

6. ERROR RECOVERY

Medley on the Sun Workstation has an error handling system which includes the following:

- The Xerox Lisp error system, described in the IRM
- A diagnostic program, URAID, which handles emulator errors

Occasionally, you may encounter SunOS error messages. Refer to your Sun documentation set for recovery procedures when these errors occur. When running Medley on a Sun Workstation, previous Lisp error handling such as Teleraid and MP errors are no longer available. However, you can still use Teleraid from a Sun Workstation to debug a Xerox 1100 series workstation.

URAIID

The Medley system normally operates as a self-contained environment. In some unusual circumstances Medley may encounter a situation from which it cannot recover. In this case, when an unrecoverable emulator error is encountered, the emulator halts and enters into a small debugger called URAID. URAID allows you to inspect memory, or to look inside the sysout file, and attempt to recover from the error.

If you produce the same type of error condition in Medley on a Sun Workstation as you did on a Xerox workstation, you get a URAID error instead of an MP error.

Entering URAID

Normally, the emulator automatically enters URAID when an unrecoverable emulator error occurs. However, there are two additional methods available when you want to enter URAID directly.

- Use the SHIFT-CTRL-DELETE key combinations to enter URAID between opcodes. Note that the DELETE key referred to here is in the L10 position on the left keypad of the Sun keyboards. This sequence allows you to return to Lisp later.
- Use SHIFT-CTRL-NEXT for emergency interrupts only. Note that the NEXT key is in the ALTERNATE key position on the regular Sun keyboard. These combinations are useful for exiting from an opcode infinite loop. SHIFT-CTRL-NEXT does not necessarily enter URAID between opcodes; once you are in URAID mode, another URAID command such as "f" could cause the emulator to crash. At this point it is unlikely that you could return to Lisp. USE WITH CAUTION!

Conventions

URAIID uses these display conventions:

- Numbers are displayed in hexadecimal unless otherwise noted.
- The *litatom* should be an uppercase string when used with a package prefix (e.g., XCL:EVAL).
- Symbols re displayed with a package prefix, but with no escape character.

- Symbols in the Interlisp (IL:) package are case-sensitive (e.g., IL:\InterfacePage); symbols in other packages are case-insensitive (e.g., XCL:EVAL).

In addition, these input conventions apply:

- Symbols may only be qualified by their home package.
- A full package name may prefix an input symbol. URAID also supports approved abbreviations of package names (e.g., XCL:, SI:, CL:, XCLC:).

A symbol without a prefix is treated as a symbol in the Interlisp package. For instance, `\InterfacePage` is the equivalent of `IL:\InterfacePage`.

- Type-in is uppercase for symbols in any package except the Interlisp package; type-in is in mixed case for `IL:` package symbols or symbols with no prefix.

URAIID Commands

URAIID has a few simple commands which you can use to attempt diagnosis and error recovery. All URAID commands are case-sensitive.

- h** Hard Reset. Attempts to recover by resetting the Lisp stack. Quits URAID and causes Lisp to resume execution. This command should not be used unless you are sure that execution can be resumed.
- e** Exits to SunOS. Medley will end.
- q** Quits URAID and returns to Lisp.

NOTE: An error may occur while the Medley system is running uninterruptably. The following message signals this error:

```
Error in uninterruptable system code -- ^N to continue into
error handler
```

Disregard the `^N` command; it is not supported by URAID. Use the **q** command to continue.

Displaying a Stack

For casual users, the **l** command followed by several **f** commands generally provides the most useful information. Many of the other commands require some knowledge of the internal representation of Lisp objects and stack frames.

- c** Checks all user stack contents; stack inconsistency is displayed.
- k type** Changes the stack link that precedes the **l** command to be *type*, which is either **a** (to follow ALinks) or **c** (to follow CLinks). The default is to trace ALinks. ALinks follow the chain of free variable access.
- l type** Shows the stack as a back trace consisting of a numbered sequence of frame names. The default is the user stack. The argument *type* is a single letter denoting the stack to view. The system has a number of special contexts, which are areas of stack space used by certain system routines. Legal values of *type* are as follows:
 - g** (garbage collect)
 - k** (keyboard handler)
 - m** (miscellaneous)
 - p** (page fault)
 - r** (reset)
 - u** (user stack) - Default

type := g|k|m|p|r|u or nil

- C** Checks the contents by scanning all stack space in the sysout. For example:

```
0x11880 BF,[ivar:0x1800]
0x11802: FX for CL:T[ ]
0x11816 BF,[ivar:0x1816]
0x11818: FX for IL:\TURN.ON.PROCESSES[ ]
```

Viewing Frames From a Stack

After displaying a particular stack with the **I** command, the following commands view individual frames from that stack:

- f number** Displays the contents of frame *number* (decimal) with its basic frame, IVars and PVars. The frame is printed in two parts, a basic frame containing the function's arguments and a frame extension containing control information, the function's local (PROG) variables, and dynamic values. On the left side of the printout are the hexadecimal contents of each cell of the frame, with an interpretation, usually as a Lisp value, on the right. The following message appears as you display a frame with the **f** command:
- Press Return (To quit ESC and RET)
- To abort the printing of a frame, first press the ESC key then the RETURN key. The URAID prompt "<" reappears.
- <CR>** Displays the next frame (closer to the root, or bottom, of the stack). This is the same as **f n+1**, where *n* is the number of the frame most recently viewed. Immediately after an **I** command, *n* is zero, so **<CR>** views the first frame.
- a litatom** Displays the top-level value of the *litatom*
- d litatom** Displays the contents of definition cell for the *litatom*. If it is compiled code, this command prints a CCODEP hexadecimal address pointer; for example,
- {CCODEP}0x14ccc4
- Otherwise, it prints a Lisp definition; for instance, interpreted code returns
- (LAMBDA () ...)
- M** Displays TOS, CSP, PVar, IVar, PC.
- m func1 func2** Moves the definition of *func1* to *func2*.
- t Xaddress** Displays the type of this object.
- p litatom** Displays the contents of the *litatom*'s property list.
- w** Displays the current function name and PC.
- x Xaddress[Xnum]** Prints *Xnum* word (16-bits) of the raw contents of the virtual memory starting at virtual address *Xaddress*. This

is most useful for examining the contents of a datatype which other commands simply print as its virtual address.

@litatom[snumber | NIL | T] Sets the TOPVAL of *litatom* to the specified value. *snumber* is a signed smallp number.

<Xaddress val Sets the the contents of the word (16-bits) at the *Xaddress* to *val*.

Miscellaneous

v filename Saves the current virtual memory on the *filename*. This file can be examined using the functions READSYS and VRAID in the TeleRaid Lisp Library module, but cannot be used as a sysout file.

NOTE: This sysout cannot be restarted.

s Invokes a subshell.

(num Sets the print level (default is 2).

? Displays this summary.

! Prints the error message passed from the emulator.

Other Fatal Error Conditions

Occasionally, other emulator, operating system, or system administration errors may occur from which the URAID program cannot recover. Such error conditions include the process dying, the emulator going into an infinite loop, the keyboard being lost, or the system freezing up.

If any of these emulator errors occur, use the UNIX `kill` command to kill the `lde` process.

Lisp Errors

Errors While Running Medley

The following Lisp errors may occur when running Medley on the Sun Workstation.

<u>ERROR MESSAGE</u>	<u>CAUSE</u>
File access timed out	Occurs when you try to access a file when the remotely mounted file system or NFS service is down, or when network traffic is heavy. See the File System Errors subsection of Chapter 5.
File too large	Self-explanatory.
Too-Many-Files-Open	Occurs when you exceed one of the following: <ul style="list-style-type: none"> SunOS open file limit (see Chapter 5, Medley File Systems)

<p>Nonexistent directory</p> <p>No-Such-Directory</p> <p>Connection timed out</p> <p>Bad Host Name</p> <p>FS-RENAMEFILE-SOURCE- COULDNT-DELETE</p>	<ul style="list-style-type: none"> • System file resources while writing a sysout (using IL:SYSOUT) <p>Occurs when you try to connect to a nonexistent directory using the IL:CNDIR function or the CONN command.</p> <p>CHDIR</p> <p>Self-explanatory.</p> <p>Self-explanatory.</p> <p>Occurs when you try to rename a file which exists on a directory or which you do not have delete permission.</p>
--	---

Xerox Workstation-Specific Errors

These Xerox workstation-specific errors may occur if certain functions are inadvertently used on the Sun Workstation.

<u>ERROR MESSAGE</u>	<u>CAUSE</u>
<p>Floppy: No floppy drive on this machine.</p> <p>Device error: {FLOPPY}</p> <p>Wrong machinetype</p>	<p>Self-explanatory.</p> <p>Occurs when the user tries to enter a Lisp floppy function while running on the Sun Workstation.</p> <p>Occurs when functions controlling Xerox disk drive device-specific behavior are entered while running in SunOS.</p>

Virtual Memory Errors

<u>ERROR MESSAGE</u>	<u>LISP FUNCTION RESPONSIBLE</u>
<p>File-System-Resources-Exceeded</p> <p>Protection-Violation</p> <p>File-Wont-Open</p>	<p>IL:SYSOUT, IL:LOGOUT, IL:SAVEVM</p> <p>IL:SYSOUT, IL:LOGOUT, IL:SAVEVM</p> <p>IL:SYSOUT, IL:LOGOUT, IL:SAVEVM</p>

[This page intentionally left blank]