HOTLINE!

Bulletin 6
November 6, 1987

HOTLINE! is published periodically by the Customer Support group of Xerox Artificial Intelligence Systems to assist its customers in using the Xerox Lisp environment. Topics covered include answers to questions that are most frequently asked of Customer Support, suggestions to help you work in the Xerox Artificial Intelligence Environment (XAIE) as well as announcements of known problems that may be encountered.

Feel free to make copies of individual bulletin pages and insert them in the appropriate place(s) in your Interlisp Reference Manual, Lisp Library Modules manual or other relevant manual. The documentation reference at the end of each topic can be used as a filing guide.

For more information on the questions or problems addressed in this or other bulletins please call us toll-free in the Continental United States 1-800-228-5325 (or in California 1-800-824-6449). Customer Support can also be reached via the ArpaNet by sending mail to AISUPPORT.PASA@Xerox.com, or by writing to:

Xerox AIS Customer Support
250 North Halstead Street
P.O. Box7018
Pasadena, CA 91109-7018
M/S 5910-432

In this issue

In response to user requests we have decided to have *HOTLINE!* cover all supported releases of XAIE, instead of Lyric only. Supported releases include Koto and Lyric. Each item now contains a "Release" field for any item that is release specific.

- COPYFILE to floppy LOGXOR break
- Error found when installing a sysout from floppy: "File name not found"
- Error found when installing a sysout from floppy: "Floppy label error"
- Disk scavenging
- How to recover from internal garbage collection table overflow
- How to diagnose the cause of internal garbage collection table overflow
- LOGOUT resets the TTY parameters
- Open RS232 stream
- Using the left cluster keys in SEdit
- Changing fonts in SEdit

Terminology

Terminology used in this *HOTLINE!* bulletin:

UG – Users' Guide

AR – Action Request, a Xerox problem tracking number (e.g. AR 8321)

IRM – Interlisp Reference Manual

COPYFILE to floppy LOGXOR break

Release Koto

Keywords COPYFILE, floppy, LOGXOR, break, SETTIME

Problem COPYFILE to floppy results in LOGXOR break if file was made when time was not set.

Example If you did a MAKEFILE with the time not set the CREATIONDATE of the file will be something like "31-DEC-00 16:05:03." If you attempt to copy this file to a floppy you will get the break:

```
NON-NUMERIC ARG
NIL
(LOGXOR broken)
```

Workaround If this is a source file, data file or TEdit file you should first set the time on your machine using the function SETTIME with the current date and time. If the file is a source file you should then load it and then reMAKEFILE so that it has the valid date. With a TEdit file you can do a GET and then a PUT to resave it with the valid date. With other types of files you can use COPYFILE to put a new copy back to the source device with a valid date. You can then copy the file to a floppy.

If this is a compiled file and you have no access to the source, proceed as follows when the LOGXOR break window occurs:

Middle mouse button (or press the left and right mouse buttons simultaneously on a two button mouse) in the break window and select "BT" from the menu. Middle mouse button on the item OPENSTREAM in the back trace window and a window will be attached to the break window showing the argument names of OPENSTREAM and the current values. Using the left mouse button, select the value of the *PARAMETERS* argument. The value should now be inverted. Middle button the value and a menu should appear. Select "Inspect" from this menu. The inspector window that is now opened should have an item number with a CREATIONDATE list next to it. It will look something like (CREATIONDATE "31-DEC-00 16:05:03"). We are now going to change this date to something valid. Select the item number associated with the CREATIONDATE with the left mouse button, which should invert it, and then middle button on the item number, which should bring up a menu with a single item "Set." Choose "Set" and a window will be attached to the inspector with a prompt, indicating "The expression read will be evaluated." At this point we are now ready to type in a "valid" date. Type in something like:

```
'(CREATIONDATE "12-SEP-87 12:00:00")
```

Note the quote preceding the CREATIONDATE list. This is needed since we don't want the expression evaluated. After hitting RETURN the inspector should now show the CREATIONDATE item with the new value.

Go back to the original break window, middle button in it to bring up the break menu and select "Revert." This should close the

break window and bring up another break window indicating "(OPENSTREAM broken)." Middle button in this break window and select "OK" from the break menu. The COPYFILE should now complete successfully.

Error when installing a sysout from floppy: "File name not found"

Release Koto and Lyric

Keywords Sysin!, FLOPPY.MODE

Question I wanted to specify a sysout name other than lisp.sysout, so I copied my sysout to floppy with FLOPPY.MODE set to HUGEPILOT. When I tried to SYSIN! from floppy, I got the message "Floppy error: File name not found." What should I do?

Background If a sysout is copied to a floppy with the mode set to SYSOUT, the sysout name defaults to lisp.sysout. With mode set to HUGEPILOT, the user is permitted to specify a file name other than lisp.sysout, e.g. somename.sysout. The file name will be written to floppy with a version number, somename.sysout;1. The floppies will automatically be formatted and named properly, e.g 'Somename Sysout #1.'

Answer To install this sysout from the System Tool, the user must specify the file name complete with version number, e.g. somename.sysout;1. Upon selecting the sysin! command the user will be prompted to insert the floppy labeled 'Somename Sysout #1.'

If the user neglects to provide a version number, the message 'Floppy Error: File name not found' will appear in the message sub-window of the System Tool.

Reference See "Floppy Disk Support" under the INPUT/OUTPUT chapter of the appropriate Users' Guide.

Error when installing a sysout from floppy: "Floppy label error"

Release Koto and Lyric

Keyworks Sysin!, FLOPPY.NAME

Question My installation of a sysout from floppy fails because the sysout floppies do not appear to have the correct name. How can I recover?

Background When installing from the floppy device, the name of the floppy to be inserted is computed from the file name specified. That is, if the user has specified a file 'somename.sysout,' the first floppy they will be prompted to install is 'somename sysout #1.'

If the sysout is written to a floppy using the SYSOUT function or COPYFILE with floppy mode set to SYSOUT or HUGEPILOT, and the floppies have not been explicitily renamed, this problem will not occur. By default, these functions correctly name and format the floppies.

Answer Use the Interlisp function FLOPPY.NAME to reset the floppy name of every floppy in the sysout.

For example, assume the file path name specified in the System Tool is Colossus.sysout and the device specified is floppy. When the user selects sysin! the following message will appear:

```
Insert floppy labeled 'Colossus Sysout #1.' Confirm to
continue
```

If you insert floppy 'Colossus Version 3.7 Sysout #1' and then confirm, the following will appear:

```
Retrieving . . . Colossus.sysout
The floppy is labeled 'Colossus Version 3.7 Sysout #1'
Insert floppy with label 'Colossus Sysout #1' Confirm when
ready else to exit
```

At this point you should click on the right mouse button. The volume will be erased. You must rename each of the floppies from Interlisp using the following:

```
(FLOPPY.NAME "Colossus Sysout #n")
```

where n ranges from 1 to the number of floppies contained in the sysout.

Reference See "Floppy Disk Support" under the INPUT/OUTPUT chapter of the appropriate Users' Guide.

Disk scavenging

Release Koto and Lyric 1186

Keywords Hard disk error, scavenge, physical volume scavenge, disk pages

Question What should I do when I get a "needs scavenging" error when I attempt to scavenge a logical volume (e.g. the Lispfiles volume) from the System Tool?

Background Usually, when one is having problems with the hard disk (as identified by a HARD DISK ERROR message/break in Lisp when attempting to read a file or evaluate a directory), one will attempt to scavenge the Lispfiles logical volume from the System Tool. Under certain circumstances, a message will be returned stating that the disk needs scavenging.

Answer This message indicates that the physical volume needs scavenging before any of the logical volumes can be scavenged. The solution is to scavenge the physical volume of the disk and then scavenge each of the logical volumes.

Using your Offline Diagnostics floppies (there are two of them), press the B Reset button and insert the Offline Diagnostics Disk #1 in the floppy drive. Press the F2 key on your keyboard and when prompted to insert the Offline Diagnostics Disk #2, please do so.

When prompted as to "What class of user do you belong to?" answer "1 - Normal User." At this point another menu will appear. Select "4 - Formatter, Scavenger, and Bad Page Utility." At the next menu, to scavenge the physical volume, select "1 - Run Physical Volume Scavenger."

The "Run Physical Volume Scavenger" selection provides a way to correct minor problems on the local rigid disk without destroying the data.

The utility will prompt:

```
"Should I perform safe repair?"
```

Type YES. This will repair data structures without destroying your data.

If the physical volume scavenge does not complete successfully, call your system administrator or your Xerox hardware service representative for further assistance.

After successfully scavenging from the Offline Diagnostics, you can go back to the System Tool and scavenge each logical volume on your disk.

A successful completion (takes no more than 10 - 30 seconds) looks as follows:

```
Scavenging ... Completed
No problem found
```

Note to 1108 Users: If you find that you need to run the physical volume scavenger, call your Customer Support representative for more information on how to execute this procedure.

References Xerox 1186 Users' Guide, Lyric Release, pages 42 - 44, and pages 116 - 117.

How to recover from internal garbage collection table overflow

Release Koto and Lyric

Keywords Garbage collection, reference count > 1

Question What do I do when the system prompts me with: "Internal garbage collector tables overflowed. Too many pointers with reference count greater than 1 *** the garbage collector is now disabled*** save your work and reload as soon as possible"?

Answer Once the internal garbage collection is full, you will get the "GC Disabled Warning" message. After the message appears, the garbage collecter is disabled. (There is no way to enable it again.) If you ignored the "GC disable warning" message, you would be heading for a fatal loss of work. This means that you will eventually get the message "Your virtual memory backing file is nearly full. Save your work and reload a.s.a.p".

So when the system prompts the "GC Disabled Warning" message, we recommend that you save the work. Check (VMENSIZE) and (VOLUMESIZE VolumeName). If (VMENSIZE) is close to (VOLUMESIZE), run (GAINSPACE) prior to saving your work (MAKEFILE).

When running (GAINSPACE), the system prints a list of questions to allow you to specify at each point what should be discarded and what should be retained. You should make sure that the answers for the following questions are NO:

- erase properties?
- function definitions on property lists?
- advice information?
- erase CLISP translations?
- erase system hash array?
- erase filepkg information?

Reference IRM 22.11-22.12

How to diagnose the cause of internal garbage collection table overflow

Release Koto and Lyric

Keywords Garbage collection, reference count >1

Question My application is failing with the error message: "Too many objects with reference count >1." What should I do to determine the cause of this problem?

Answer Actually the message is misleading. It should read "Too many objects with reference count not equal to 1." To diagnose the problem, you will need to load GCHAX and use \SHOWGC to see the type of objects or pointers that fill up the table. The argument "onlytypes" allows you to specify which data types to check; the default is to check all data types. You should also set the mincnt argument to 0, so objects with reference count 0 will also be counted.

Reference Lisp Library Modules manual, GCHAX

LOGOUT resets the TTY parameters

Release Lyric

Keywords RS232, DLTTY

Problem LOGOUT resets the TTY port parameters, but not to the values specified by the variable TTY.DEFAULT.INIT.INFO.

Symptom A TTY device fails to work after you have returned from a LOGOUT.

Workaround After every LOGOUT, you must re-initialize the TTY parameters with the TTY.INIT function.

One way to do this is by defining a function to initialize the TTY parameters, and adding this function to the variable AROUNDEXITFNS. If the function is defined with the single argument AFTERLOGOUT, it will be called whenever you return from a LOGOUT.

If you have accidentally attempted to use the TTY port before resetting the parameters, the port may become "stuck." Then you must LOGOUT, log back in, and use TTY.INIT to re-initialize the port parameters.

Example You want the TTY port to be initialized to 4800 baud, 8 bits, no parity, 1 stop bit, and XOnXOff enabled.

With SEdit, edit the variable AROUNDEXITFNS. Add the following function to the list of functions:

```
(LAMBDA (AFTERLOGOUT) (TTY.INIT4800 8 'NONE 1 'XOnXOff))
```

This function will be called everytime you return from a LOGOUT.

References AR #9278
Lisp Library Modules manual, Lyric Release, pages 212-213.
Lyric Release Notes, Changes to Interlisp-D, pages 58-59.

Open RS232 stream

Release Lyric

Keywords RS232, RS232C, RS232CMENU

Problem RS232C.GET.PARAMETERS or RS232CMENU "Show!" leaves the RS232 device stream open.

Symptom An RS232 device fails to work after you have called the function RS232C.GET.PARAMETERS; or after you have used the "Show!" command provided by RS232CMENU.

After either operation, an attempt to use the RS232 port will display the following message in the Prompt Window:

```
RS232 port is busy on output
```

Note: Open RS232 streams are not returned by the function OPENP.

Workaround The function RS232C.SHUTDOWN should be called to close the RS232 stream.

References AR #9281
Lisp Library Modules manual, Lyric Release, page 208.

Using the left cluster keys in SEdit

Release Lyric

Keywords SEdit, keypad

Question How are the left cluster keys used in SEdit?

Note: This bulletin (page 6.9) corrects a printing error in Issue #4 of *HOTLINE!*

Answer The functions of the left cluster keys are described below.

STOP	DEL
UNDO	AGAIN
FIND	COPY
SAME	MOVE
HELP	EDIT

1186 Left Cluster

AGAIN	DELETE
FIND	COPY
SAME	MOVE
OPEN	PROPS

1108 Left Cluster

STOP — Redisplays the SEdit window.

On the 1108 keyboard, STOP is in the right cluster keypad.

UNDO — Undoes the last edit (Meta-U).

On the 1108 keyboard, UNDO is in the right cluster keypad.

FIND — Finds the next instance of a specified structure (Meta-F).

SHIFT-FIND — Finds and replaces a structure (Meta-S).

SAME — No function.

HELP — Shows the argument list for the function currently selected (Meta-H).

On the 1108 keyboard, HELP is in the right cluster keypad.

If the SEdit window is too small, the forms will be displayed in the PROMPT window.

DEL — Deletes the selected part of a structure.

AGAIN — Redoes the edit change that was just undone (Meta-R).

COPY — Copies a structure from one location to another.

1. Position the SEdit caret at the destination for the text.
2. Hold the COPY key down while using the mouse to select the structure to be copied.
3. Release the COPY key to execute the operation.

MOVE — Moves a structure from one location to another.

Use the same steps as COPY.

EDIT — The CONTROL key.

May be used as a delete key.

1. Hold the EDIT key down.
2. Select the structures to be deleted.
3. When the key is released, the structures will be removed.

On the 1108 keyboard, the EDIT key is labeled as PROPS.

OPEN — The META key for the 1108 keyboard.

References

Lyric Release Notes, Appendix B. SEdit - The Lisp Editor.
1186 Users' Guide Lyric Release, pages 16-18.
1108 Users' Guide, page 14.

Changing fonts in SEdit

Release Lyric

Keywords SEdit, fonts

Question Can I change the display fonts for SEdit?

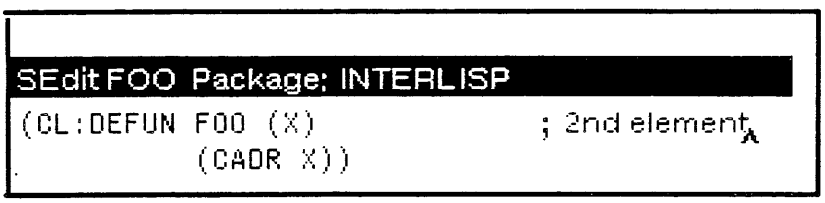
Note: This bulletin (page 6.10) corrects a printing error in Issue #4 of *HOTLINE!*

Answer Yes. The SEdit display fonts can be modified by editing the global variable FONTPROFILE. On this variable, the SEdit text font is determined by the list associated with DEFAULTFONT, and the SEdit comment font is determined by the list associated with COMMENTFONT. The third element in each list determines the display font; so these are the values that should be modified to change the SEdit display fonts.

After the variable FONTPROFILE has been modified, you should call the function FONTPROFILE to set up the modified font classes – DEFAULTFONT and COMMENTFONT. Then the function SEdit.RESET must be called to put these changes into the SEdit environment. This change will effect SEdit in both the XCL and the Interlisp Exec.

Since you probably do not want to change the system environment's DEFAULTFONT and COMMENTFONT; FONTPROFILE should be restored to its original value by re-editing this variable and calling FONTPROFILE again. The SEdit fonts will not change back to their original values unless SEdit.RESET is called again.

Example You want to make your SEdit fonts larger and bolder. Currently, a function defined in SEdit looks like this:



In this example, the variable FONTPROFILE includes the sub-lists:

```
(DEFAULTFONT 1 (GACHA 10) (GACHA 8) (TERMINAL 8))
(COMMENTFONT 6 (HELVETICA 10) (HELVETICA 8) (MODERN 8))
```

With SEdit, you edit FONTPROFILE, changing (GACHA 10) to (GACHA 12 BRR), and (HELVETICA 10) to (HELVETICA 12 BRR):

```
(DEFAULTFONT 1 (GACHA 12 BRR) (GACHA 8) (TERMINAL 8))
(COMMENTFONT 6 (HELVETICA 12 BRR)(HELVETICA 8)(MODERN 8))
```

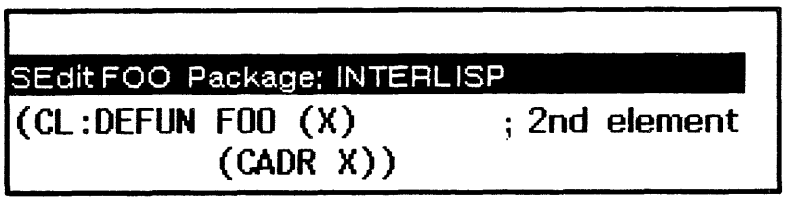
Then you save the new font classes by calling the function FONTPROFILE on the variable FONTPROFILE:

```
(FONTPROFILE FONTPROFILE)
```

Finally, you put these changes into the SEdit environment by calling SEDIT.RESET. All SEdit windows must be closed before this function can be called.

```
(SEDIT.RESET)
```

Now, editing FOO again looks like this:



Finally, you should restore the original value of the variable FONTPROFILE.

References Lyric Release Notes, SEdit - The Lisp Editor, page B-11.
Interlisp-D Reference Manual, Volume 3: Input/Output, Section 27-13, 27-14.