

GL2-W3.4 Workstation Release Notes

Version 1.0

**Silicon Graphics, Inc.
2011 Stierlin Road
Mountain View, CA 94043**

Document Number 007-3203-010

Conventions

This document uses the standard UNIX convention for referring to entries in the UNIX documentation. The entry name is followed by a section number in parentheses. For example, *cc(1)* refers to the *cc* manual entry in Section 1 of the *UNIX Programmer's Manual*.

In command syntax descriptions and examples, square brackets ([]) surrounding an argument indicate that the argument is optional. Words in *italics* represent variable parameters, which should be replaced with the string or value appropriate for the application.

In text descriptions, filenames and UNIX commands are written in *italics*. IRIS Graphics Library and PROM monitor commands are written in *typewriter font*.

Acknowledgments:

Marcia Allen
Gail Kesner
Susan Luttner

Special thanks to Susan Ellis and Donl Mathis.

© Copyright 1986, Silicon Graphics, Inc.

All rights reserved.

This document contains proprietary information of Silicon Graphics, Inc., and is protected by Federal copyright law. The information may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without prior written consent of Silicon Graphics, Inc.

The information in this document is subject to change without notice.

GL2-W3.4 Workstation Release Notes

Document number: 007-3203-010

IBM is a trademark of the International Business Corporation.

UNIX is a trademark of AT&T Bell Laboratories.

XNS is a trademark of Xerox Corporation.

Ethernet is a trademark of Excelan Corporation.

GL2-W3.4 Workstation Release Notes

These Release Notes describe the features of Software Release GL2-W3.4 for IRIS series 3000 workstations and IRIS series 2000 workstations with the Turbo Option upgrade. The seven sections of this document cover these topics:

1. Major Enhancements
2. Shared Memory
3. Installing Software Updates and Options
4. Additions to Graphics, Systems, and Options
5. Changes to Graphics, Systems, and Options
6. Bug Fixes to Graphics, Systems, and Options
7. Known Problems

Because of the level of increased functionality provided in this release, you must recompile all programs after installing this release.

1. Major Enhancements

The GL2-W3.4 workstation release provides these major enhancements:

- System V shared memory (see Section 2 below).
- Improved virtual memory subsystem performance.
- Many improvements to the window manager *mex(1G)*, including support for pop-up menus and multiple windows per process. Examples of window manager programs are contained in */usr/people/gifts/mextools*; see the *README* in that directory to use these programs.
- Language tools enhancements:
 - You can make interlanguage calls between C and FORTRAN directly (without wrappers) in both directions. The C compiler storage class *fortran* now supports these calls. See the *C/FORTRAN Interface* document in the version 1.2 update to the *UNIX Programmer's Manual*, volume IIB, which is distributed with this release.

Version 1.0

- The source-level debugger *dbx* now debugs FORTRAN programs. See the *dbx* tutorial (C and FORTRAN versions) for new users.
- A new **+s** option to the FORTRAN compiler adds a SAVE statement to the beginning of each routine. This option causes variables local to a subroutine to retain their values between calls. See the section on the *FORTRAN f77 Revision Notes* in the *C/FORTRAN Interface* document in the version 1.2 update to the *UNIX Programmer's Manual*, volume IIB.
- Pascal users now have the same control as FORTRAN users over floating-point exceptions. See *IRIS Floating Point* in the version 1.2 update to the *UNIX Programmer's Manual*, volume IIB.
- The default rules of *make(1)* reflect this change in object file names. You will need to update your makefiles. See the *C/FORTRAN Interface* (FORTRAN users) or version 2.0 of the *Pascal Release Notes for the Graphics Library* (Pascal users).

The standard UNIX archiver *ar(1)* replaces the *library(1)* utility, eliminating library cross-referencing problems.

- A local Pascal Graphics Library.

2. Shared Memory

The System V shared memory facility is now fully supported. Shared memory operates under these restrictions:

- The maximum shared memory size (SHMMAX) is 64kb, or 0x10000 bytes.
- The minimum shared memory size (SHMMIN) is 1 byte.
- The maximum number of shared memory segments per system (SHMMNI) is 40.
- The maximum number of shared memory segments per process (SHMSEG) is 10.
- The maximum number of shared memory pages per system (SHMALL) is 0x40, or 256kb of memory.
- The default separation between the end of *bss* space (see the manual page for *sbrk(2)*) and the first system-chosen shared memory segment address (SHMBRK) is 32768 bytes. This parameter controls where the system places a shared memory segment when you use the *shmat* command on the segment with an address of zero.

All restrictions are applied simultaneously. While a process may not use up its number of segments-per-process limit, it may still overflow the SHMMNI limit, or, more likely, the SHMALL parameter.

Under this implementation, the shared memory segments are not paged. Therefore, using many shared memory segments reduces the amount of memory for user programs. This increases paging and slows down the system. By default, shared memory segments are not destroyed when a process exits. *ipcrm(1)* deletes shared memory segments.

Shared memory manipulation is described in these manual pages:

- *ipcrm(1)*
- *ipcs(1)*
- *shmctl(2)*
- *shmget(2)*
- *shmop(2)*

3. Installing Software Updates

Before installing updates, it is important to back up your user files onto tape and make sure you have a bootable back-up tape on hand.

NOTE
If you have a problem during the update procedure, resolve it before continuing. Failure to perform certain steps successfully can render the system unusable. If you encounter a problem that you cannot resolve, contact the Geometry Hotline at 800/345-0222 within California (toll-free), 800/252-0222 elsewhere in the U.S. (toll-free), or 800/443-0222 for Canada.

3.1 Installing from Tape

To install software updates and options, follow the steps below. Updates and options can be installed separately or together. The basic procedure assumes that only one machine is involved. If the tape drive is on a remote machine to be accessed using XNS, see "Network Installation" at the end of this section.

1. Log in as root. Make sure the system is in multi-user mode, but that there are no other users logged on.
2. Change your working directory to the *root* directory:

```
cd /
```

3. If there is an old copy of the distribution directory present, remove it:

```
rm -rf dist
```

4. Put the distribution tape in the tape drive.
5. Read in the distribution tools with *cpio*:

```
cpio -ivhmud1
```

6. To make it easier to access the distribution tools, put the */dist* directory in your search path. Enter this command carefully; if the search path is wrong, the shell will not be able to find the programs you try to run.

```
set path = ( /dist /usr/bin /bin /etc )
```

7. Determine which entries are on the tape, and which of them you want to install. To see the names of the entries on tape, *toc* use the *cat* command to examine the table of contents file:

```
cat /dist/toc
```

To find out which each entry name means, examine the description file *desc*:

```
cat /dist/desc
```

Most of the distribution tools accept a *names* argument that determines which of the entries listed in *toc* will be processed.

To process all of the entries listed in *toc*, omit the *names* argument. To process a subset of the available entries, list the entry names as the *names* argument, in the same order as they are given in the *toc* file.

8. Unless the system is new, or has no user files on it, check the disk space to ensure that the new entries will fit on the disk. To do this, use the *Spchk* and *df* commands:

```
Spchk names
df
```

The *Spchk* command reports the projected change in disk usage for each filesystem (*/* and */usr*) as plus or minus some number of disk blocks. If the number of disk blocks reported by *Spchk* is positive, compare the increase with the number of available blocks reported by the *df* command. Remember that the increase projected by *Spchk* is approximate, so be generous — allow a few hundred extra blocks if possible. If it looks close, or if there are not enough free blocks, back up some user files to tape and delete them from the disk.

When you have installed the software, you can reorganize the disk to make room for restoring the files that you deleted.

Remember that the disk is divided into *root* and *usr* file systems. If the space problem is on */usr*, for example, only deleting files from */usr* will help.

9. Read in the entries from the tape:

```
Read names
```

If you would like feedback while the command progresses, give the `-v` (verbose) option:

```
Read -v names
```

The `v` option of the *Read* command shows the names of the files as they are extracted from the entries on the tape. The disadvantage of the verbose option is that error messages can scroll off the screen unnoticed.

10. Perform the installation details for each entry:

```
Install names
```

This command indicates its progress and the names of the affected files. As the *Install* command runs, it reports the names of updated configuration files. The old configuration files are saved under a different name, i.e., with a trailing equal sign (=), and are replaced by the new configuration files. On all systems, changes will be made to these files:

```
/etc/bcheckrc  
/etc/brc  
/etc/inittab  
/etc/termcap
```

If you have the TCP/IP option, changes will be made to these files:

```
/etc/hosts  
/etc/hosts.equiv  
/etc/rc.tcp
```

If you have the IEEE 488 option, changes will be made to these files:

```
/etc/rc.488
/etc/libtab
```

11. After the *Install* command is finished, use the *diff* command to compare old files to the new ones. For example, to compare */etc/bcheckrc=* to */etc/bcheckrc*, issue these commands:

```
cd /etc
diff bcheckrc bcheckrc=
```

Make site-specific changes to the new files. Do not substitute any old files for the new ones; the new files are compatible with the update.

12. While you are running the *Install* command, you may receive a message indicating that the you need to run the *Install* command again after the system is rebooted. (See step 16, below.)
13. Verify that certain critical files (listed in Table 4-4 of the *IRIS Series 3000 Owner's Guide, Version 1.0*) are present before you reboot the system. *Verify* performs an incomplete check, but does serve to catch some of the more significant problems. If any of these problems exist, you must be correct them before the system is rebooted. Failure to do so may make it impossible to bring the system back up again. To verify that the critical files are present, type:

```
Verify
```

If you need assistance correcting problems that *Verify* reports, call the Geometry Hotline.

14. If any communications options were installed, reboot the system:

```
reboot
b
multi
```

15. Log in as root.
16. If you were instructed to do so (see step 12), perform the installation cleanup details:

```
/dist/Install -cleanup
```

17. Delete the distribution directory:

```
rm -rf /dist
```

18. If you deleted user files to make room on the disk, restore them. See *tar(1)* or *cpio(1)* for information on how to restore the files.

3.2 Network Installation

If you are installing the distribution on a workstation that does not have a tape drive, use a remote workstation with a tape drive and the XNS network protocol.

Follow the procedure above, but with these changes:

1. Log in as root on the remote machine. Follow steps 2 - 4 as above. Using the commands in step 5 above, read the distribution tools onto the remote machine, using *cpio*.

Log in as root on the machine on which you are installing the software. Copy the distribution tools to the machine on which you are installing the software using these commands (*remote* is the name of the machine with the tape drive):

```
cd /  
xcp -r -v remote:/dist .
```

2. Follow steps 6, 7, and 8 as above. Instead of step 9, issue the *Read* command with the *-x* option. This option tells the *Read* command the name of the remote machine.

```
Read -x remote names
```

As in step 9, the *names* argument is optional. Continue following the procedure above.

4. Additions

The next three sections list additions to graphics, systems, and options software since the GL2-W3.3.1 release.

4.1 Graphics Additions

This section lists graphics features added since the GL2-W3.3.1 release.

Item	Description
<code>blanktime()</code>	A new function <code>blanktime(nframes)</code> sets the duration before the screen is blanked. If <i>nframes</i> is 0, the screen never unblanks.
<code>getothermonitor()</code>	The new function <code>getothermonitor()</code> returns the display mode available and not currently used.
<code>ismex()</code>	The <code>ismex()</code> routine returns TRUE if the window manager, <i>mex</i> , is running.
<code>isqueued()</code>	A new function <code>isqueued(device)</code> returns TRUE if the specified device is queued.
<code>loadfont</code>	The new program <code>/usr/bin/loadfont</code> allows the default font to be set on system start-up.
<code>pop-up menu</code>	A simple pop-up menu package allows user processes to use the same pop-up bit planes as the window manager. The following routines were added to the Graphics Library:
<code>addtopup()</code>	adds items to an existing pop-up menu.
<code>defpup()</code>	defines a pop-up matrix.
<code>dopup()</code>	calls up the indicated pop-up menu.
<code>endfullscrn()</code>	ends full screen mode, returning the screen mask and viewport to the bounds of the current graphics port.
<code>endpupmode()</code>	ends pop-up mode.
<code>fullscrn()</code>	enables the entire screen for writing under the window manager.
<code>freepup()</code>	returns a menu and its structures to the system.

<code>newpup()</code>	allocates and initializes a structure for a new menu.
<code>pupcolor()</code>	specifies the current pop-up drawing color.
<code>pupmode()</code>	provides access to the pop-up menu bit planes.
<code>winattach()</code>	attaches the input focus to the current graphics port and calling process.
<code>winclose()</code>	closes the identified graphics port.
<code>winconstraints()</code>	changes the constraints associated with the current graphics port.
<code>winmove()</code>	moves the lower left corner of the current graphics port.
<code>winopen()</code>	creates a graphics port under the window manager.
<code>winpop()</code>	moves the current graphics port in front of all other ports.
<code>winpush()</code>	moves the current graphics port in back of all other ports.
<code>winset()</code>	makes the identified port current.
<code>wintitle()</code>	adds a title bar to the current graphics port.
<code>rcrv()</code> , <code>rcrvn()</code> , <code>rpatch()</code>	New rational cubic routines are <code>rcrv()</code> , <code>rcrvn()</code> , and <code>rpatch()</code> . These are rational versions of the <code>crv()</code> , <code>crvn()</code> , and <code>patch()</code> commands.
<code>rot()</code>	The new <code>rot()</code> command takes a floating point angle argument.
<code>setfastcom()</code> , <code>setslowcom()</code>	<code>setfastcom()</code> and <code>setslowcom()</code> were added to the workstation versions of the Graphics Library, but have no actions. The addition of these routines allows remote programs to be recompiled to run locally without change.

<i>/usr/include/device.h</i>	A set of timer devices, TIMER0, TIMER1, TIMER2, and TIMER3, were added to the file <i>/usr/include/device.h</i> . These can be queued and read like valuator. Setting noise on a queued timer specifies how often the timer should add events to the input queue.
	A <code>MENUBUTTON</code> device, added to <i>/usr/include/device.h</i> file, is defined to be the same as the right mouse button. Applications that use pop-up menus should use <code>MENUBUTTON</code> .
<i>/usr/people/gifts</i>	Many window manager examples were added to <i>/usr/people/gifts</i> .

4.2 Systems Additions

This section lists additions to UNIX since the GL2-W3.3.1 release.

Item	Description
<i>/etc/inittab</i>	A <i>sysint</i> state has been added to <i>/etc/inittab</i> . It executes a new script <i>/etc/rc.s0</i> that resets the virtual system console to the actual physical system console.
<i>System V shared memory</i>	System V shared memory is now implemented.

4.3 Additions to Options

This section lists additions to optional peripherals, communications options, and FORTRAN and Pascal tools since the GL2-W3.3.1 release.

Item	Description
<i>light pen</i>	A light pen option was added. A device called <code>LPENVALID</code> that acts like a button was created to indicate when the light pen position is valid.
<i>Pascal</i>	A Pascal local Graphics Library is now implemented.
<i>TCP/IP</i>	A new TCP/IP utility, <i>route</i> , provides host system access to the on-board routing table. This permits the system administrator to install routes, through gateways on the Ethernet, to systems on other networks.

A new TCP/IP utility, *arp*, provides host system access to the on-board ARP table. The system administrator can examine the table and can add or delete entries.

A new TCP/IP daemon, *ud*, extends *uucp* services (including mail) across the Internet protocols.

The 4.2BSD standard TCP/IP *rwho* and *ruptime* utilities and their corresponding daemon, *rwhod*, were added.

5. Changes

The next three sections list changes to graphics, systems, and options software since the GL2-W3.3.1 release.

5.1 Graphics Changes

This section lists graphic changes since the GL2-W3.3.1 release.

Item	Description
<code>chunksize()</code>	<p>The error message:</p> <p style="text-align: center;"><code>ERR_SIZEFIXED</code></p> <p>was reported if <code>chunksize()</code> was called after the chunksize was frozen. Now</p> <p style="text-align: center;"><code>CHUNKSIZE IS FROZEN</code></p> <p>is reported.</p>
<code>depthcue()</code>	<p>Polygons and characters, as well as lines, are now depth-cued.</p>
<i>European video standard</i>	<p><code>/usr/include/get.h</code>, <code>setmonitor()</code>, and <code>getmonitor()</code> now handle the PAL display option.</p>
<i>graphical objects</i>	<p>The method of freeing graphical objects has been improved. You are now less likely to run out of memory during long graphics sessions.</p>
<code>lint(1)</code>	<p><code>getdev()</code> and <code>devport()</code> have been added to the Graphics Library <i>lint</i> library.</p>

mex(1G)

When a window is created or moved, the new region is no longer cleared to black. Windows can now slide off the screen. Window size and shape can be controlled from a process.

The cursor shape now belongs to the process that currently has the input focus.

Multiple windows per process are now supported.

When *mex* puts itself in the background, it now exits with a return code of 0 rather than -1.

`imakebackground()` is now a window manager hint, just like `winaspect()` or `prefposition()`. `imakebackground()` must be followed by a `winopen()` command.

Events are now distributed to applications and the window manager in a new way. The current convention makes the right mouse button the menu button. All input events (from other buttons and valuator) are delivered to the window that currently has the input focus, with the exception of the menu button. Pressing the menu button has two effects:

- If the cursor is over the window that currently has the input focus and this window has the right button queued, a button-down event is be handed to this window. This is used by applications that provide pop-up menus.
- Otherwise, the button-down event goes to the window manager.

If the cursor is over the background and the menu button is pressed, a short menu appears. If the cursor is over another window, a longer menu allows that window to be moved, pushed or popped, etc. Pressing the menu button in the header of any window at any time brings up a menu that allows that window to be moved, pushed, popped, etc.

To access the pop-up menus provided by applications, you must not have a RESERVEBUT entry for the right mouse button in your *.mexrc* file. A sample *.mexrc* file is included in */usr/lib/gl2/mexrc*.

NOTE: If titles do not appear correctly on your windows while using *mex*, examine your *.mexrc* file. Look at the bind index lines. The color index for the background of the title bar is specified by *inborder* and *hiinborder*, while *titletextout* and *hititletextout* specify the color index for the horizontal lines drawn in the title bar. *titletextin* and *hititletextin* are used for the text in the title. The names starting with "hi" are used when the input focus is attached to a window.

objects and curves

Object code and curve code are now loaded only if objects and curves are used. This makes many graphics programs smaller.

screen unblanking

The screen unblanks whenever any valuator moves or any button changes state. Before, only the keyboard and the mouse caused the screen to unblank.

5.2 Systems Changes

This section lists changes to UNIX since the GL2-W3.3.1 release.

Item

Description

alternate keypad mode

You now enter alternate mode with `ESC ;` or `ESC =`. The key codes transmitted are now VT52 standard.

Key	Alternate keypad code
0	ESC ?p
1	ESC ?q
2	ESC ?r
3	ESC ?s
4	ESC ?t
5	ESC ?u
6	ESC ?v
7	ESC ?w
8	ESC ?x
9	ESC ?y
- (dash)	ESC ?m
, (comma)	ESC ?l
. (period)	ESC ?n
ENTER	ESC ?M
PF1	ESC P
PF2	ESC Q
PF3	ESC R
PF4	ESC S

bcopy(), bcmp(), bzero()

The routines *bcopy()*, *bcmp()*, and *bzero()* were removed from */usr/lib/libxns.a*, because they are in */lib/libc.a*.

ci(1)

The length of the log message in RCS files was increased from 512 to 4096 characters.

/etc/passwd

The user *iris* was removed.

error messages

The system now prints certain user-specific error messages on the user's console as well as on the standard console window.

floating point division

Software floating point division has been made more accurate.

mdfex, ipfex(1M)

The file system checking program now reports the device name before it checks each disk.

<i>make</i> (1)	The command <i>make -p</i> now sends all of its output to <i>stdout</i> , instead of sending some to <i>stdout</i> and some to <i>stderr</i> .
<i>reboot</i> (2)	<i>reboot</i> asks for verification if it is being executed from a network tty.
<i>reset</i> (1)	This command now defines the terminal interface as in <i>/usr/include/termio.h</i> . For example, the EOF character is now CTRL-D instead of CTRL-A .
<i>/usr/include/gl2/fastfeed.h</i>	This file has been removed.
<i>/usr/include/termio.h</i>	This file now includes a B19200 define for 19.2K baud.

5.3 Changes to Options

This section lists changes to FORTRAN and Pascal tools and to communications options since the GL2-W3.3.1 release.

Item	Description
<i>FORTRAN</i>	<i>/usr/include/fgl.h</i> now contains a NAMEST parameter that tells the depth of the name stack. <i>dbg</i> , the debugger for FORTRAN, was removed from the release and replaced by <i>dbx</i> . <i>f77</i> now includes a \$SAVE option to add a SAVE statement at the beginning of each routine.
<i>IBM</i>	<i>mT3279</i> was removed, since its functions were incorporated into <i>t3279</i> .
<i>Pascal</i>	Floating point exceptions are now trapped. You now have more control over floating-point exceptions in Pascal programs. <i>dbg</i> , the debugger for Pascal, was removed from the release and replaced by <i>dbx</i> .
<i>TCP/IP</i>	Routing is now supported. The TCP/IP protocol module dynamically adapts routing policy according to ICMP redirect messages. The ARP implementation can now advertise the physical addresses of third-party systems.

This facilitates communication with systems that do not support the ARP protocol.

telnet and *rlogin* daemons are implemented on-board via the *xtty(7)* driver. This obsoletes the old server programs *rlogind* and *telnetd*.

TCP/IP special files */dev/ttyT?* replace */dev/pty?* and */dev/typ?*. */etc/inittab* and */etc/ttytype* have been updated to reflect this name change.

The TCP/IP *netload* utility provides new protocol module configuration options. In particular, the local host is selected by the alias "localhost" in */etc/hosts*.

The TCP/IP header files in */usr/include/EXOS* have new names:

Old Name	New Name
<i>net/if_uba.h</i>	(no longer exists)
<i>net/in.h</i>	<i>netinet/in.h</i>
<i>net/in_pcb.h</i>	<i>netinet/in_pcb.h</i>
<i>net/in_sysm.h</i>	<i>netinet/in_sysm.h</i>
<i>net/ip.h</i>	<i>netinet/ip.h</i>
<i>net/ip_icmp.h</i>	<i>netinet/ipicmp.h</i>
<i>net/ip_var.h</i>	<i>netinet/ip_var.h</i>
<i>net/misc.h</i>	<i>exos/misc.h</i>
<i>net/netdb.h</i>	<i>netdb.h</i>
<i>net/pup.h</i>	<i>exos/pup.h</i>
<i>net/tcp.h</i>	<i>netinet/tcp.h</i>
<i>net/tcp_debug.h</i>	<i>netinet/tcp_debug.h</i>
<i>net/tcp_fsm.h</i>	<i>netinet/tcp_fsm.h</i>
<i>net/tcp_seq.h</i>	<i>netinet/tcp_seq.h</i>
<i>net/tcp_timer.h</i>	<i>netinet/tcp_timer.h</i>
<i>net/tcp_var.h</i>	<i>netinet/tcp_var.h</i>
<i>net/tcpip.h</i>	<i>netinet/tcpip.h</i>
<i>net/udp.h</i>	<i>netinet/udp.h</i>
<i>net/udp_var.h</i>	<i>netinet/udp_var.h</i>

6. Bug Fixes

The next three sections list bugs fixed in graphics, systems, and options software since the GL2-W3.3.1 release.

6.1 Graphics Bug Fixes

This section lists graphic bugs fixed since the GL2-W3.3.1 release.

Bug	Description
<code>backface()</code>	The <code>backface()</code> command now correctly handles rectangles made with <code>rectf()</code> .
<code>crv()</code>	<code>crv()</code> no longer causes extraneous lines to be drawn.
<code>crvn()</code>	<code>crvn()</code> in a picked object no longer generates the error message: <div style="text-align: center;"><code>No feedback in pickmode in getmatrix</code></div>
<code>devport()</code>	This command now works remotely and has been added to the Graphics Library <i>lint</i> library.
<i>feedback mode</i>	<code>readpixels()</code> , <code>getmatrix()</code> , <code>pushmatrix()</code> , and <code>loadmatrix()</code> now work in feedback mode.
<i>font RAM</i>	The code that loads RAM no longer wraps around and writes over other programs' font RAM. The font RAM no longer causes horizontal lines.
<i>FORTTRAN</i>	The <i>FORTTRAN</i> remote Graphics Library version of the <code>winopen()</code> command now returns a value. Depth cuing in <i>FORTTRAN</i> now works. The <i>FORTTRAN</i> version of <code>shaderange()</code> now has the correct number of parameters.
<code>getdev()</code>	The <code>getdev()</code> command now works remotely.
<code>getport()</code>	If you use the <i>wsiris</i> terminal emulator, you no longer need to call <code>foreground()</code> before <code>getport()</code> to prevent a problem with the keyboard.

<code>gexit()</code>	Keypad mode is now restored with <code>gexit()</code> , if the keypad mode is turned on and a graphics program was run on a terminal other than the console.
<i>keyboard buttons</i>	Both keyboard buttons and the keyboard are queued on exit, so that even on an exception or interruption, the keyboard is usable.
<code>linewidth()</code>	A call of <code>linewidth()</code> without an argument does not generate wide lines.
<code>mex(1G)</code>	<p>A call to <code>setcursor()</code> to change the cursor glyph no longer causes the cursor to disappear and not reappear until <code>mex</code> sets it. <code>setcursor()</code> now ignores its color and writemask arguments when <code>mex</code> is running.</p> <p>Running the <code>gclear(1)</code> command after exiting <code>mex</code> now works correctly.</p> <p>Pop-up menus near the edge of the screen now work correctly.</p> <p><code>mex</code> now handles mouse buttons correctly when the <code>NOSCROLL</code> key is used in hogwhiledown mode.</p> <p>All keyboard input is now sent to the attached process.</p> <p><code>mex</code> now maintains the requested aspect ratio when the mouse runs into the edge of the screen.</p> <p>Multiple windows in a double-buffered <code>mex</code> program no longer confuse the display mode.</p> <p><code>mex</code> no longer dumps core if <code>keepaspect()</code> is called with no arguments.</p>
<i>microcode</i>	Stack overflows no longer occur when character moves are made outside of a bounding box, causing the character position to be clipped. After nine such clipped positions occurred, some subsequent microcode commands, such as polygons, did not work.

<code>objreplace()</code>	Incorrectly freed memory no longer causes a problem with <code>objreplace()</code> related to the <code>crv()</code> and <code>patch()</code> commands.
<i>pick mode</i>	Entering depth cuing while in pick mode now works.
<code>popattributes()</code>	The current font is now restored by <code>popattributes()</code> .
<code>ps(1)</code>	Network tty names are no longer truncated to two characters in <i>ps</i> output.
<code>rectcopy()</code>	Several <code>rectcopy()</code> bugs are fixed. These bugs had appeared when you called this command in the window manager or used it with some copy directions. The screen coordinates are now relative to the physical screen, instead of to the lower left corner of the graphics port.
<code>shaderange()</code>	The <i>lint</i> library for <i>libgl.a</i> now contains the correct four parameters for <code>shaderange()</code> . The C and FORTRAN parameter type declarations are now correct.
<code>textport()</code>	The <code>textport</code> now handles tabs that wrap past the end of a line. The call <code>textport(0,0,0,0)</code> no longer causes a kernel bus error panic after a few keystrokes.
<i>wsiris</i>	<code>%M</code> now echoes the correct monitor type. <i>wsiris</i> now recognizes <code>gbegin()</code> , <code>winopen()</code> , and <code>getport()</code> as graphics initialization routines.
<i>zbuffer1.c</i>	This program in <i>/usr/people/gifts/examples</i> no longer dumps core.

6.2 Systems Bug Fixes

This section lists bug fixes to UNIX, the C compiler, XNS, and the disk and tape drives since the GL2-W3.3.1 release.

Bug	Description
<i>ASCII terminals</i>	Spurious error messages no longer appear when an ASCII terminal is connected to an IRIS workstation.
<i>block mode serial I/O</i>	Block mode serial I/O is included in the kernel.

Version 1.0

- C routines* Recursive C routines that are of class *fortran* can now be called from C.
- cp(1)* The command *cp -r* incorrectly changed the owner of all new files to root and did not preserve hard links within a tree.
- csh(1)* Changing the predefined variable environment *USER*, *LOGNAME*, *TERM*, or *HOME* now causes a corresponding change to the shell variables of the same names in lower case.
- The built-in *echo* command no longer causes *csh* to exit when given the command:
- ```
"echo 'echo" [file]""
```
- cxref(1)* *cxref* now processes any C file that includes */usr/include/stdio.h*.
- dbx(1)* The *print* command in *dbx* no longer prints the original value of a variable after the value has changed within a *while* loop.
- eqn(1)* *eqn* now correctly forms the "there exists" special character.
- /etc/termcap* The */etc/termcap* entries for start and end underline, for IRIS systems, are now correct.
- find(1)* *find* now interprets a question mark (?) correctly and no longer prints a period (.) for the current directory.
- kernel* The kernel now handles buffers of 250K passed through pipes.
- The kernel scheduler no longer charges idle time to the last running process.
- The system no longer panics if a *-l* is passed as a system address on a machine without an FPA board.
- The paging system no longer generates the error message:

```
dup mem alloc
```

Paging now works correctly in the following program:

```
system("date");
ginit();
gexit();
```

- lex*(1) *lex* no longer dumps core on short erroneous input.
- make*(1) *make* now expands the pseudo-variable *\$?* correctly.
- man*(1) Several minor bugs in the *man* command are fixed.
- mv*(5) *mv* now handles the command *mv xxx dir* properly when *xxx* is a file and a directory named *dir/xxx* already exists.
- permuted index* The permuted index generation macros */usr/lib/tmac/tmac.ptx* were missing.
- ps*(1) *ps* no longer produces the error message:
- ```
ps: error locating command name for pid %d from
```
- when it dumps core or cannot find user program arguments.
- rogue*(6) *rogue* now correctly checks the load average and exits.
- serial communications* Serial communications are now reliable.
- serial tty ports* Serial tty ports no longer cause machines to crash with the error message:
- ```
trap type 2
```
- because of incorrect handling of modem interrupts.
- strip*(1) *strip* now strips multiple archives.
- tar*(1) *tar* commands that used the *-f* option (used by *stdin*) no longer produce a segmentation violation.
- tbl*(1) *tbl* no longer dumps core.

|                     |                                                                                                                                                                                                                                      |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>test -x</i>      | If the super-user issues <i>test -x</i> , the system no longer returns TRUE when no execute permissions are set.                                                                                                                     |
| <i>trap</i>         | Catching signal 18 with <i>trap</i> in <i>sh</i> caused a core dump. Signal 18 can no longer be trapped.                                                                                                                             |
| <i>USIZE</i>        | As documented in <i>core(4)</i> , <i>USIZE</i> is now defined in <i>/usr/include/sys/param.h</i> .                                                                                                                                   |
| <i>/usr/lib/cpp</i> | The <b>-p</b> option of <i>/usr/lib/cpp</i> now handles nested <i>#includes</i> properly.                                                                                                                                            |
| <i>uucp(1C)</i>     | <i>uucp</i> now works with system names shorter than six characters.                                                                                                                                                                 |
| <i>XNS</i>          | If you interrupt a program that is writing to an XNS connection, it no longer logs you out of the program reading the connection.<br><br>XNS no longer hangs and gives the error message:<br><br><code>nx: no message buffer.</code> |
| <i>xlogin(1)</i>    | XON/XOFF no longer causes inappropriate delays when you log in remotely to a workstation.                                                                                                                                            |
| <i>xx</i>           | Background <i>cs</i> h jobs started over the net with the command:<br><br><code>xx host 'prog &amp;'</code><br><br>now function.                                                                                                     |

### 6.3 Options

This section lists bug fixes to peripheral options, communication options, and the FORTRAN and Pascal compilers since the GL2-W3.3.1 release.

| <b>Bug</b>      | <b>Description</b>                                                                                                                                    |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>f77 code</i> | The <i>f77</i> code generator no longer causes incorrect code to be generated when returning double-precision results of functions called indirectly. |

|                                  |                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>floating point</i>            | <p>Calls to the user's cleanup routine for floating point exceptions from FORTRAN are now performed.</p> <p>A FORTRAN compiler hardware floating-point bug is fixed.</p>                                                                                                                                                                                                    |
| <i>FORTRAN</i>                   | <p>FORTRAN now truncates sequential files correctly upon close, using the new system command <i>ftruncate</i>.</p>                                                                                                                                                                                                                                                          |
| <i>FORTRAN functions</i>         | <p>Several intrinsic FORTRAN functions are now defined:</p> <ul style="list-style-type: none"> <li>• <i>getarg()</i></li> <li>• <i>iargc()</i></li> <li>• <i>ran()</i></li> </ul>                                                                                                                                                                                           |
| <i>FORTRAN graphics commands</i> | <p>The interface library <i>libfgl.a</i> now contains entries for these FORTRAN graphics commands:</p> <ul style="list-style-type: none"> <li>• <i>gflush()</i></li> <li>• <i>ginit()</i></li> <li>• <i>greset()</i></li> <li>• <i>gexit()</i></li> <li>• <i>shaderange()</i></li> </ul>                                                                                    |
| <i>IEEE 488</i>                  | <p>The IEEE 488 driver now handles odd-length writes correctly.</p> <p>When you use the IEEE 488 interface, sequences of short (one- or two-byte) transfers no longer fail randomly.</p> <p><i>wsiris</i> now works with IEEE 488.</p> <p>When IEEE 488 is used for graphics and serial line for text, all of the child processes are killed after <i>wsiris</i> exits.</p> |
| <i>mkf2c(1), mkc2f</i>           | <p><i>mkf2c</i> and <i>mkc2f</i> no longer produce a core dump on input from <i>stdin</i>.</p>                                                                                                                                                                                                                                                                              |

*TCP/IP*

Using TCP/IP with multiple connections no longer causes the system to hang.

TCP now follows TIME-WAIT specifications. In the past, TCP had not closed TCP connections promptly, which violated specifications across networks with a long round-trip time.

The TCP/IP *telnet* server has improved its processing of AYT. It returns a `CTRL-G` directly to the user and does not emulate a `CTRL-G` in the user's input stream.

Mismatches between advertised and actual header/data lengths in incoming packets are now properly accommodated.

TCP/IP *select()* works on UDP sockets.

The TCP/IP *telnet* utility sends a `< cr >` `< lf >` sequence, as recommended by the TELNET specification. Previously, it sent only a `< cr >`.

*wsiris*

*wsiris* no longer hangs if a bad host IP address is entered. IP addresses are now checked to verify that they contain three periods (.).

## 7. Known Problems

This list contains known problems with this release, and, where possible, suggested workarounds.

*check(1)*

*check -1 [rcsdir]* does not work.

*ci(1)*

*ci* does not let you interrupt a check-in of more than one file.

*dbx*

*dbx* garbles output when debugging from a terminal connected to a serial port. This problem does not occur, however, if the serial terminal is connected through *xlogin* to a workstation running *dbx*.

*depthcue()*

A small number of pixels just outside a shaded range of colors may appear in an image where lines are clipped by the far clipping plane. A work-around is to extend the color range to include the overflow bits, or to reduce the values passed to shaderange. For example, if

indices 128-191 are used for a shaded range of colors and pixels appear outside that range, call

```
shaderange(135,185,...)
```

or extend the color range to 125-196 and call

```
shaderange(128,191,...)
```

*/dev/rmt2*

Interrupting */dev/rmt2* (the no-rewind cartridge tape device) on the IRIS 2400T and 3020 makes it necessary to reboot the system to use the tape again. This problem does not apply to the IRIS 3030.

*DSD controller*

Swapping the DSD controller for a Storer on a 2400T or a 3020 and adding an ESDI disk causes device name linking problems. This happens because */etc/model* is used to determine the controller type, but there is no longer a correlation between the model number and the controller type.

*fast immediate mode macros*

When using the fast immediate mode macros, there are times when the file */usr/include/gl2/glerror.h* should also be included. To be safe, add the line:

```
#include "gl2/glerror.h"
```

when you use the fast immediate mode macros.

*font RAM*

When large fonts are repeatedly defined and deleted, the space in the font RAM is sometimes not cleaned up correctly, and a font cannot be defined correctly, even though there ought to be enough space to hold it.

*getmem()*

*getmem()* sometimes returns the wrong value.

*lines*

Wide lines do not work correctly in depth cue mode.

- mex*(1G) The `textcolor()` and `pagecolor()` commands do not work correctly under *mex*. Sometimes the color of the wrong textport is changed, and sometimes the commands appear to have no effect.
- Pascal Graphics Library* The Graphics Library commands `readRGB()`, `writeRGB()`, `capture()`, and `rcapture()` do not work in the Pascal version of the Graphics Library. A Pascal program cannot be used to generate pictures on the color printer.
- phys*(2) This system call is not supported in this release. For a work-around, contact the Geometry Hotline.
- pick*() Picking sometimes fails when objects are nested to a depth of more than six or seven.
- sqrt*(3M) This program prints a very large number when it should print zero.
- ```
extern float sqrt();
main()
{
    long float dddl = 1.0;
    printf("%f0.sqrt(float)((float)(dddl-dddl));
}
```
- TCP/IP* An incoming *rsh* may hang if at the same time another *rsh* in either direction is starting or ending.
- zbuffer*() A small number of pixels just outside a shaded range of colors may appear in an image. Use the same work-around described for `depthcue()`, above.