DISKBACKUP

The DISKBACKUP module provides a way to make a floppy-disk backup copy of the files on a local hard-disk partition. Each file is copied to the floppy disk, then compared with the original.

Requirements

You will need an 1108/09 or an 1185/86 whose hard disk you wish to back up. You will also need to load the library module COPYFILES, or have it in a place where DISKBACKUP can find it to load it.

You will need to have on hand enough *pre-formatted* floppy disks to hold the entire partition that you intend to save. If you plan to use new, unformatted floppies, please format them before running BACKUP.

Loading the Module

Load DISKBACKUP.DCOM; it will automatically load COPYFILES if you haven't already loaded it.

Backing Up Files

The main function is

(BACKUP *VolumeName*) [Function]

VolumeName is the name of the logical volume on your hard disk that you want to back up. For example, if you keep your files on the LISPFILES volume (as most people do), you will want to type

(BACKUP 'LISPFILES).

You will be asked to load floppy disks as the program needs them; after you load each floppy, you will be asked to confirm with the left mouse button. Make sure each floppy is write-enabled.

As each file is copied onto disk, a message is printed showing progress. The "Verifying" message is printed while the copy is being compared to the original; "Done" is printed when the copy has been verified correct. If the file on the rigid disk has an illegal creation date (e.g. date and time were not set when the

file was created) then a default date will be used (1-Jan-87 12:00:00) and the message "illegal date" is displayed.

BACKUP first identifies all files which are too large to contained on a single diskette. Such files are copied to diskettes in HugePilot mode. You will be prompted to insert each diskette during the copy operation. **Be sure to write-enable the diskettes before inserting them in the drive.** When the copy operation for each large file is complete, you will be prompted to place the first floppy in the drive to start the verification step.

Once all large files have been copied and verified, BACKUP proceeds to save files which are small enough to be contained on a single diskette. As each diskette fills up, the system promps you for the next. Each floppy is labelled **Backup Floppy #N**, N = 1, 2, 3...

Troubleshooting

If the verification step for any file should fail, a break window is opened and an error message is displayed. At this point you have two options: (1) abort the BACKUP process with ↑D and start the disk backup operation from the beginning; or (2) note the name of the file which failed the verification step (for later copying) and up-arrow (↑) out of the break, which resumes the BACKUP process with the *next* file to be saved. The bad copy of the previous file is not deleted from the floppy. If verificatic fails, it is a good idea to run floppy diagnostics.

Warning: when the verification step fails, open input streams exist for the two files being compared. They should be closed before leaving the break.

Restoring Files

Files saved with DISKBACKUP can be restored to the local file system in Lyric by using either the COPYFILES.LCOM or FILEBROWSER.LCOM Lyric Library Modules. Be sure the floppy system is in the correct mode: **PILOT** or **HUGEPILOT**, before copying them from floppy.

FILEWATCH

Johannes A. G. M. Koomen
(Koomen.wbst@Xerox or Koomen@CS.Rochester)

May 14, 1987

SUMMARY

FILEWATCH is a facility for keeping an eye on open files. It continually updates a display showing each open file, its current file pointer location, the total file size, and a percentage bar.

DESCRIPTION

Invoking the function FILEWATCH (or selecting the "FileWatch" entry on the BackgroundMenu) starts up the FileWatch process if not already running, or brings up a FileWatch control menu allowing you to forget a currently displayed file (*i.e.*, stop displaying the file), recall a previously forgotten file, close an open file (user beware!), change some or all FileWatch display properties, or quit the FileWatch process. The Forget, Recall and Close entries on the FileWatch control menu have roll-outs to let you perform the operation on several files at once.

FileWatch can be customized by setting the FileWatch properties (see below) using the function FILEWATCHPROP. Right buttoning any FileWatch window brings up the FileWatch control menu, with the provision that the Forget and Close commands apply to the file displayed in that FileWatch window. Middle buttoning any FileWatch window allows you to move the entire FileWatch display, and left buttoning cause the window to be redisplayed.

DETAILS

(FILEWATCH Command) [Function]

If Command is 'ON and no FileWatch process is already running, starts a process to watch open files. If Status is 'OFF or 'QUIT and there is a FileWatch process running, kills the process. If Command is neither one of the above nor one of the FileWatch commands listed below, starts a process to watch open files if not already running, otherwise brings up the FileWatch control menu. Returns the process if running, otherwise NIL.

FORGET [FileWatch command]

Brings up a menu of files currently being watched. Select the one you no longer want to have watched.

FORGET-MANY [FileWatch command]

Repeatedly performs the FORGET command until no other files are being watched or you make a null selection.

RECALL [FileWatch command]

Brings up a menu of forgotten files. Select the one you want to have watched again.

RECALL-MANY [FileWatch command]

Repeatedly performs the RECALL command until all forgotten files are being watched again or you make a null selection.

CLOSE [FileWatch command]

Brings up a menu of open files. Select the one you want to have closed.

CLOSE-MANY [FileWatch command]

Repeatedly performs the CLOSE command until all open files have been closed or you make a null selection.

MOVE [FileWatch command]

Performs the SET-ANCHOR, SET-POSITION, and SET-JUSTIFICATION commands.

SET-ANCHOR [FileWatch command]

Brings up a menu of four corner names. Select the one on you wish to anchor the FileWatch display. For instance, selecting Top-Right causes FileWatch windows to be stacked downwards witht the top right corner of the first FileWatch window at the FileWatch display position.

SET-POSITION [FileWatch command]

Indicate where the FileWatch display should be positioned by moving the region of the combined FileWatch windows.

SET-JUSTIFICATION [FileWatch command]

Requests confirmation to turn FileWatch window justification on, *i.e.*, make all FileWatch windows the same width as the largest one.

(FILEWATCHPROP PropName [PropValue]) [Function]

If PropValue is given, sets the property value accordingly. Always returns the current (old) value of the property. This is a general facility which you can use for whatever purpose you deem appropriate. However, there are some properties that have a predefined meaning to FileWatch:

ALL-FILES? [FileWatch property]

If NIL, FileWatch displays only user visible open files; otherwise all open files (including, for example, file cacher files). Initially set to T if you are running the Koto release of Interlisp-D, otherwise NIL. (Later releases have no facility yet to obtain all open files.)

ANCHOR [FileWatch property]

Each open file that is being watched gets its own FileWatch window. Multiple windows are stacked automatically. The total region occupied by this stack is anchored at the corner indicated by this property. The only legal values are TOP-LEFT, TOP-RIGHT, BOTTOM-LEFT, BOTTOM-RIGHT. Initially set to BOTTOM-RIGHT. If the anchor is at one of the bottom corners the stack grows upward, otherwise downward. If the anchor is at one of the left corners the stack is aligned by left edge, otherwise by right edge (see also the JUSTIFIED? property).

FILTERS [FileWatch property]

A list of file patterns, for example '("{CORE}*.*;*"). An open file that matches any of the patterns will not be watched. Initially set to NIL. Note that each pattern is expanded to include the HOST and DIRECTORY equal to that of (DIRECTORYNAME), EXTENSION and VERSION equal to "*", unless already specified. For example, in my case, the filter "*JUNK*" expands to "{Ice}<Koomen>Lisp>*JUNK*.*;*". If you really wanted to filter all junk files, use the filter "{*}*JUNK*".

FONT [FileWatch property]

The font used for the FileWatch displays, specified in a form suitable to give to the function FONTCREATE. Initially set to '(GACHA 8).

INTERVAL [FileWatch property]

The value given to the function BLOCK. This should be either NIL or an integer indicating the number of milliseconds to wait between FileWatch display updates. Initially set to 1000. Note that FileWatch generates several FIXP's for large files every time throught the loop, so setting this to NIL may cause excessive storage allocation and reclamation.

JUSTIFIED? [FileWatch property]

If T all FileWatch windows are aligned along both left and right edges, and are grown or shrunk as needed to accomodate the maximum filename length currently in use. This is aesthetically more pleasing but incurs increased overhead due to frequent reshaping of the windows. Initially set to NIL.

POSITION [FileWatch property]

The location of the anchored corner of the FileWatch display. Initially set to the bottom right corner of the screen: (CONS SCREENWIDTH 0).

SHADE [FileWatch property]

The shade used for the FileWatch thermometers. Initially set to GRAYSHADE.

SORTFN [FileWatch property]

Either NIL or the name of a function taking two filenames as arguments (such as ALPHORDER), which is used to sort the list of open files being watched. Initially set to NIL (*i.e.*, no sorting).

WHO-LINE

By: sML (Lanning.pa@Xerox.com)

INTRODUCTION

Need to know what package you're in? Don't know what your connected directory is? Fret not. The Who-Line is here.

The Who-Line is a window that displays this information on your screen. It is continually updated to reflect the current state of the world (thanks to an entry on BACKGROUNDFNS).

STARTING WHO-LINE

Loading Who-Line.dfasl from any Executive will automatically open the Who-Line at the bottom of the display. If you close the Who-Line window, it can be reopened with the function (INSTALL-WHO-LINE-OPTIONS) described below

Defining the information displayed in the Who-Line

The values displayed in the Who-Line are determined by the setting of the variable *WHO-LINE-ENTRIES*.

WHO-LINE-ENTRIES [Global Variable]

WHO-LINE-ENTRIES is a list that describes the items that will be displayed in the who-line. Each item in the list should be a list of up to five things: the name of the item; a form that, when evaluated, will produce the value to display; the maximum number of characters in the value; an optional function to call if the item is selected (with the mouse) in the Who-Line; and an optional form that will reset any internal state of the entry when evaluated.

[[NOTE: Since the items on the Who-Line are evaluated rather often, it is best if they are fast and efficient (= don't CONS or allocate any space).]]

The following are standard members of *WHO-LINE-ENTRIES*.

WHO-LINE-USER-ENTRY [Variable]

Displays the current user in the Who-Line. Selecting this item in the Who-Line will let you change the logged in user.

WHO-LINE-HOST-NAME-ENTRY [Variable]

Displays the (ETHERHOSTNAME) of the machine you are running on.

WHO-LINE-PACKAGE-ENTRY [Variable]

Displays the package of the current TTY process in the Who-Line. Selecting this item in the Who-Line will let you switch the package of the current TTY process.

WHO-LINE-READTABLE-ENTRY [Variable]

Displays the (name of the) readtable of the current TTY process in the Who-Line. Selecting this item in the Who-Line will let you switch the readtable of the current TTY process.

WHO-LINE-TTY-PROC-ENTRY [Variable]

Displays the name of the current TTY process in the Who-Line. Selecting this item in the Who-Line will let you give the TTY to a different process.

WHO-LINE-DIRECTORY-ENTRY [Variable]

Displays the current connected directory in the Who-Line; the directory is shown in the format "Dir>Subdir>...>Subdir on {Host}". Selecting this item in the Who-Line will let you connect to another directory: the variable *WHO-LINE-DIRECTORIES* (see below) is used to produce a menu of interesting directories. If you are holding down a SHIFT key when you select an item from this menu, the directory name will be COPYINSERTed into the current tty input stream, otherwise you will be connected to that directory.

WHO-LINE-VMEM-ENTRY [Variable]

Displays the percentage of the VMem file that is currently being used in the Who-Line. If the VMem file is inconsistant, the number will be preceeded by an asterik ("*"). Selecting this item in the Who-Line will let you do a (SAVEVM).

WHO-LINE-TIME-ENTRY [Variable]

Displays the current time in the Who-Line. Selecting this item in the Who-Line will let you do a (SETTIME). If you hold down a shift key when you select this item, the current time will be COPYINSERTed into the current tty input stream instead.

The default value of *WHO-LINE-ENTRIES* contains all these items

Other ways to tailor the Who-Line

WHO-LINE-ANCHOR [Variable]

WHO-LINE-ANCHOR describes where the who-line will be displayed. If *WHO-LINE-ANCHOR* contains the symbol :TOP, the Who-Line will be anchored at the top of the screen; if it contains the

symbol :BOTTOM it will be anchored at the bottom of the screen. If *WHO-LINE-ANCHOR* contains the symbol :LEFT, it will be anchored to the left side of the display; if it contains the symbol :CENTER it will be centered on the screen; if it contains the symbol :JUSTIFY it will run the width of the screen; if it contains the symbol :RIGHT it will be anchored to the right side of the screen. Finally, if *WHO-LINE-ANCHOR* is a POSITION, it will be used as the lower left corner of the Who-Line. The default value is (:CENTER :BOTTOM).

WHO-LINE-NAME-FONT [Variable]

The font used to display the names of the items in the who-line. The default is HELVETICA 8 BOLD.

WHO-LINE-VALUE-FONT [Variable]

The font used to display the values in the who-line. The default is GACHA 8.

WHO-LINE-COLOR [Variable]

The color of the Who-Line. Legal values are the keywords :WHITE and :BLACK. The default is :WHITE.

WHO-LINE-BORDER [Variable]

The border width of the Who-Line window. The default is 2.

WHO-LINE-TITLE [Variable]

The title of the Who-Line window. The default is NIL.

WHO-LINE-DISPLAY-NAMES? [Variable]

If *WHO-LINE-DISPLAY-NAMES?* is true, the names of items in the who-line will be displayed; otherwise they will not be shown. The default value is T.

WHO-LINE-UPDATE-INTERVAL [Variable]

The number of milliseconds between updates of the who-line. The default is 100 milliseconds.

Installing new Who-Line options

Changing the above variables has no direct effect on the who-line. These values need to be installed in the Who-Line before they can take effect.

(INSTALL-WHO-LINE-OPTIONS) [Function]

INSTALL-WHO-LINE-OPTIONS installs the above options in the Who-Line, and updates the Who-Line accordingly.

Who-Line process state

The who-line entry *WHO-LINE-TTY-STATE-ENTRY* tries to display the current state of the TTY process.

WHO-LINE-TTY-STATE-ENTRY [Variable]

A Who-Line entry that displays the "state" of the current TTY process in the Who-Line. The typical state of a process is the name of the function that is currently running in that process. This simple minded result can be altered by use of the following items.

[[NOTE: Because of the nature of the Xerox Lisp scheduler, this information is almost always out of date.]]

The Who-Line "state" can be explicitly controlled from code. If the special variable *WHO-LINE-STATE* is bound, its value is taken to be the state of that process. You can use this feature to provide visual indiation of the state of your code by using the programming idiom:

```
(LET ((*WHO-LINE-STATE* indicator))
 (BLOCK)                          ;Give the Who-line a chance to run
 ...your-code...)
```

This will run the *...your-code...* with the Who-Line state of the process set to (the value of) *indicator*. The call to BLOCK insures that the Who-Line has a chance to update before *...your-code...* is run.

WHO-LINE-STATE-UNINTERESTING-FNS [Global Variable]

If there is no declared who-line state (via a WITH-WHO-LINE-STATE form), then the name of the function that is currently running is used as the who-line state. However, if the function is on the list *WHO-LINE-STATE-UNINTERESTING-FNS*, the function that called *it* is used instead. The default value of *WHO-LINE-STATE-UNINTERESTING-FNS* is (BLOCK AWAIT.EVENT).

WHO-LINE-STATE [Property]

If the function that is currently running has a WHO-LINE-STATE property, the value of that property is used as the who-line state. This is used to convert functions like \TTYBACKGROUND to meaningful values like "TTY wait".

(WHO-LINE-REDISPLAY-INTERRUPT) [Function]

Updates the Who-Line. It is intended that this function be installed on an interrupt character, so that the user can easily force an update of the Who-Line. For example,
(ADVISE 'CONTROL-T 'BEFORE '(WHO-LINE-REDISPLAY-INTERRUPT))
will cause a ↑T interrupt to update the Who-Line as well as its current behavior of printing state

information in the Prompt window. Alternatly, you can define a new interrupt character that will force an update of the Who-Line;
(INTERRUPTCHAR (CHARCODE ↑U) '(WHO-LINE-REDISPLAY-INTERRUPT) 'MOUSE)
will cause the Who-Line to be updated whenever the user hits a ↑U.

Other interesting things

WHO-LINE-DIRECTORIES [Global Variable]

A list of interesting directories used to generate a pop-up menu of directories to connect to when you select the DIRECTORY item in the Who-Line. The default value is a list containing just your LOGINHOST/DIR.

(CURRENT-TTY-PACKAGE) [Function]

Returns the name of the package of the current TTY process. This function is used in the default value of *WHO-LINE-ENTRIES*.

(CURRENT-TTY-READTABLE-NAME) [Function]

Returns the name of the readtable of the current TTY process, or the string "Unknown" if it can't figure out the name. This function is used in the default value of *WHO-LINE-ENTRIES*.

(SET-PACKAGE-INTERACTIVELY) [Function]

Pops up a menu of currently defined packages. If the user selects one of them, the current package is changed to the selected package.

(SET-READTABLE-INTERACTIVELY) [Function]

Pops up a menu of currently known readtables. If the user selects one of them, the current readtable is changed to the selected readtable.