

GL2-W3.6 Workstation Release Notes

Version 1.0

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

Document Number 007-3206-010

Technical Publications:

Kathleen Chaix
Diane Wilford
Anne Wilson

Engineering:

Greg Boyd
Mark Bradley
Dave Ciemiewicz
Joseph Fota
Bob Green
Kevin Gurney
Lynne McDonald
Liz Mehta

© Copyright 1987, Silicon Graphics, Inc - All rights reserved

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

U.S. Government Limited Rights

Use, duplication or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (b) (2) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. Contractor/manufacturer is Silicon Graphics Inc., 2011 Stierlin Road, Mountain View, CA 94039-7311.

GL2-W3.6 Workstation Release Notes

Document Number: 007-3206-010

The words IRIS, Geometry Link, Geometry Partners, Geometry Engine and Geometry Accelerator are trademarks of Silicon Graphics, Inc.

NFS is a trademark of Sun Microsystems.

UNIX is a trademark of AT&T Bell Laboratories.

XNS is a trademark of XEROX Corporation.

CONTENTS

1. Introduction	1-1
2. Major Enhancements	2-1
2.1 Overview	2-1
2.2 Descriptions	2-1
2.2.1 <i>edge</i>	2-1
2.2.2 Netboot Process	2-2
2.2.3 IBM 3270 Coax Link Option	2-2
2.2.4 380Mb ESDI Disk Drive Option	2-2
2.2.5 Intelligent 8-Port Serial I/O Controller Option	2-3
2.2.6 SMD Disk Support	2-3
2.2.7 German Keyboard Support	2-3
3. Installation Procedures	3-1
3.1 Understanding the Installation Process	3-1
3.1.1 How the Distribution Tape is Organized	3-2
3.2 Installing from Tape	3-2
3.3 Installing Over the Network	3-7
3.4 Compiling Your IRIS Programming Tutorial	3-12
3.4.1 Compiling the Tutorial in C	3-13
3.4.2 Compiling the Tutorial in FORTRAN	3-14
4. Additions and Changes	4-1
4.1 Additions	4-1
4.2 Changes	4-3
5. Fixed Bugs	5-1
6. Known Problems	6-1
7. Documentation	7-1

1. Introduction

This set of release notes describes Software Release GL2-W3.6 for these IRIS workstations: IRIS Series 3000, including all models in the 3000 and 3100 product lines; and the series 2000 Turbo products (2400T and 2500T).

The GL2-W3.6 software release is compatible with the GL2-W3.6 version of these options: FORTRAN, Pascal, NFS, XNS, and Laser Printer. It is also compatible with the GL2-W3.5r2 version of EMACS, and the GL2-W3.5r3 version of Ada.

These release notes are divided into the following six sections:

- Major Enhancements
- Installation Procedures
- Additions and Changes
- Fixed Bugs
- Known Problems
- Documentation

Silicon Graphics, Inc. provides comprehensive product support and maintenance for IRIS products. For further information, contact Customer Service through the toll-free Geometry Hotline:

Silicon Graphics Geometry Hotline	
(800) 345-0222	U.S. and Canada (toll-free)
350613	Worldwide (telex number)

When you report problems through the Geometry Hotline, please have the serial number of the Silicon Graphics, Inc. machine available. It is suggested that you list the serial numbers of the machines at your site in the */etc/hosts* file.

2. Major Enhancements

2.1 Overview

These enhancements are included in the GL2-W3.6 release:

- *edge* graphical interface to the *dbx* standard UNIX debugger
- Netboot process for system crash recovery

These options are supported in the GL2-W3.6 release:

- IBM 3270 Coax Link
- 380Mb ESDI disk drive
- Intelligent 8-Port Serial I/O Controller

Support for these other products is included in the GL2-W3.6 release:

- SMD disk
- German keyboard

2.2 Descriptions

2.2.1 *edge*

edge, a window-based graphical interface to *dbx*, a standard UNIX debugger, is supported as part of the GL2-W3.6 release. The *edge* debugging environment consists of three windows that display the *dbx* prompt, the source code you are debugging, and the standard input/output results you see when you run the program.

The *edge* menu-oriented debugging environment lets you issue *dbx* commands through both the keyboard and the mouse.

Learning to Debug with edge (C or FORTRAN edition) explains the use of *edge*.

2.2.2 Netboot Process

With the netboot process, a workstation on a network can be rebooted after a software system crash. The process uses a bootable backup tape on another workstation on the network. Instructions for using the netboot process are listed in the *IRIS Series 3000 Owner's Guide*.

2.2.3 IBM 3270 Coax Link Option

The IBM 3270 Coax Link supports file transfer and terminal emulation between the IRIS and an IBM mainframe, running either the MVS/TSO or VM/CMS operating systems. The IBM 3270 Coax Link has three menus. The 3270 Emulation Menu displays commands that transfer files and cause the IRIS to emulate an IBM 3278 or 3279 Model 2 terminal. There is also a Setup and Diagnostics Menu, and an Advanced Diagnostics Menu.

To use this option, your IRIS workstation must be running the IBM 3270 Coax Link software, and have the 3270 interface card installed. The IBM mainframe must be running IBM host 3270 software from Silicon Graphics, Inc.

The *IBM 3270 Communications Guide* is included with this option.

2.2.4 380Mb ESDI Disk Drive Option

The 380Mb ESDI (Enhanced Small Device Interface) disk drive includes a 380Mb Winchester disk drive. This option is compatible with IRIS workstations using the standard ESDI controller.

Installing and Configuring ESDI and QIC-02 Modules is included with this option.

2.2.5 Intelligent 8-Port Serial I/O Controller Option

The standard IRIS Series 3000 workstation contains four serial ports. This option is an intelligent controller board used to connect up to eight additional RS-232 devices (such as ASCII terminals or peripherals) to an IRIS Series 3000 workstation. The option is available for IRIS 2500T, 3115, 3120, 3120B, and 3130 workstations.

2.2.6 SMD Disk Support

Two SMD (Storage Module Device) disk drives are supported: the Control Data Corporation Model 9766 disk drive, and the Fujitsu Model 2351A disk drive.

2.2.7 German Keyboard Support

This release supports the use of German keyboards with the IRIS Series 2000 turbo products and the IRIS Series 3000 workstations. Data is displayed on the screen in the correct font for the German keyboard. (If a keyboard is not recognized, the system defaults to U.S. keyboard font.) The system recognizes and implements PROM commands from both U.S. and German keyboards.

There are some general restrictions to the use of international keyboards. The use of 8-bit characters may cause incorrect translation of these characters into 7-bit characters, through *sh* or *csk*. As a result, programs that have not been modified to expect 8-bit characters may fail. Data files such as */etc/passwd* may not contain 8-bit data. Application programs that read key state transitions directly from keyboard input must be re-compiled using the proper *device.h* header file entry.

3. Installation Procedures

Follow the procedures listed in this chapter to install the GL2-W3.6 release on your IRIS workstation.

GL2-W3.6 should not be installed on a machine running a release older than GL2-W3.5. If the machine to be updated is running an older release, update it to GL2-W3.5 (or a revision) before updating to GL2-W3.6.

Make sure you are using the standard system distribution tapes included with these release notes. (Before installing your software, you must back up your user files onto tape. See Chapter 4, *Workstation Administration*, in the *IRIS Series 3000 Owner's Guide*.)

This chapter includes three sections.

- Understanding the Installation Process, Section 3.1, describes how the software options and updates are organized and provides an overview of the installation process.
- Installing from Tape, Section 3.2, provides step-by-step procedures for installing the standard system update and the options from a local tape drive.
- Installing Over the Network, Section 3.3, describes how to install the update and options on a workstation without a tape drive.

3.1 Understanding the Installation Process

You receive software options and updates from Silicon Graphics, Inc., on one or more quarter-inch *distribution tapes*. The standard system update tape contains all the UNIX utilities, the new TCP/IP kernel, the on-line manual pages, games, demonstration programs, the programming tutorial, new electronic mail, and gifts.

Any options, such as XNS, NFS, or FORTRAN, are on separate tapes.

To use this software you must first transfer it from the tape to your disk, and then install it in the correct locations in the file system.

3.1.1 How the Distribution Tape is Organized

A distribution tape contains a set of distribution tools used to install the software, and one or more *images*, or collections of the files to be installed. Each of these images constitutes a separate *physical* file.

The standard system update tape, for example, contains several physical files. The first is the distribution tool directory, */dist*, followed by separate images for the system itself, the on-line manual pages, the graphics demonstrations, and so on.

The general procedure for installing the software is to install the distribution tools from the tape, and then use the distribution tools to position and read the rest of the software from the tape and perform various installation tasks.

3.2 Installing from Tape

The installation process is done in two phases. In the first phase you will install the standard system update; in the second phase you will install the options you purchased and install the programming tutorial.

To install the standard system update and the options from a local tape, follow these steps:

1. Log in as *root*; make sure no one else is logged in, and then reboot your workstation. Respond to the prompts as shown.

```
who
reboot
b
multi
Is the date correct? y
Do you want to check file system consistency? y
root
```

2. If there is an old copy of the distribution directory present, remove it with this command:

```
rm -rf dist
```

3. Put the distribution tape (beginning with the standard system update tape) in the tape drive and rewind and retension it with these two commands:

```
mt rewind  
mt retension
```

Wait a few minutes for the tape to finish rewinding.

4. When the tape drive light goes off, read in the distribution tools with the *cpio* command:

```
cpio -ivhmod1
```

5. To see the names of the images on tape, use the *cat* command to examine the table of contents file *toc*:

```
cat /dist/toc
```

To find out what each image name means, examine the description file *desc*:

```
cat /dist/desc
```

Most of the distribution tools accept a list of image names as arguments that determine which of the images listed in *toc* will be processed. If you are concerned about using disk space for images that you do not really need (such as the *games* image), you should install only the images that you want by specifying them as arguments, in the same relative order as they are listed in *toc*.

To process all of the images listed in *toc*, omit the list of names.

6. Check the disk space to ensure that the new software will fit on the disk. To do this, use the *Spchk* and *df* commands.

To check the projected change in disk space use this command:

```
/dist/Spchk
```

To check the disk space for a subset of images use this command:

```
/dist/Spchk name(s)
```

The *Spchk* command reports the projected change in disk usage for each file system (*/* and */usr*) as plus or minus some number of disk blocks.

If the number of disk blocks reported by *Spchk* is positive, compare this increase with the number of available blocks reported by the *df* command:

```
df
```

Remember that the increase projected by *Spchk* is approximate, so be generous — allow 300-500 extra blocks if possible. **Failure to provide enough disk space could leave your disk in an unrecoverable condition.**

If there are not enough free blocks on */usr*, back up some user files to tape and delete them from the disk. If there are not enough free blocks on root, you may want to delete all the files in */lost+found* or in */tmp*. If you are installing the standard system update, you may also remove all of the kernels in the */kernels* directory, and the two standalone tools that are not used for your disk. For example, if your root disk is */dev/si0a*, you can remove */stand/ipfex* and */stand/mdfex*. (If necessary, the standalone tools can be booted again from the distribution tape.)

When you have installed the software, you can reorganize the disk to make room for restoring the files that you deleted. (See Chapter 6, Disk Configuration, in the *IRIS Series 3000 Owner's Guide*.)

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.

7. Read in the images from the tape:

```
/dist/Read -v      or      /dist/Read -v name(s)
```

The verbose (**-v**) option gives you feedback while the command progresses. This procedure takes several minutes to complete.

8. Perform the installation:

```
/dist/Install      or      /dist/Install name(s)
```

This command indicates its progress and the names of the affected files.

9. Before you reboot the IRIS, make sure that some of the more critical files are present with the *Verify* command:

```
/dist/Verify
```

The *Verify* command gives you a message that all the critical files are in place and that you can reboot the IRIS. If you need assistance correcting problems that *Verify* reports, call the Geometry Hotline. The Geometry Hotline number is listed on page 1-1 of these release notes.

10. Reboot the system and log in as *root*. Respond to each prompt as shown:

```
reboot
b
multi
(if prompted, enter the model number of your IRIS)
Is the date correct? y
Do you want to check filesystem consistency? y
root
```

11. If you were instructed to do so perform the installation cleanup:

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

12. The message displayed on your screen will tell you which configuration files were installed.

The names of the inactive configuration files are marked with a trailing equal sign (=) or comma (,) where *filename=* is the old version, and *filename,* is the suggested version. Use the *diff* command to compare the old and new files, and customize your configuration files to best use the available information.

Only the new files are compatible with the update. You can, however, install the new suggested configuration files (those with a trailing comma) directly.

Editing the configuration files is particularly important in a network environment.

Once you have edited the configuration files appropriately, remove the saved copy (the files with a trailing = or ,).

13. If you were instructed to run */dist/Install* again, repeat steps 8 through 12.

14. Delete the distribution directory:

```
rm -rf /dist
```

15. Remove the distribution tape from the tape drive.

3.3 Installing Over the Network

This procedure shows you how to install the GL2-W3.6 update on a workstation that does not have a tape drive. You will use the tape drive on an IRIS workstation that your workstation can access across the network. TCP/IP is the standard communications kernel for this release.

If you are using TCP/IP protocol, install the standard system update first, then install the options. If you are using XNS protocol, you must install the standard system update and the XNS option before you reboot in steps 13 and 14.

In the following instructions, *remote* refers to the IRIS that has a tape drive, and *local* refers to the IRIS that does not have a tape drive.

1. On the *local* machine, log in as *root*; make sure no one else is logged in, and then reboot your workstation. Respond to the prompts as shown.

```
who
reboot
b
multi
Is the date correct? y
Do you want to check filesystem consistency? y
root
```

2. On the *remote* machine, put the tape in the tape drive and log in as *root*.
3. If there is an old copy of the distribution directory present, remove it with this command:

```
rm -rf dist
```

4. Put the distribution tape in the tape drive and rewind and retension it with these two commands:

```
mt rewind
mt retension
```

Wait a few minutes for the tape to finish rewinding.

5. When the tape drive light goes off, read in the distribution tools with the *cpio* command:

```
cpio -ivhmud1
```

6. Log off the *remote* machine.
7. If there is an old copy of the distribution directory present, remove it with this command:

```
rm -rf /dist
```

8. Copy the distribution directory from the remote machine:

For TCP/IP:

```
rsh remote: /dist /
```

For XNS:

```
xcp remote: /dist /
```

9. To see the names of the images on tape, use the *cat* command to examine the table of contents file *toc*:

```
cat /dist/toc
```

To find out what each image name means, examine the description file *desc*:

```
cat /dist/desc
```

Most of the distribution tools accept a list of image names as arguments that determine which of the images listed in *toc* will be processed. If you are concerned about using disk space for images that you do not really need (such as the *games* image), you should install only the images that you want by specifying them as arguments, in the same relative order as they are listed in *toc*.

To process all of the images listed in *toc*, omit the list of names.

10. Check the disk space to ensure that the new software will fit on the disk. To do this, use the *Spchk* and *df* commands.

To check the projected change in disk space use this command:

```
/dist/Spchk
```

To check the disk space for a subset of images use this command:

```
/dist/Spchk name(s)
```

The *Spchk* command reports the projected change in disk usage for each file system (*/* and */usr*) as plus or minus some number of disk blocks.

If the number of disk blocks reported by *Spchk* is positive, compare this increase with the number of available blocks reported by the *df* command:

```
df
```

Remember that the increase projected by *Spchk* is approximate, so be generous — allow 300-500 extra blocks if possible. **Failure to provide enough disk space could leave your disk in an unrecoverable condition.**

If there are not enough free blocks on */usr*, back up some user files to tape and delete them from the disk. If there are not enough free blocks on root, you may want to delete all the files in */lost+found* or in */tmp*. If you are installing the standard

system update, you may also remove all of the kernels in the */kernels* directory, and the two standalone tools that are not used for your disk. For example, if your root disk is */dev/si0a*, you can remove */stand/ipfex* and */stand/mdfex*. (If necessary, the standalone tools can be booted again from the distribution tape.)

When you have installed the software, you can reorganize the disk to make room for restoring the files that you deleted. (See Chapter 6, Disk Configuration, in the *IRIS Series 3000 Owner's Guide*.)

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.

11. Read in the images from the tape:

For TCP/IP:

```
/dist/Read -v -t remote
```

or

```
/dist/Read -v -t remote name(s)
```

For XNS:

```
/dist/Read -v -x remote
```

or

```
/dist/Read -v -x remote name(s)
```

The verbose (*-v*) option gives you feedback while the command progresses. This procedure takes several minutes to perform.

12. Perform the installation:

```
/dist/Install      or      /dist/Install name(s)
```

This command indicates its progress and the names of the affected files.

For XNS users, repeat steps 2-12 for the XNS option tape.

13. Before you reboot the IRIS, make sure that some of the more critical files are present with the *Verify* command:

```
/dist/Verify
```

The *Verify* command gives you a message that all the critical files are in place and that you can reboot the IRIS. If you need assistance correcting problems that *Verify* reports, call the Geometry Hotline. The Geometry Hotline number is listed on page 1-1 of these release notes.

14. Reboot the system and log in as *root*. Respond to each prompt as shown.

```
reboot
b
multi
(if prompted, enter the model number of your IRIS)
Is the date correct? y
Do you want to check filesystem consistency? y
root
```

15. If you were instructed to do so perform the installation cleanup:

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

16. The message displayed on your screen will tell you which configuration files were installed.

The names of the inactive configuration files are marked with a trailing equal sign (=) or comma (,) where *filename=* is the old version, and *filename,* is the suggested version. Use the *diff* command to compare the old and new files, and customize your configuration files to best use the available information. Only

the new files are compatible with the update. You can, however, install the new suggested configuration files (those with a trailing comma) directly.

Editing the configuration files is particularly important in a network environment.

Once you have edited the configuration files appropriately, remove the saved copy (the files with a trailing = or ,).

17. If you were instructed to run */dist/Install* again, repeat steps 12 through 16.

18. Delete the distribution directory:

```
rm -rf /dist
```

19. On the *remote* machine, log in as root and remove the distribution directory:

```
rm -rf /dist
```

20. Remove the distribution tape from the tape drive.

3.4 Compiling Your IRIS Programming Tutorial

You need to compile the software that accompanies the *IRIS Programming Tutorial* before you use it. The tutorial software uses 5Mb of disk space in */usr* after it is compiled. If you have a new IRIS, you do not need to add the special *tutor* account in the */etc/passwd* file. If you are updating the software on your IRIS, you must add a special account for the tutorial software.

3.4.1 Compiling the Tutorial in C

Follow these steps to add an account called *tutor* to the */etc/passwd* file and compile the software for the IRIS programming tutorial:

1. Log in as *root*.
2. At the end of the *passwd* file in the *etc* directory, add this line:

(Note: this is actually one line)

```
tutor::994:997:Tutorial User:
/usr/people/tutorial/c.graphics:/bin/csh
```

3. Log out, and then log in as *tutor*:

```
logout
tutor
```

4. You are now in the */usr/people/tutorial/c.graphics* directory. To set up the directory, type this command:

```
make clean
```

5. To compile and install the software, type this command:

```
make install
```

6. To commit the changes to the *.login* file, type this command:

```
source .login
```

The C software for the *IRIS Programming Tutorial* is now installed.

3.4.2 Compiling the Tutorial in FORTRAN

If you have the FORTRAN 77 option, the FORTRAN tutorial software resides in a new directory called */usr/people/tutorial/f.graphics*.

Follow these steps to add an account called *tutorf* to the */etc/passwd* file and compile the software for the IRIS programming tutorial:

1. Log in as *root*.
2. At the end of the *passwd* file in the *etc* directory, add this line:

(Note: this is actually one line)

```
tutorf::993:997:Tutorial User:
/usr/people/tutorial/f.graphics:/bin/csh
```

3. Log out, and then log in as *tutorf*:

```
logout
tutorf
```

4. You are now in the */usr/people/tutorial/f.graphics* directory. To set up the directory, type this command:

```
make clean
```

5. To compile and install the software, type this command:

```
make install
```

6. To commit the changes to the *.login* file, type this command:

```
source .login
```

The FORTRAN software for the *IRIS Programming Tutorial* is now installed.

4. Additions and Changes

Additions and changes made since the GL2-W3.5 release are listed in this section.

4.1 Additions

Category	Description
<i>cc</i> (1)	<p>The C compiler now supports a new class of warnings. These warnings are issued when verbose mode is set, using the -Zv option.</p> <p>The C compiler now supports a -w option that suppresses compiler warnings.</p>
<i>gethostident</i> (3)	<p><i>gethostident</i> prints the 64-byte value returned by the <i>gethostident</i> system call that is unique to the particular machine on which it is invoked. See the <i>gethostident</i>(3) manual page for restrictions.</p>
<i>Install</i> (1M)	<p>The <i>Install</i> command is used in Makefiles to install a <i>filelist</i> (a single updated target file or a quoted list of several target files) in a specific place within a file system. Each file in <i>filelist</i> is installed by copying it into the appropriate directory, thereby retaining the mode and owner of the original command. The program prints messages telling the user exactly what files it is replacing or creating and where they are going. <i>Install</i> is provided for compatibility with the IRIS-4D Series software building methods.</p>

- LP spooler The printing server's */usr/spool/lp/rhost* file requires two entries per printing client: the original line printer entry and an additional root entry. This script, */usr/spool/lp/etc/util/rhostfix*, adds the additional entry to the */usr/spool/lp/rhost* file on your system. The script runs automatically when the GL2-W3.6 software is installed.
- monitor(3C)* The IRIS now has the ability to suspend and resume profiling. See the *monitor(3C)* manual page for instructions.
- preset(1M)* The printer script *preset* can be used to reset the printer spooling mechanism on laser printers. The line printer spooler consists of a number of files that can be corrupted. When this happens, the spooling system prints confusing messages to the console, and the job is not printed. *preset* deletes and recreates crucial system files to restore the line printer spooler to its original state. All queued files and all printers are deleted. See the *preset(1M)* manual page for more information.
- TCP/IP flag There is a new flag in TCP/IP that causes the system to treat all networks as if they are local. When the flag is used, data is transmitted through gateways in large segments (packets). To use the flag (after installing the GL2-W3.6 software), it must be patched in the 3.6 kernel. If your system has a reconfigurable kernel, set the configuration option **ALLNETSARELOCAL**. If the system does not have a reconfigurable kernel, contact Customer Service through the Geometry Hotline for instructions.

timed(1M)

The *timed* process daemon runs on the master server in a network, keeping all system clocks synchronized. This is helpful when NFS is running, because */usr/etc/timed* eliminates time skew problems that may occur when the NFS server and client machine clocks keep a different time. (When they keep different times, the client cannot use *make*(1) with an NFS-mounted file that appears to be created in the future.) See the *timedc*(1M) and *timed*(1M) manual pages for more information.

4.2 Changes

Category

Description

cc(1)

The IRIS now issues a warning message for bit fields which have not been declared as unsigned. This increases the compatibility between IRIS series 2000 and 3000, and IRIS-4D Series machines. On the IRIS series 2000 and 3000, bit fields are inherently unsigned; on the IRIS-4D Series, bit fields may be declared either signed or unsigned. This new warning message is useful when programs are ported between IRIS-4D Series machines and IRIS series 2000 and 3000 machines. Bit fields which are enumerated types cause no warning message to be issued.

getmem(3G)

The *getmem* function has been moved from *libgl2.a* to *liboldmalloc.a*.

kernel installation When installing a new kernel on an IRIS, the *kernel(1M)* command must be used. The *kernel* command links the appropriate kernel to *defaultboot* and *vmunix*. Manually copying a new kernel (using *cp*) to either *defaultboot* or *vmunix* causes the previously-linked kernel to be overwritten. Always use the *kernel* command to switch kernels. See the *kernel(1M)* manual page for more information.

malloc A new *malloc*, located in *libc.a*, allocates memory faster than the previous version (which is still available from */usr/lib/liboldmalloc.a*). Use **-oldmalloc** with *cc(1)* and *ld(1)* to link with the old version.

matherr(3M) The use of *matherr* routines is no longer supported. There is a warning issued at load time stating that these routines are present in a program. Although the presence of *matherr* routines do not cause a program to fail, it is suggested that you remove these routines from existing programs.

TCP/IP The address notation used in TCP/IP has been changed. With the GL2-W3.6 release, this notation is recognized:

user@host

The notation used in previous releases (shown below) is no longer supported by the network software.

host.user

tutorials

The file ownership of tutorials has been changed in this release. If you installed the IRIS programming tutorials as part of the GL2-W3.5 release update (December, 1986), change the current tutorial file ownership to agree with the GL2-W3.6 release. To do this, edit the */etc/passwd* file to read as shown below before you attempt to compile and use a tutorial:

```
tutor::994:...  
tutorf::993:...
```

The following functions and their accompanying manual pages are obsolete:

- *quota*(1)
- *servers*(4)
- *tek*(7)
- *kermit*(1C)
- *matherr*(3M)

5. Fixed Bugs

Bugs fixed since the GL2-W3.5 release are listed in this section.

Category	Description
<code>cc(1)</code>	<p>When hardware floating point is used, auto-incrementation and decrementation of pointers to <i>floats</i> is now performed correctly in all cases, even when the pointer is declared as <i>register</i>.</p> <p>Auto-incrementing or auto-decrementing of bit fields that are passed as parameters now works correctly. The bit field no longer has to be altered outside of the argument list.</p> <p>This C expression, <code>*p++ op= x</code>, where <code>p</code> is a pointer to character, now compiles correctly.</p> <p>The allocation of temporary registers for the compiler's internal use has changed. As a result, these two errors occur less frequently: no entry for <code>op REG</code> with a compiler abort due to insufficient address registers; and expression causes compiler loop - try simplifying due to insufficient data registers. See the <code>cc(1)</code> manual page for more information.</p>

In previous releases, inter-language calls between C and FORTRAN functions of type *real* resulted in erroneous output if the FORTRAN function was compiled for hardware floating-point and the C function was not. This has been corrected. Although programs with a mix of hardware and software floating-point modules are not recommended, intercommunication should now work correctly.

In previous releases, some instances of constant expressions involving *integer* and *long float* types yielded type *float* rather than type *long float*. This has been corrected.

ctype(3C)

The macro *isascii()* now functions properly for values outside the range -127 ... 255.

diff(1)

When comparing files, *diff* no longer ignores fragments of lines at the end of either file.

dog(1D)

The demonstration program *dog* works on an IRIS running NFS or TCP/IP.

FORTRAN

FORTRAN now supports recursion, with the stack size limited to 30K bytes for local variable storage and compiler-generated temporary space.

If a C function of type *real* is called from a FORTRAN program compiled with the **-Zf** option, the results are no longer expected to be on the FPA board.

getpwent(3C)

The declarations of the return values of functions described in *getpwent* are now correctly placed in *pwd.h*.

- man*(1) If you interrupt the output process of the *man* command by typing **q** in response to the `--More--` prompt, the *cat* version of the manual page is no longer truncated.
- NFS In previous releases, commands such as *pwd*(1), which compared i-numbers returned by *stat*(2) to i-numbers returned by *readdir*(3X), sometimes failed if the files or directories in question were NFS-mounted from servers with 32-bit i-numbers in their filesystems. Such comparisons sometimes failed because *stat* returned a 16-bit i-number, while *readdir* returns 32-bit i-numbers in its directory entries. *stat* now returns a 32-bit i-number.
- pow*(3M) The math functions *pow* and *_lpow* now work similarly. For both functions, if the base (first argument) is non-positive, the power (second argument) must be a whole number. See the *pow*(3M) manual page for more information.
- termcap*(4) The “vt100” *termcap* entry works with vt100 emulator terminals.
- tutorial ownership* The graphics tutorial ownerships are correctly set as listed below:
“tutor” owns */usr/people/tutorial/c.graphics*
“tutorf” owns */usr/people/tutorial/f.graphics*

6. Known Problems

Known problems are listed in this section, with suggested work-arounds, where possible.

Category	Description
<i>addtopup</i>	<i>addtopup</i> does not accept more than 43 characters in a C program string, or more than 44 characters in a FORTRAN program string. Attempts to use longer strings cause programs to fail without explanation.
<i>bind(2)</i>	If an invalid socket descriptor is used with the <i>bind</i> system call, not only does the call fail, but the system crashes with a kernel trap.
<i>cc(1)</i>	When operations are performed on bit fields within the parameter list of a function, the result passed to the function may be incorrect. This may occur during several types of operations. It may occur when the operation involves assignment of the bit field (although the value assigned may be correct). It may also occur when the operation involves negation, or taking the complement of the bit field. Work-around: Perform operations on bit fields outside of the parameter list, and pass the result.

There is a discrepancy concerning the equivalence of these two C expressions:

```
i = i/f
```

and

```
i /= f
```

The de-facto C language standard indicates that they must be equivalent. It also implies that the data type controlling the calculation is the type of the left hand side. The emerging ANSI standard resolves this discrepancy in favor of equivalence, which is implemented in the IRIS-4D product line, but not in the IRIS series 2000 and 3000 product line, and thus not in this release.

There is a bug which causes incorrect code to be generated for the autoincrementation of some register pointer variables which have been cast. In this code fragment,

```
register char *p;  
*(int *)p++;
```

the compiler increments *p* by `sizeof(int)`, rather than the correct `sizeof(char)`. Due to the possible reliance of existing code on this (incorrect) behavior, the bug has not been fixed. Instead, *ccom* prints a warning message indicating that the pointer has been incremented by the incorrect value.

- When graphics programs are compiled on an IRIS running the GL2-W3.6 release software, and then executed on an IRIS running GL2-W3.5 release software, this message is displayed: unknown grioct1 cmd: 85. The message is only a warning; the program functions correctly.
- check*(1) *check -l [rcsdir]* does not work.
- ci*(1) *ci* will not let you interrupt out of a check-in when it is performing a check-in on more than one file.
- crv* *crv* works incorrectly in feedback mode. The coordinates of the curve displayed may be different from the input coordinates.
- cu*(1C) Pressing the **BREAK** key causes *cu* to hang.
- After *cu* is used to connect with another UNIX system, data may be lost from the currently running process if you switch windows at this time.
- dbx*(1) The command *print p+1*, where *p* is a pointer, functions as the literal *p+1* rather than *p + 1 *sizeof(p)*.
- DSD Controller Multi-volume backup with *tar*(1) does not work with 1/4" tapes on a drive receiving instructions from the DSD controller.
- Swapping the DSD controller for a Storager on a 2400T or a 3020 and adding an ESDI disk causes device name linking problems. The */etc/model* file is used to determine the controller type, but there is no longer a correlation between the model number and the controller type. Work around: Change the model number reference in */etc/model* to "3030" to avoid this problem.

filled polygons Filled polygons in which all the vertices satisfy a plane equation (32-bit floating point and 3-D planar) are not always drawn on the screen.

floppy incompatibility Due to differences between the Storager and the 5217 controllers, floppy disks that are formatted on IRIS 3030, 3120, and 3130 machines with the Storager controller are not readable by IRIS machines with the 5217 controller. The inverse is also true.

FORTTRAN There is a design restriction that may cause problems in your FORTRAN programs: FORTRAN forces all data items to be contiguous. If you initialize one data item, all data items are initialized at compile time and the program may produce a large executable image. The test program below produces a large executable image.

```

        program test
        dimension x(6),g(1000000)
        data x/6*1. 0/
        do 10 i = 1,6
                g(i) = 1.0
10    continue
        end

```

Workaround: Initialize data sections at run time. In the test program above, replacing the `g` in line five with an `x` produces a data section initialized at run time.

The `%f` and `%F` item types do not work with `addtop` in FORTRAN.

Tracing multi-dimensional array elements in a FORTRAN program is not possible.

When FORTRAN uses formatted write statements to print real numbers, it does not print the leading zeros.

In many UNIX FORTRAN environments, the standard input is assigned to unit 5 and the standard output to unit 6. On the IRIS series 2000 and 3000, these standard “files” are both attached to unit0. On both systems, the special unit “*” refers to the (implementation-dependent) unit for standard input and output. When porting programs to the IRIS that expect to use units 5 and 6 for interaction with the terminal, you can explicitly open these units to the special file */dev/tty*. If redirection to a file is attempted, this method fails.

If you plan to use *scrsav* to save screen images from within a FORTRAN program, you must make the following two edits to the file */usr/people/gifts/mextools/imglib/scrsave.c*:

Change the first section declarations under *FORTRAN version* to read as shown below:

```
fortran
scrsav(top,bottom,right,left,cmap,len,name)
/* change to pointers and pass the
/* argument in reverse order */
char *name;
long len;
long len;
long *cmap;
long *left,*right,*bottom,*top;
```

Change the call to *scrsave* (about 15 lines farther down) to read as shown below:

```
scrsave(newstr,*cmap,(Screencoord) *left,
(Screencoord) *right,
(Screencoord) *bottom,
(Screencoord) *top);
```

Problems occur when a FORTRAN program tries to call `winclo(gid)` to close a graphics port because `winclo(gid)` is defined in the FORTRAN version of the Graphics Library as a subroutine, and in a header file as a parameter.

`getbuffer`

After opening a second window from a program, `getbuffer` returns a value of 1. This incorrectly indicates that only the back buffer is enabled, although both buffers are actually enabled.

`heme`

The demonstration program `heme` disables `ttyd3` devices and requires a system reboot to run again.

`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.

The `mex` window manager supports ten open windows at one time. An attempt to open an eleventh window hangs the system.

The `mex` window manager prevents graphics programs from successfully using `winopen` and `winattach` to open and attach to the textports that spawned the programs.

While using `mex`, an attempt to invoke the `script` utility in a directory where you do not have “write” privileges may cause processes in the current window to be killed. This error message also appears: `Can't create typescript.`

If *mex* is used without the `-d` option for double-buffer mode, previously closed shell windows may reappear when you try to run new programs. Press `ctrl-c` remove the extra windows.

If the MOUSEX and MOUSEY valuator are detached from the mouse in a program, *mex* fails to re-attach them automatically when the program has exited. Workaround: Insert a statement at the end of any program in which the MOUSEX and MOUSEY valuator are detached that re-attaches them before the *gexit* statement.

If you log out without exiting from *mex* beforehand, another user may open a window into your account. Choose the *exit* option from the three-option background menu to kill *mex* before you log out.

multmatrix

Matrix operation functions such as *multmatrix* may produce underflow values in a graphics program, ending in undefined results. Matrix calculations that are expected to produce values of zero may instead produce large negative numbers due to this underflow problem. To check for results that are in error, use *getmatrix* and print the matrix it returns.

NFS

If you gain concurrent access to a file on a remote machine via NFS and also through a remote login, you may see two different versions of the file. This happens because NFS caches the file, updating the cache every three seconds.

Pascal Graphics Library

The commands *readRGB* and *writerGB* do not work in the Pascal version of the Graphics Library.

picking

picking sometimes fails when objects are nested to a depth of more than six or seven levels.

<i>tset</i> (1)	The <i>tset -s</i> option does not work properly.
<i>winattach</i>	If <i>winattach</i> is used twice in a graphics program, the system crashes when the program is executed. Check each program at compile time to make sure <i>winattach</i> is not used twice.
<i>winopen</i>	Each graphics program may execute <i>winopen</i> nine times. If you place more than nine <i>winopen</i> statements in a graphics program, the program fails and the system displays this error message: No more <i>gfports</i> for this process!!
<i>xx</i> (1C)	When you perform an <i>xlogin</i> to a remote workstation you must type “~.” (tilde dot) to be disconnected after logging out if you had a process running in the background.

7. Documentation

The following documents are new or have been changed due to the GL2-W3.6 release:

Document Title	Part Number
<i>UNIX Programmer's Manual, Volume IA, Version 3.0</i>	007-0101-030
<i>UNIX Programmer's Manual, Volume IB, Version 3.0</i>	007-0102-030
<i>IRIS Series 3000 Owner's Guide, Version 3.0</i>	007-5220-030
<i>IRIS User's Guide, Version 4.0</i>	007-1101-040
<i>GL2-W3.6 Workstation Release Notes</i>	007-3206-010
<i>Learning to Debug with edge, C Edition</i>	007-0403-010
<i>TCP/IP User's Guide, Version 3.0</i>	007-0330-030
<i>IRIS Communications Guide Update Package</i>	007-0390-011
<i>Learning to Debug with edge, FORTRAN Edition</i>	007-0404-010
<i>IBM 3270 Communications Guide, Version 3.0</i>	007-0340-030
<i>Using the Laser Printer, Version 2.0</i>	007-6103-020
<i>Using the Color Printer, Version 2.0</i>	007-6104-020
<i>Using the 8-port RS-232 Option</i>	007-5254-010

Manual pages for optional software are now packaged as tabbed sections of the *UNIX Programmer's Manual*. If you have optional software from previous releases, remove the manual pages from the software option document and refer to the updated version of the *UNIX Programmer's Manual, Volume IB* for manual page information.

