in the egrep spirit), as well. Regular expressions constitute a simple yet powerful pattern matching language that far exceeds the standard '*' and '?' wildcard characters TOS and many programs limit you to. If you are unfamiliar with regular expressions, please refer to the section titled "Regular Expressions."

• Saveable Search Strings

LookIt! allows you to define up to ten independent Search Strings for both Text and Binary modes. When you run LookIt!, the active

Search string always defaults to “Search String #0” (SS0). SS0 is intended to be used as a scratch string; its contents are never saved in a configuration file. You can identify and step through the Search Strings by clicking on the arrows at the right side of the Search dialog box. You can also move directly to a Search String by pressing the like-numbered function key; that is, to move to SS2, press F2. Pressing F10 takes you to Search String 0. After defining a Search String, you lock it in by pressing Return to begin the search or using the scroll gadget or function keys to move to another Search String. Once you’ve locked in a Search String, LookIt! remembers it for the remainder of the session, and as long as it is one of Search Strings 1 though 9, LookIt! will save it in configuration files.

Note that LookIt! remembers not only the text of the Search String but also the Search option settings. And while LookIt! never saves the SS0 text, it does save the SS0 Search options in CFG files. Thus, you may always have the scratch Search String default to case-sensitive searching, which as mentioned earlier, is somewhat faster than its counterpart.

• Topic Searches

Many online magazines delimit topics with certain strings; for example, ST-Report uses ” >” at the beginning of lines. To skip to the next topic, you should press Alt plus the function key corresponding to the desired Search string. For example, invoke the Search dialog box and click once on the up arrow gadget, so that ‘1’ appears in the box to its left. Then enter “^ >” in the edit field and select both the regular expression and exact match buttons (this regular expression ensures that matches will be found only at line beginnings). Press either the up or down arrow to “lock in” the Search string, and finally press Undo to exit the dialog. Now, to move to the next topic, press Alt-F1. If LookIt! finds a match, it will place the matching line at the top of the screen, without entering the real Search mode. This is one scenario in which the utility of saveable search strings becomes very apparent.

More About the File List

• Viewing the Current File

The "Current File" is the one you are viewing, or if in the GEM screen, the one you most recently viewed. If there is no current file, this is reflected by the top entry in the View menu. If there is a current file, its name appears as the first entry in the View menu. While in the GEM screen, to view this file, all you need do is select this menu item or press 'C'.

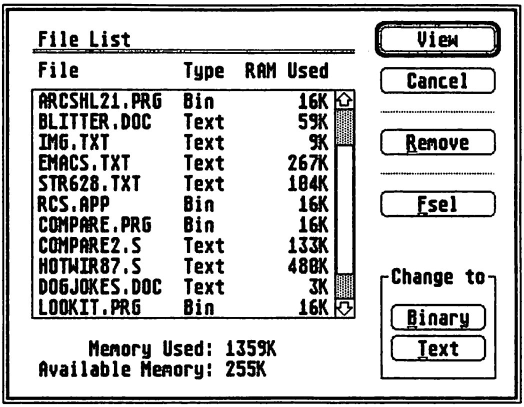
If a non-recoverable error occurs while viewing a file, say, a write error upon updating a binary file, the current file will be invalidated, and you will return to the GEM screen. If this should happen, to view a new file, you will have to explicitly select one from the File List or open one through the File Selector.

• Moving to the Next or Previous File

To leave the current file and view the next or previous one in the file list, press 'N' or 'P'. This works in both the GEM and View screens. If the list contains only one file, a bell will sound. If you are viewing the last file in the list and attempt to view the next file, an alert will appear, giving you the option to cancel, wrap around to the beginning of the list, or choose a file from the File List dialog. A similar alert appears when you're at the beginning of the list and try to view the previous file.

• The File List Dialog

This dialog box is accessed via the View File List menu item or by pressing 'F', and it presents a scrollable list consisting of the filenames currently in the File List. To view a file, simply select it and press OK, or double-click on the file. You may select multiple files and remove them from the list, or you may alter their types by pressing the Binary or Text buttons. The type of each file is indicated by a 'B' or a 'T' following the name in the list. If you wish to load a new file, selecting the Fsel button takes you to the File Selector.



File List

File	Type	RAM Used
ARCSHL21.PRG	Bin	16K
BLITTER.DOC	Text	59K
IMG.TXT	Text	9K
EMACS.TXT	Text	267K
STR628.TXT	Text	104K
RCS.APP	Bin	16K
COMPARE.PRG	Bin	16K
COMPARE2.S	Text	133K
HOTWIR87.S	Text	480K
DOGJOKES.DOC	Text	3K
LOOKIT.PRG	Bin	16K

Memory Used: 1359K
Available Memory: 255K

The File List provides statistics on the memory used by each file; "NA" means that the file isn't loaded into memory. A memory usage summary statistic appears near the bottom of the File List dialog.

This particular dialog box is rather protean; it appears in slightly different forms as the Print Files dialog box and as the Remove Files dialog box. The latter appears when you attempt to load multiple files, but there isn't enough room in the File List (that is, the File List contains 30 filenames, and you attempt to add 10 more -- you would be asked to remove 8 files).

Printing Files

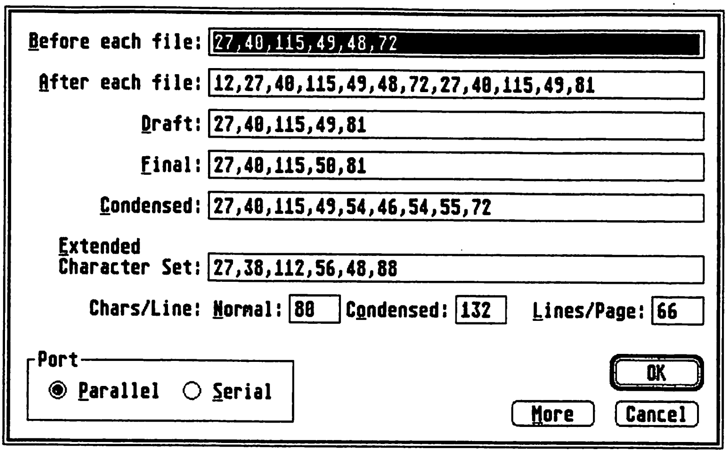
While in the View screen, you may print the current file in its entirety, or you may print a block or just what is currently on screen. In the GEM screen, you may select the files you wish to print from a scrollable list; just choose File Print or press F5 to invoke the Print dialog box (note that printing from the GEM screen acts on entire files). While printing, a status dialog box appears in the lower right corner of the screen. To abort a print job, press both Shift keys.

• Print Options

LookIt! offers a useful set of print options, divided between two dialog boxes (you will have to be in the GEM screen to access these). One of these dialogs is accessible through the menu; choose Options Printer or press Control-P. This dialog box contains the options you will need to alter infrequently, such as the printer control codes, lines per page, etc. The second dialog box contains the "task-specific" options, such as print quality, margins, etc. You may access this dialog box through the "More" button in the Print Options dialog and also from the "Options" button in the Print dialog box. Let's look at each of these dialogs.

• The Main Printer Options

Invoke the Main Print Options dialog (shown on the next screen) by choosing Options Printer from the menu bar. You will find six editable fields, in which you may enter printer control codes. Each field may contain up to 40 decimal values, and you must separate them with commas.



Before each file: 27,40,115,49,48,72

After each file: 12,27,40,115,49,48,72,27,40,115,49,81

Draft: 27,40,115,49,81

Final: 27,40,115,50,81

Condensed: 27,40,115,49,54,46,54,55,72

Extended Character Set: 27,38,112,56,48,88

Chars/Line: Normal: 80 Condensed: 132 Lines/Page: 66

Port: ● Parallel ○ Serial

The six fields are:

- **a.** Before each file,
- **b.** After each file,
- **c.** Draft quality,
- **d.** Final quality,
- **e.** Condensed mode,
- **f.** Extended character set.

Except for the “Extended character set,” these should all be self-explanatory. The “Extended character set” is only relevant when printing binary files, and it is intended to allow your printer to display the ASCII representation of the bytes, similar to what the View screen does. This is required since the printer will interpret certain characters as control codes; we definitely don’t want to send it a string of 80 formfeed characters! Some printers, such as the Hewlett-Packard DeskJet, provide a print mode in which control codes are ignored; instead, actual characters are printed (these characters can be anything; it really depends on the printer). There is a catch; you must specify the number of characters to be printed in this mode, which on the DeskJet, is called "Transparent." Thus, in binary mode, LookIt! always creates lines exactly 80 bytes long, allowing you to make effective use of such printer modes.

Other parameters accessed through the main printer options are:

a. Port, which is Parallel or Serial,

b. Characters/line, in normal mode and condensed mode,

c. Lines/Page.

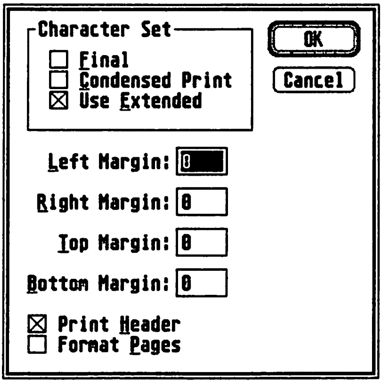
• The Task-Specific Print Options

This dialog box may be accessed via the "More" button in the main printer options or from the "Options" button in the Print dialog box. At the top of this dialog is a set of buttons, which allow you turn the following options on and off:

a. Final Quality,

b. Condensed Mode,

c. the Extended character set (only applicable for binary files).



Character Set

□ Final
□ Condensed Print
☒ Use Extended

Left Margin: 0

Right Margin: 0

Top Margin: 0

Bottom Margin: 0

☒ Print Header
□ Format Pages

Below this is a set of editable fields in which you enter the numeric values for top, bottom, left, and right margins. Top and bottom margins refer to the number of lines from the top or bottom of the page, while left and right margins consist of the number of characters forming the borders. Finally, you may choose to have a header line printed at the top of each page. This header line is always printed one line down from the top of the page, and one blank line is always printed below it. If you are printing headers, and you want a top margin, be aware that the top margin specified includes the header area. That is, a top margin of 5 lines remains 5 lines with use of headers; the header occupies the first 3 line on the page, but it has 2 additional blank lines inserted below it. However, a top margin of 2 lines performs no action with headers, since the top margin is less than the space required by the header itself.

You may receive some ASCII files which are preformatted, containing literal formfeed characters, explicit margins, etc. LookIt! supports printing of such files through the "Format Pages" button. When disabled, Format Pages causes LookIt! to ignore the current margin and header settings, and also the lines/page value (specifically, LookIt! won't send formfeeds to cause page breaks; it will let page breaks fall where they will). Even when "Format Pages" is deselected, LookIt! will send the printer codes as defined under "The Main Printer Options" above.

Note: The left margin has no effect when printing binary files, due to the desirable requirement that each line be 80 characters long. See above for explanation.

• Making Use of the Print Options

Probably the simplest use of printer options is to indent text files, which are often created with no left margin. However, the real power lies in creating a variety of LOOKIT.CFG files for specific purposes. LookIt! is especially good for program listings; as often as programs change, listings just don't stay valuable very long, and at 66 lines per page a lot of paper can be wasted. An example: the HP DeskJet is a very flexible printer, but one can only access a tiny fraction of its power through the front panel of the printer. Normally, it's best to keep it set for draft mode at 66 lines/page. But the DeskJet also provides a 132 lines/page mode, so it's possible to create a DJET132.CFG file containing the proper codes to set it for this mode. Then when you need to print a source listing, you can just load this CFG file and get the job done, reducing the amount of paper and ink required by half. The "After each file" codes return the DeskJet to the default 66 line/page mode, leaving you ready to print out “normal” 66 lines/page files from LookIt! or the Desktop.

Note: Example CFG files are provided on the disk for the HP DeskJet and for Epson compatible printers.

• Printing from the Command Line

If you use a shell which can pass a command line to a program, such as Gulam, msh, or CodeHead Software’s HotWire, LookIt!’s command line features let you take advantage of printer control codes. You have two options for printing files from the command line: -p and -px. The -p option causes LookIt! to print the files specified on the command line, and when done, you will be in the GEM screen. The -px option is just like the first, except that LookIt! exits immediately after printing the files, returning you to the shell.

One other useful command line feature is the -cCFGfile switch. The -c switch allows you to load a given configuration file instead of the default. To use -c, append the full path and CFG file name to it, with no intervening spaces. As an example of this, under a CLI shell you can define an alias “prs” to print source files. “prs” really stands for “lookit -px -cC:\TOOLBOX\DJET132.CFG”; this allows you to use prs much like a standard CLI “pr” or “type” command, except that you can take advantage of LookIt! configuration files and printer control codes.

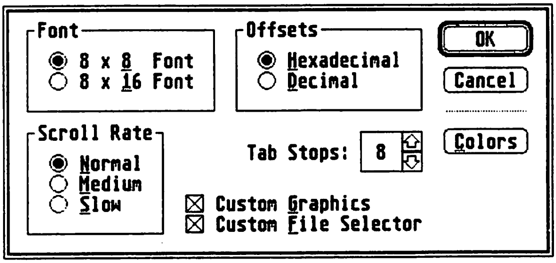
If you use CodeHead’s HotWire shell, you can set up entries in the menu for LookIt! with predefined command lines to do these same functions, and use LookIt! to print files with a single keypress or mouse click. See the HotWire documentation for more details.

Screen/File Options

LookIt! allows many configuration possibilities; the printer options have been described under "Printing Files." Here, we'll discuss the Screen and File Options.

• Screen Options

The Screen Options dialog box is accessed by choosing Options Screen from the menu or by pressing Control-S while at LookIt!'s GEM screen. It allows you to set:



Font
● 8 x 8 Font
○ 8 x 16 Font

Offsets
● Hexadecimal
○ Decimal

Scroll Rate
● Normal
○ Medium
○ Slow

Tab Stops: 8

☒ Custom Graphics
☒ Custom File Selector

a. in high resolution modes, the character set, which is either the 8x8 or 8x16 font (F10 in the View Screen accomplishes this, too). In medium resolution, only the 8x8 font is available.

b. the numeric base, decimal or hexadecimal, for the row and column offsets for binary file viewing (F8 in the View screen).

c. the spaces per tab stop (Keyboard equivalents: + and -, up and down arrows).

d. use of Custom Graphics. LookIt! uses optimized assembly routines in the View screen to provide very fast text display, paging, and scrolling. While LookIt! understands the current ST video modes and should be compatible with the forthcoming TT, it's possible that it will be incompatible with some existing modifications. If so, you can disable Custom Graphics, and LookIt! will use the (slow) built-in GEM graphics routines, although programs like Turbo ST will speed up the display (but probably not as much on average as the LookIt! routines do). This button will be disabled if LookIt! determines that it doesn't understand the video mode when it is first launched. For the technically-minded, this will occur if the number of bit planes is something other than 1, 2, or 4, or if the bytes per scan line is not one of 80, 160, or 320.

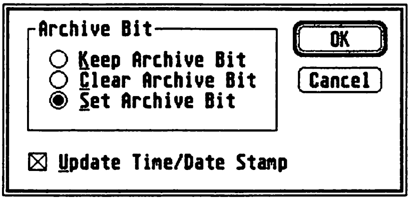
e. foreground and background color used in the View screen. The foreground color is used to render the characters. Note that you are not restricted to the colors currently in the palette. For example, consider the 4 color medium resolution mode. My normal colors are white, black, red, and cyan. However, I like to view white text with a blue background. To accomplish this, I use the Control Panel desk accessory to set one of the colors to blue and choose the proper colors from the Colors dialog box. I save a CFG file, and then I return to the Control Panel and reset the colors to their original values. But when I enter the View screen, LookIt! displays a blue background, even though blue is no longer in the current palette. And it will keep doing so, as long as I don't invoke the Colors dialog box and press the OK button; the Colors dialog box only displays the colors currently in the palette (otherwise, it would have to change the palette, possibly changing the GEM screen colors in the process). LookIt! maintains separate color settings for monochrome and 'multi-color modes.

f. File Selector: by default, LookIt! uses the InterLib file selector, which is unique in that it allows you to select multiple files (in a way that's meaningful to the program which invoked it!) and offers superior Path/File Name handling. But if you are one of those passionate people who would rather fight than switch file selectors (and you know who you are), you can continue to use your favorite by deselecting the "Use Custom File Selector" button.

g. Scroll rate: Since LookIt's scrolling and paging is very fast and is not limited by the keyboard repeat rate, a means for controlling the scroll rate is required (imagine paging 4-8 times faster on a TT!). "Normal" equates to maximum speed; that is, the scroll rate is limited only by the processor speed. "Medium" provides a maximum of 40 auto-repeat scroll/page operations per second. "Slow" gives a maximum of 20 auto-repeat operations per second. Note that for a standard ST, "Normal" and "Medium" will be very nearly the same.

• File Options

This dialog box allows you to control LookIt!'s actions upon updating a binary file that you have edited or upon saving or appending a block. As mentioned earlier, if one writes to an existing file, the ST operating system doesn't update the file's time/date stamp. You may choose whether or not LookIt! should update the time/date stamp to the current system time. You may also elect to set or clear the Archive bit, or you may have LookIt! keep it as it is. In TOS versions prior to 1.4, altering a file cleared the Archive bit, which is exactly the opposite to MS-DOS and TOS versions 1.4 and above, which set the Archive bit when a file is changed. This bit is intended for use by backup programs, so they can perform incremental backups; i.e., backup only those files that have changed since the last backup. If you are using TOS 1.4 or higher, just choose "Set Archive Bit;" for earlier versions, you'll probably want to choose "Clear Archive Bit."


Archive Bit
○ Keep Archive Bit
○ Clear Archive Bit
● Set Archive Bit

☒ Update Time/Date Stamp

• Configuration Files

LookIt! allows you to save the screen, file, search, and printer options in configuration files. These files have the extension CFG, and LookIt! attempts to load the default CFG file when it is first launched. This file is normally named LOOKIT.CFG, and LookIt! looks in the default system directory for it. This is often adequate, but if you have installed LookIt! as a Desktop application and launched it by double-clicking on the installed file type, chances are, it will be unable to find the CFG file, as the Desktop program sets the system path to that of the file clicked upon, and not LookIt!'s path. This is similar to what happens when applications use separate resource files; they often can not find them when installed as an application (this is one example of Atari and DRI throwing out the good parts of MS-DOS, at least for the Desktop program -- why don't we have system-wide environmental variables and a built-in command line interpreter??). To circumvent this problem, if LookIt! can not find the LOOKIT.CFG file, it will ask you to locate it with the file selector. You also will have the chance to save the path and name of this file within the LookIt! program file itself, so that LookIt! will always be able to find the default CFG file. Thus, you should be able to maintain a default LOOKIT.CFG file that LookIt! will be able to find, no matter how it was launched. And remember, the -c command line switch allows you to override the default CFG file for special purposes.

The Command Line

LookIt! can be run very effectively from a shell which supports command lines, such as CodeHead's HotWire. The following options are available through LookIt!'s command line interface:

```
Usage: lookit [-b] [-t] [-p] [-px] [-h] [file1 file2 ... file32]
-b: binary mode
-t: text mode
-p: print files
-px: print files and exit
-h: this screen
-cCFGfile: loads specified CFG file instead of default
The mode switches may appear multiple times,
as in 'lookit -b a.prg a.rsc -t a.c -b a.wks'
File specifications may include wildcards.
```

The mode switches, -b and -t, remain in effect until their counterpart (if present) is encountered, and they force the following files to be opened in the requested mode. In the absence of a mode switch, LookIt! opens files that match its built-in extensions in the appropriate mode, and it opens files that do not match in Text mode.

Note that the -c option must be followed immediately by the full path designation of a CFG file. That is, -cC:\TOOLBOX\DJET132.CFG causes LookIt! to load DJET132.CFG from the C:\TOOLBOX directory.

About InterLib

LookIt

• A Better GEM Interface

LookIt! is the first program to use my new GEM interface library, which I will refer to as InterLib. I've always been of the opinion that GEM is only a useable graphical interface; it's not exceptionally attractive and in some cases not very functional. In particular, the built-in dialog box text editing capability is pretty abysmal. As another example, there's no way to visually tell the difference between a radio button and a simple push (on/off) button, because all buttons are rectangular, and the borders of these two button types share the same thickness. GEM also doesn't make very effective use of icons (for example, window and list arrow controls are characters from the system font), probably because it was designed for slow 8088 processors with limited memory and graphics resolutions with non-uniform aspect ratios (such as CGA [like ST medium res] and ST high res). These things contribute to a rather stark-looking display. And GEM pretty much forces you to use the mouse, even when the keyboard is obviously more convenient.

From a programmer's standpoint, things are worse. For example, there is no high level support for concepts like scrollable lists or free scroll bars, no support for multi-line edit fields with word wrap, dynamic reformatting, block selection, etc:, and there are many other short-comings. InterLib provides these features and much more.

• The Menu Bar

One of the first things you will notice is that the menu bar titles have an underlined character in monochrome or a red character in color.

These characters are the hotkeys for the menu titles; yes, you can operate an InterLib menu solely from the keyboard! To drop down a menu, press the Alt key plus the appropriate hotkey. Use the up and down arrow keys to advance the highlight bar through the menu items; press Return or Enter when it is over the desired menu item. You will notice that the menu items also have hotkeys, much like the menu titles. To choose one directly, press the key by itself (Alt isn't required). Cancel a menu by pressing Esc or Undo. To switch to adjacent menus, use the left and right arrow keys.

• The Cancel Button

Just about every dialog box has a default button, which you can select by pressing Return, and where there's a default button, there's usually a Cancel button not too far away. Unfortunately, GEM provides no way to select a Cancel button from the keyboard.

InterLib supports the Cancel button; you select it by pressing Esc or Undo. However, GEM defines Esc as clearing an edit field, so the keyboard equivalent for the Cancel button in a dialog containing edit fields is Control-Esc, Alt-Esc, or Undo (which work in any dialog).

• Keyboard Equivalents

InterLib supports keyboard equivalents for all buttons and edit fields. The hotkey will be underlined in monochrome and displayed in color 2 (usually red) in multi-color video modes. If the dialog contains any edit fields, the hotkey consists of Alt plus the indicated letter. If no edit fields are present, the letter by itself is all that's required (although the Alt combination still works).

Note that the Default and Cancel buttons use Return and Esc/Undo as their hotkeys, so you won't be seeing "hot letters" in these buttons. The Default button is identified by its heavy border, while the Cancel button is identified (usually) by the word "Cancel."

• Radio Buttons

These buttons come in sets out of which none or only one can be selected. They consist of a circle followed by some text. The selected state is indicated by a solid circle drawn within the circle, while the off state is signified by a hollow circle.

• Push Buttons

These buttons have two states, on and off, and they consist of a small box followed by some text. The on state is indicated by a cross drawn in the box, while the off state is signified by a hollow box.

Note: while you certainly can click a push or radio button's control gadget, you can also click on the associated text to select the button.

• Edit Fields

InterLib edit fields allow you to position the insertion point with the mouse or the keyboard protocol. You may also select text by dragging the mouse. When a block is selected, the characters will be highlighted, and whatever you type next replaces the block. You may cancel the block by clicking elsewhere in the field, or you may press the left arrow to cancel the block and place the cursor at the beginning of the block. Pressing the right arrow cancels the block and places the cursor at the end of the block.

The keyboard protocol is as follows:

Left arrow: cursor left, Right arrow: cursor right, Up arrow: cursor up (multi-line field), Down arrow: cursor down (multi-line field), Shift+left/right arrow: to end of line, Shift+up/down arrow: page down (multi-line field), Control+left arrow: to previous word, Control+right arrow: to next word, Home: to first character in field, Shift-Home: to last character in field, Delete: deletes character or block, Backspace: backspaces, or when a block is selected, cancels the block and backspaces over the last character involved in the block Esc: clears the field, Tab: to next edit field, Shift-Tab (backtab): to previous edit field.

Note that unless the insertion point is in a multi-line field, the up and down arrows function just like they do in standard GEM dialog boxes; they activate the next or previous edit field. Try Shift-Tab to back up to previous edit fields, and notice that as you advance beyond the first or last field in the dialog box, the active edit field wraps to the other end of the dialog box.

The mouse protocol is as follows:

Single-click: position cursor, or activate new edit field, Double-click: select word, Click and drag: select block, Double-click and drag: select block by words, Shift-click or right-left-click: extend block, Shift-double click or right-double left-click: extend block by words, Ctrl-Alt-click: select entire field.

Single-line fields may be pannable in the horizontal direction. If you are selecting a block in such a field, dragging the mouse beyond the border of the edit field causes it to pan left to right or right to left. If you find this panning too fast, you can slow it down by pressing the Control key at any time. Releasing Control resumes full-speed panning.

Multi-line fields never support panning; however, they may support scrolling up and down. Selection in such fields is similar to the above, except that dragging the mouse above or below the field results in scroll-selecting, and the Control key again serves to slow this down.

Note: once you have started to extend a selection, you may release the shift key or right mouse button. “Right-left click” means press the right mouse button, hold it down, and press the left mouse button.

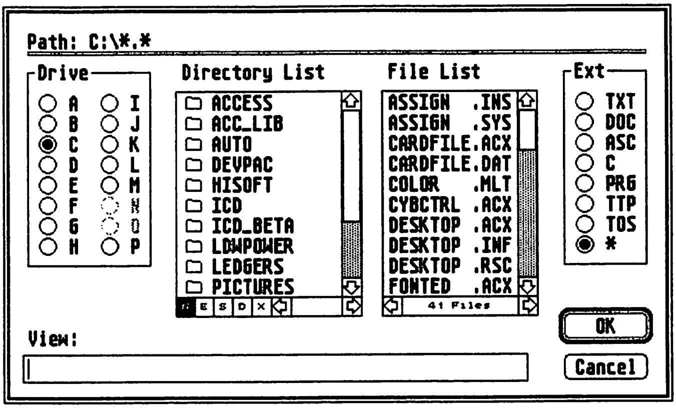
• Scrollable Lists

These work in the expected fashion, except that some lists support multiple selection. To select several contiguous items, drag the mouse over them. If you drag above or below the list, the list will scroll automatically, selecting items as you go. As with edit-fields, you may slow this scrolling by pressing Control at any time. You may also extend a selection by Shift-clicking or right/left clicking. If the item isn't selected, it will become selected, and vice versa. Clicking on any item (as opposed to extending a selection) deselects all currently selected items. Finally, Ctrl-Alt-clicking selects every item in the list.

Note: once you have started to extend a selection, you may release the shift key or right mouse button. "Right-left click" means press the right mouse button, hold it down, and press the left mouse button.

The File Selector

If you've read the preceding section, operation of the File Selector should be straightforward. However, some features bear mentioning. And remember, the InterLib File Selector often supports selection of multiple files!



Path: C:\*.*

Drive

○ A ○ I
○ B ○ J
● C ○ K
○ D ○ L
○ E ○ M
○ F ○ N
○ G ○ O
○ H ○ P

Directory List

```
□ ACCESS
□ ACC_LIB
□ AUTO
□ DEVPAC
□ HISOFT
□ ICD
□ ICD_BETA
□ LOWPOWER
□ LEDGERS
□ PICTURES
```

File List

```
ASSIGN  .INS
ASSIGN  .SYS
CARDFILE.ACX
CARDFILE.DAT
COLOR   .MLT
CYBCTRL .ACX
DESKTOP .ACX
DESKTOP .INF
DESKTOP .RSC
FONTED  .ACX
```

41 Files

Ext

○ TXT
○ DOC
○ ASC
○ C
○ PRG
○ TTP
○ TOS
● *

View:

• Sorting

Under the directory list are five buttons, N, E, S, D, and X. These control directory sorting, and respectively, they stand for: by Name, by Extension, by Size, by Date, and No Sort (i.e. display files in the order they appear in the directory). For sorting by size or date, conflicts between files of equal size/date are resolved by sorting by extension then by name. Regardless of the sorting method, all directories are always displayed. Sorting by size doesn't affect directories, since directories always have zero-length size fields.

• File Masks

The File Selector improves on wildcard usage, compared to the TOS facility. For example, *?M*.* is the same as *.* under TOS and MS-DOS; once a star is seen, whatever follows it is unfortunately ignored. The InterLib File Selector's algorithm interprets this correctly; it means, "match those files containing an M in the name portion preceded by at least one character." Use "*" by itself to match all files without extensions, and use "*.*" to match all files.

• Changing Drives

The File Selector will remember the last path you accessed on each drive. Of course, you may click on the drive radio buttons to switch drives, but you can also type the letter of the drive and a colon or colon plus backslash. For example, A:\ takes you to the last path accessed on drive A. On the other hand, A: takes you to the root of drive A. No matter what, the current file mask is maintained upon switching drives.

• Media Change/Rereading Directories

To force a media change, or to make the File Selector reread a disk, click on the drive button that is already selected. It must already be selected to force a media change; otherwise, the File Selector just rereads the disk (which may be inappropriately cached in memory, TOS [especially versions earlier than 1.4] has problems with media change)! Thus, if the File Selector is currently displaying A:\*.*, and you change the disk in drive A, click on the A button to update the file list display.

• Entering Paths and File Names

In the InterLib file selector, the current path and file mask is displayed at the top of the dialog box. In order to "back out" to a parent directory, click on the directory name in the Path line. If you click on the name of the current directory, the entire path field will be copied to the edit field, so you can still manually edit a path. If you click on the file mask, only the file mask is copied to the edit field.

When you have entered a path into the edit field, pressing Return or clicking OK causes the file selector to display the contents of that path. If you have entered a file mask alone, the path remains the same, and the new file mask is applied to the file listing.

Let's look at an example. Suppose the current path is D:\SRC\*.*, and you want to change to D:\SRC\VIEW\*.*. You could click on the VIEW entry in the directory listing, or you could type in VIEW and press Return. The file mask would remain *.*, but you are now in D:\SRC\VIEW. Now suppose you wanted to change to D:\SRC\LIBGEM. The GEM item selector would require you to edit the path line, which is terribly cumbersome, or an alternative would be to first back out to D:\SRC and then click on LIBGEM in the list. But if your D:\SRC contains several hundred files, you might have to wait a couple of seconds for GEM to sort the directory. Under InterLib's file selector, all you need do in such a situation is type ..\LIBGEM and press Return. This is what we command line aficionados do all the time, and once you get used to it, it's very convenient and fast. The .. stands for the parent of the directory. By entering ..\LIBGEM, I created a path D:\SRC\VIEW\..\LIBGEM. Reading left to right, this says go to D:, then to D:\SRC, then D:\SRC\VIEW, then D:\SRC\VIEW\.., then the algorithm, scratching its head, says to itself, "well, D:\SRC\VIEW\.. is the same as D:\SRC, so make it D:\SRC", and then finally, we arrive at D:\SRC\LIBGEM. Still another way to do this would be to click on VIEW in the path line, which copies D:\SRC\VIEW\*.* to the edit field, and then drag the mouse through VIEW, type LIBGEM, and then press Return. Either way, you bypass displaying the parent directory, and with some practice, I think you will find either of these methods superior to what standard GEM has to offer (although you certainly can continue to navigate the directories in GEM item selector fashion).

By now it's probably apparent that the File Selector always checks to see if the edit field actually contains a path or directory name when you press Return. What may not be apparent is that the File Selector doesn't allow you to change to paths that do not exist, something the GEM item selector merrily permits.

Finally, you can enter a path plus a filename directly into the edit field and return that to the calling program. The advantage here is that the File Selector won't first change to the path, sort the contents, display the new file listing, and then return. Rather, it will just validate the path and return right away, which can be a bit faster.

• Changing File Extension Masks

Depending on the program, the File Selector may contain up to 8 wildcard extensions. These will be found as a set of radio buttons located on the right side of the dialog box. Clicking on one of these only changes the extension of the file mask, not the name portion. If you enter a new file mask in the edit field, and the extension matches one of the provided defaults, the appropriate button will become selected.

Troubleshooting

As far as we know, LookIt! is free of bugs. If you do encounter any quirks, please let us know. What follows are problems you may find due to bugs in TOS or other programs.

• TOS 1.0

Note: TOS 1.0 has a bug in that it sometimes fails to recognize when the mouse button has entered the "up" state. This affects LookIt! in two ways: a. LookIt! may appear to "stall" during certain operations (example: open a new file, and the View screen doesn't appear right away), and b. some auto-repeat functions may keep on repeating after you release the mouse button (example: scrolling a list). The solution is simple; all you need do is nudge the mouse, and TOS will figure out the mouse button state. Fortunately, this is a relatively infrequent problem, but it sure is annoying when it occurs!

• Turbo ST

Note: I've noticed that a very few programs, when used in conjunction with Turbo ST 1.6, conflict with InterLib, causing rather bizarre displays. For example, button text will appear above rather than within the button, and fill patterns may appear solid black. This is not a problem with InterLib or LookIt!, because they do nothing illegal, and the solution is to disable Turbo ST. Some programs that elicit this behavior include CALC.ACC from the defunct Compute! ST magazine, InterLink 1.70 (when LookIt! is run from within it), and some written with (early?) versions of Personal Pascal. Compute! provided the source for CALC.ACC, and examination revealed the author opened his virtual workstation using work_in[0] = 0, rather than the recommended work_in[0] = Getrez()+2. Changing the first statement to the latter corrected the problem, so apparently, opening a virtual workstation for the "default screen" confuses Turbo ST at some level, at least when another application, like LookIt!, comes along and opens two virtual screen workstations (InterLib needs its own, and LookIt! needs one, too).

• Gulam and msh

Note: if you are running LookIt! from a command line shell such as Gulam or msh, and you force a media change on the shell's current drive, strangely, the shell "loses" its idea of the current path. That is, if I was in D:\SRC\LIBGEM in the shell, and I forced a media change on drive D while in LookIt!, when I returned to the shell, its current path would be D: (even though it might still display D:\SRC\LIBGEM at the prompt). This problem isn't limited to LookIt!; several other programs and desk accessories that can force a media change all cause this behavior. In my opinion, this is a problem with the shells, because programs they launch should not be able to, in effect, alter private shell variables (except possibly through a well-defined protocol).

Regular Expressions

LOOKIT

• What are Regular Expressions?

Regular expressions provide a powerful means of specifying patterns for use in searching applications. In fact, if you've used filename wildcards or the search feature in many word processors, then you're already familiar with a limited form of regular expression. When you designate a filename wildcard, you are actually creating a language, and the filenames a program displays for that wildcard are sentences that belong to or match the language. Full regular expressions enrich the pattern-matching language, making possible searches like: "find all strings beginning with 'form_' followed by 'do', 'alert', or 'button', then '(', then one or more of any character, and finally terminated by ')'." In regular expression notation, the pattern which would denote this language would be: "form_(do|alert|button)\(.+\)".

• Formulating

Regular expressions are subtly different from the familiar filename and word processor wildcard facility. For one, the symbols "*" and "?" have somewhat different meanings. Also, valid regular expressions are not free-form in nature. Rather, they must conform to a well-defined grammar, which fortunately is not difficult to learn.

• Simple Patterns

The simplest patterns are those consisting of but a single character; the regular expression A matches the letter "A" anywhere within a line of text. Naturally, you concatenate these simple expressions to form words; the regular expression EXPR matches the string “EXPR” wherever it occurs. Note that these are exact matches; that is, EXPR would not match “EXXR”.

• Metacharacters (Operators)

There are thirteen special characters in this implementation of regular expressions. They belong to the set “*+?.|()[]-$^\”. In order to match any of these characters exactly, you must “escape it,” that is, preface it with a backslash. Thus, to match “WHY?” exactly, the regular expression would be WHY\? -- the regular expression to exactly match “C:\AUTO” would be C:\\AUTO. These special symbols are the *operators* of regular expression grammar, much like “+-*/” are the operators of simple arithmetic.

• The Star Operator, *

This stands for “zero or many.” The star operator follows the regular expression (or part of a regular expression) that you wish to match zero or more times. For example, A* would match the strings “” (empty string), “A”, “AA”, and so on. EXPR* would match “EXP”, “EXPR”, “EXPRR”, and so on. Note that in the last example, the star operator applied to the character immediately preceding it, which is “R”. This will always be the case. To have it match several characters, use parentheses to force grouping, as in EX(PR)*; this would match “EX”, “EXPR”, “EXPRPR”, “EXPRPRPR”, etc.

• The Plus Operator, +

This stands for “one or many.” You use the plus operator in the same fashion as the star operator.

• The Question Mark Operator, ?

This stands for “zero or one.” You use this operator just like the star and plus operators.

• The Dot Operator (period), .

This means “match any single character” and is the counterpart to the filename wildcard operator ”?”. To emulate the filename wildcard star operator, write the regular expression ”.*”. This matches zero or more occurrences of any character. For example, FORM.* matches all strings beginning with “FORM”, regardless of what follows, because the star operator was applied to the dot operator.

• The Or Operator (vertical bar), |

The or operator goes between two regular expressions, as in EX1|EX2, and it means match either the left-side or right-side regular expression. You may use more than one or operator, as with EX1|EX2|EX3|EX4, to match any of a set of regular expressions.

• Parentheses

Use parentheses to force grouping, as in arithmetic expressions. For example, (EX1|EX2)*EX3, matches zero or more occurrences of “EX1” or “EX2” followed by a single “EX3”. This regular expression would match “EX1EX3”, “EX2EX1EX1EX2EX3”, “EX3”, and so forth.

• The Bracket Operator, []

This operator allows for character sets and ranges to be specified. For example, at some point in the regular expression, you might like to match any character out of "A", "B", and "C". Writing [ABC] (or [CBA], the order doesn't matter) accomplishes this. If the first character within the brackets is a circumflex, "^", then the match will be for any character not listed within the brackets.

Note: Except for ^ and -, all the metacharacters lose their special meanings when inside the brackets (^ is special only when it's the first character within the brackets, and - is special only if it's a part of a character range). Thus, [*+.\^-] would match any of the six characters within the brackets, even though they are metacharacters. Note that -is the last character in the character set; this ensures that it will not be interpreted as the Dash operator, described below.

• The Dash Operator, -

This operator finds its purpose when used with the bracket operator; it allows specification of character ranges. For example, to match a single alphanumeric character, write [A-Za-z0-9]. This regular expression says "match any character from 'A' to 'Z', 'a' to 'z', or '0' to '9', and this encompasses all the alphanumeric characters. When you designate a character range, the first symbol must have a numeric ASCII value less than or equal to the last; that is, the ranges in the above example are all valid, but [z-a], [9-a], and [a-A] are not.

• The Circumflex Operator, ^

If placed at the beginning of the regular expression, this operator causes matches to be found only at the beginning of lines. For example, ^EXPR would match a line “EXPR something” but not “something EXPR” or ” EXPR something”.

As described under the bracket operator, the circumflex, when the first character inside the brackets, negates the sense of the character set/ range matching.

• The Dollar Operator, $

When placed at the end of the regular expression, this operator causes matches to be found at the end of lines. For example, EXPR$ would match a line “something EXPR” but not “EXPR something” or “something EXPR ”.

To match entire lines, place the circumflex operator at the beginning of the regular expression and the dollar operator at the end.

• The Backslash Operator, \

Finally, this operator allows you to “escape” any of the special characters (including itself), so that the character will be taken literally. For example, the regular expression \(matches a single open parentheses character.

• Putting It All Together

Example 1: Suppose you want to find all occurrences of “port” in a text file, regardless of whether it is within a word or occurs by itself. Under most word processors, you would do a search using “port”. To accomplish this with a regular expression, you would use the same thing. But suppose you were really interested in finding only those occurrences at the end of a word, such as in city names. With standard wildcard searching, you might be out of luck. But with regular expressions, “[A-Z]+[a-z]*port” would do the trick. This says "match at least one uppercase alphabetic character (note the plus operator), followed by zero or more lowercase alphabetic characters (note the star operator), and terminated by port". Thus, this regular expression matches "Newport" and "Logansport" but not "Port Vincent" or "port".

Example 2: Suppose you're a programmer and want to find all occurrences of the functions "form_alert", "form_do", and "form_error". Using the regular expression or operator, the expression would be "form_(alert|do|error)". This says "match all occurrences of 'form_' followed by any of 'alert', 'do', and 'error'." With standard wildcard based searching, you would have to do three separate searches or use the wildcard "form_*" and possibly accept matches you weren't interested in.

• Regular Expression Error Messages

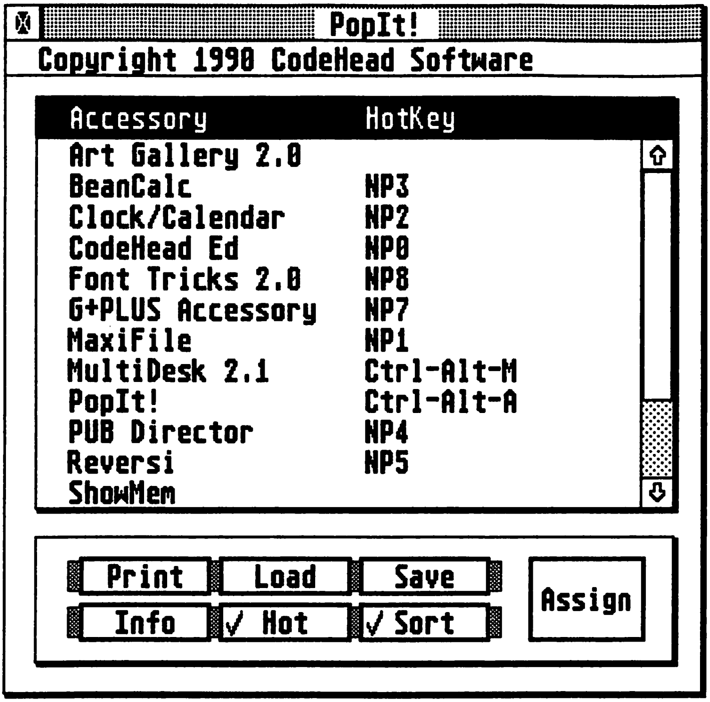
As mentioned earlier, regular expression notation constitutes a genuine language with a well-defined syntax; when LookIt! encounters a malformed regular expression, it displays an alert box describing the nature of the error. Let's look at a few examples.

Example 1: RE = *test LookIt! Response: "egrep: misplaced *, +, or ? Reason: The star operator can not begin a regular expression, as it is a postfix operator. In other words, it must follow the (sub)expression it is to apply to.

Example 2: RE = d[o-i]g LookIt! Response: "egrep: error in char class" Reason: An invalid character class was specified. o-i is an improper character range, because the ASCII value for 'o' is greater than the ASCII value for 'i'. Use i-o instead.

This page unintentionally left blank.

Using PopIt!



PopIt!

Copyright 1990 CodeHead Software

Accessory	HotKey
Art Gallery 2.0	
BeanCalc	NP3
Clock/Calendar	NP2
CodeHead Ed	NP0
Font Tricks 2.0	NP8
G+PLUS Accessory	NP7
MaxiFile	NP1
MultiDesk 2.1	Ctrl-Alt-M
PopIt!	Ctrl-Alt-A
PUB Director	NP4
Reversi	NP5
ShowMem	

• Installing PopIt!

Copy POPIT.ACC to the root directory of your boot disk; for floppy-based systems this would be A:\, and for auto-booting hard drives this would typically be C:\. Now reset your computer, and observe the screen during the boot process. You should see a message to the effect that, “RegAcc 1.0 is installed.” Now, once the Desktop appears, drop down the Desk menu and click on the PopIt! item. If all has gone well, you should be presented with a window containing the names of all the desk accessories appearing in the GEM Desk menu in a scrollable list. And if you are lucky enough to have Codehead Software's MultiDesk 2.0 or later, all of the MultiDesk accessories will be available as well!

• Assigning Hotkeys

When you run PopIt! for the first time, no hotkeys will be assigned. To assign a hotkey to a desk accessory, click once on the desk accessory name in the PopIt! list, and then click on the "Assign button" (a short-cut is to Control-click on the name). Observe the "info line" in the window; it will state "Press a HotKey combination." The legal key combinations include: the alphabetic keys, numeric keys (including the keypad), the punctuation keys, the function keys, the space key, and any member of these sets in conjunction with some combination of the Alternate, Control, Left-shift, and Right-shift keys (Alt, Ctrl, LS, and RS, respectively; LS and RS are distinct keys). Since these hotkeys will be active no matter what program you are running, you should strive to choose key combinations unlikely to be used by any program, because once RegAcc sees a keypress PopIt! is waiting for, that keypress is stolen, and a foreground program will not register it. Good choices include combinations like Ctrl-Alt-Key, LS-Ctrl-Key, etc.

Once you have designated a hotkey, you may later want to change it. Select the accessory as described above and press the new combination. Press Undo to clear the hotkey definition. Press Return, Enter, or Esc to cancel the assignment and keep the old definition. To clear all hotkey definitions, press Ctrl-Undo.

• Saving HotKey Definitions

Once you have set up the hotkeys to your liking, you will want to save a configuration file, POPIT.CFG, that PopIt! will load each time you boot your computer. To accomplish this, click the "Save" button. The Item Selector will appear, allowing you to save a CFG file under any name, which can be useful as PopIt! allows loading of arbitrary CFG files. However, remember that the default CFG file, which PopIt! loads at boot-time, is always called POPIT.CFG, and you should place it in the root directory of your boot drive. Providing you've saved POPIT.CFG, the next time you boot your ST, PopIt! will load this file and automatically assign the hotkeys.

You have two choices when you save a CFG file: "Save Active" and "Merge." Each of these saves only those definitions currently active, that is, only those accessories that currently appear in the PopIt! list and have defined hotkeys. "Save Active" differs from "Merge" in that other definitions in the CFG file are lost; the existing CFG file is erased and a new one is created, which contains only the currently active definitions. To retain those definitions in an existing CFG file, choose "Merge."

PopIt! supports hotkey definitions for as many as 50 desk accessories. If you boot your computer with different sets for different purposes, you can use the Merge feature to maintain a CFG file with definitions for all your accessories. The active hotkeys, of course, are determined by what desk accessories are present in the system at a given time.

• Loading Sets of HotKey Definitions

Although a PopIt! CFG file can contain up to 50 hotkey definitions, PopIt! is flexible enough to load alternate CFG files. To load a new set of hotkeys, click the "Load" button and use the Item Selector to choose the CFG file. If you have changed the current set of hotkeys, PopIt! will give you the chance to save a CFG file before proceeding to load the new set, so you never have to worry about losing any changes you have made.

• Duplicate Hotkeys

PopIt! requires all accessories to have unique hotkeys; this requirement extends to all entries in the CFG file. Thus, if you try to assign a hotkey in use by an unloaded member of the CFG file to one of the accessories that is resident in memory, PopIt! will inform you of this. PopIt! will allow you to override the prior definition, meaning that the hotkey for the conflicting accessory will become unassigned.

• Printing the HotKey List

To print a listing of the desk accessories and their hotkeys, click the "Print" button. An alert box will appear, asking you whether to print a "Full" or "Active" listing. A "Full" listing causes all definitions to be printed, including those for absent desk accessories that still appear in the current CFG file. An "Active" listing contains entries for just those accessories currently present that have defined hotkeys.

• Temporarily Disabling PopIt!

PopIt! responds to hotkeys only when a checkmark appears in the "Hot" button in its window. You may click on this button to toggle hotkey monitoring on and off. You may also press Control, Alt, Left Shift, Right Shift, and Tab (Ctrl-Alt-LS-RS-Tab) simultaneously to accomplish this; this key combination is always active, even when the PopIt! window isn't displayed. The latter might be useful while running a program that doesn't allow access to the accessories and uses a key you have defined as a hotkey; otherwise, you would be unable to disable PopIt! without exiting the program.

Note: you may still assign hotkeys when the "Hot" button is not checked. The status of the Hot button is saved in the CFG file.

• Sorting the Accessory List

To alphabetically order the PopIt! accessory list, click the "Sort" button. In the unsorted state, the GEM Desk menu accessories appear first, and they will be in the same order as displayed in the Desk menu. The MultiDesk accessories follow the standard accessories, and they will appear in the same order as displayed by MultiDesk in the unsorted mode.

Note: the Sort status is saved in the CFG file.

• PopIt! as a Tearaway Menu

PopIt! not only provides hotkey access to your desk accessories, it also serves as a desk accessory "tearaway menu." To call a desk accessory directly, double-click on its name in the PopIt! window. The desk accessory will appear, and if it uses a window, you should be able to bring PopIt! to the top and call up yet another desk accessory. This feature works even when hotkey monitoring has been disabled.

Command Summary

While the PopIt! desk accessory is open:

L	Load CFG file
S	Save CFG file
P	Print CFG file
I	Information
O	Sort Desk Accessories
H	Toggle HotKey monitoring (only when PopIt's window is open)
Ctrl-Alt-LS-RS-Tab	Toggle HotKey monitoring (works at all times)
A	Assign HotKey to a selected desk accessory (or Ctrl-click the accessory name)
Ctrl-Undo	Clear all hotkey definitions
Esc	Close the PopIt! window

After entering the hotkey assignment mode:

Return, Enter, Esc	Keep current definition
Undo	Clear the definition

Troubleshooting

Problem: I'm using MultiDesk 2.0 or later, but PopIt! isn't handling MultiDesk accessories.

Reason: PopIt! is loaded as a MultiDesk accessory.

Solution: PopIt! should always be loaded as a normal GEM desk accessory, so place it in the same directory as MultiDesk and any other desk accessories you wish to appear under the "Desk" menu title.

Problem: When I move the PopIt! window, and release the mouse button while the pointer is over an PopIt! button, or I click on a dialog box button that lies above an PopIt! button, the PopIt! button becomes selected.

Reason: TOS is buggy. PopIt! uses evnt_multi() with a timer event; masking out the timer event results in the expected behavior.

Solution: Wellll, since we can't mask out the timer event and have a working PopIt!, Atari needs to fix the ROMs!

Problem: Since installing PopIt!, my computer seems to "lock up" or crash more frequently.

Reason: TOS versions prior to 1.4 had a bug which caused desk accessories that use timer events to "sleep forever." I don't believe I ever encountered this bug while using TOS 1.0, and it may be a problem that appears only when the conditions are just right.

Solution: Upgrade to TOS 1.4. You won't be sorry; it really is a major upgrade, especially if you have a hard drive.

This software and accompanying written materials are provided "as is" without warranty of any kind. The entire risk as to the results and performance of this software is assumed by you. CodeHead Software hereby disclaims any real or implied warranty with respect to fitness for any particular purpose. We cannot be responsible for any damage including but not limited to your equipment, reputation, profit-making ability, or mental or physical well-being caused by the use (or misuse) of this software.

RENTAL OR LEASING OF ANY CODEHEAD SOFTWARE PRODUCT IS STRICTLY PROHIBITED!

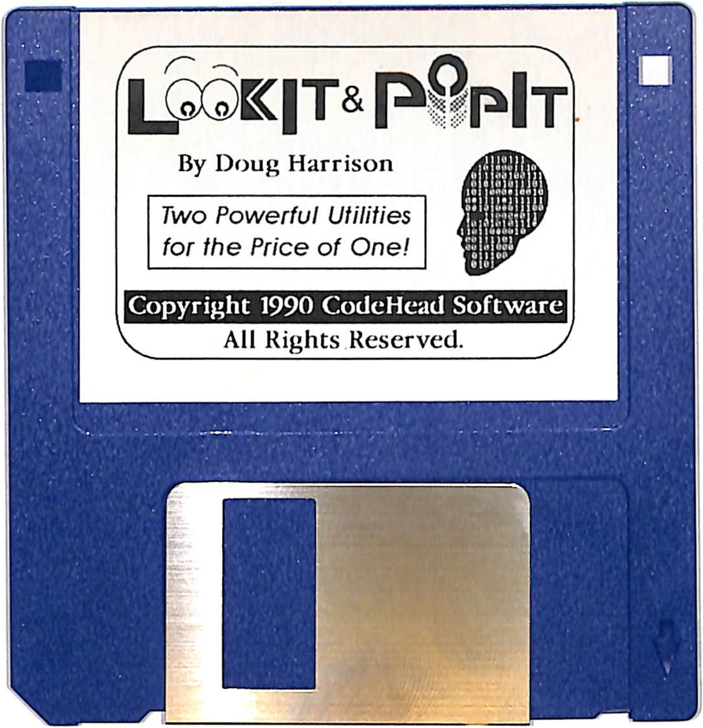
CodeHead Software
P.O. Box 74090
Los Angeles, CA 90004
(213) 386-5735

Special thanks to our beta testers: Darlah Pine, Jeff Williams, John Townsend, Mel Motogawa, Terry May, Stephen Terry, Doug Wheeler, Keith Whitton, Jason Saffer, Nevin Shalit, and the guys at ICD.

Notes

Notes

Notes



LOOKIT & POPIT

By Doug Harrison

Two Powerful Utilities for the Price of One!