SciAn Reference Manual

by Eric Pepke and Jim Lyons

May 1993

Supercomputer Computations Research Institute

Florida State University

Tallahassee, Florida

The development of SciAn is supported by the Supercomputer Computations Research Institute, which is partially funded by the U.S. Department of Energy under contract DE-FC05-85ER250000 and also supported by the State of Florida.

Table of Contents

Preface vi

1 Obtaining and Installing SciAn 1

System requirements 1

Obtaining SciAn 1
- Connecting to ftp.scri.fsu.edu 1
- Downloading the SciAn Package 2

Installing SciAn 3
- Configuring SciAn 3
- Checking the configuration 4
- Assigning font mappings 5
- Compiling and linking SciAn 5
- Releasing SciAn 6

Obtaining the documentation 6
- PostScript format 6
- Microsoft Word format 6

Getting help 7
- Using the technical notes 7
- Common problems and solutions 7

2 Command Line Arguments 9

Summary 9

Switches 9

Examples 11

3 Menus 12

The Main menu 12

The File menu 12

The Datasets menu 13

The Object menu 13

The Text menu 14

The Animation menu 14

The Window menu 15

The Color Menu 15

4 File Formats 17

File readers 17

G90 (Gaussian 90) format 18

HDF (Hierarchical Data Format) 18
- Obtaining HDF 18
- Writing HDF files 18
- Extensions to HDF 19
- HDF controls 20
- Example code 22

NFF (Neutral File Format) 23
Comment....................24
Cylinder or conical frustum....................24
Polygon....................24
Polygonal patch....................24
Sphere....................25
Time step....................25
Fill color....................25
Origin....................25
Writing NFF files....................25

PLOT3D format....................27
The PLOT3D metafile....................28
PLOT3D reference....................29
Example....................30

PDB (Protein Data Bank) format....................31
Record summary....................31
Dataset construction....................33
Atom parameters....................34

STF (Simple Text Format)....................34
Background concepts....................34
Writing the header....................35
Writing the data....................36
Writing the grid....................37
STF statement reference....................38
Examples....................40

SY format....................41

5 Scripts....................42

Syntax....................42

Script commands....................43

6 Recorder Drivers....................52

7 Programming Concepts....................53

Policy....................53

Language requirements....................53

The SciAn source code....................54
machine.h....................54
Making SciAn....................55
Source files....................55

The design of SciAn....................56
Data types....................57
Memory allocation....................57

The object system....................57
Plain objects....................58
Numeric objects....................59
Character strings....................59
Arrays....................59
Geometry....................60
Pictures....................60
Palettes....................60
Symbols....................60
Variables and methods....................60
Setting variables....................61
Getting variables....................61
Setting methods....................62
Sending messages....................63

Deferred messages....64
Variable dependencies....64
Garbage collection....65

8 Writing a Recorder Driver....67

Setting up the files....67

Implementing the recorder driver....69
Creating the object....69
Implementing the methods....69
Registering the recorder driver....75

9 Writing a File Reader....77

References....78

Preface

SciAn is a scientific visualization and animation program for high-performance graphics workstations. This document describes version 0.80 of SciAn.

This is the reference manual for SciAn. It is a companion volume to *SciAn User's Manual*, referred to in this manual simply as the user's manual. The user's manual describes the basic operation of SciAn and provides enough information to learn and begin using the program. This manual covers the more technical aspects of SciAn, including the installation process.

While *SciAn User's Manual* is designed to be easy to use, even for people who are unfamiliar with visualization and graphical user interfaces, *SciAn Reference Manual* is a lot rougher. It contains many more technical details and has a lot of sections which are not highly polished. It also assumes a higher level of experience with UNIX, computer hardware, and visualization algorithms.

If you do not find what you are looking for here or in the user's manual, first look at the technical notes located in the `pub/SciAn/technotes` directory on the ftp site. Also look in the `util` and `examples` subdirectories in the general release. If you still can't find what you're looking for, write us at `scian-info@scri.fsu.edu`.

1 Obtaining and Installing SciAn

This chapter is essentially the README file located in the SciAn directory of our ftp site. It is included here for completeness.

System requirements

SciAn should run on all Silicon Graphics Power Series, Personal IRIS, and Indigo workstations with Z-buffer capability. It will not run on very old workstations, such as the 1000, 2000, and 3000 series.

SciAn will run only on IBM RISC/System 6000 workstations that have 3-D graphics accelerators that provide GL compatibility. SciAn also requires a Z-buffer. If there is any question about whether your hardware provides this, please get in touch with your system manager or IBM sales representative. (IBM makes a bewildering variety of hardware, but the magic words are Z-buffer and 3-D GL compatibility.)

On all workstations, SciAn *absolutely requires* a Z-buffer. If you try to run it on a workstation that does not have a Z-buffer, you will get strange results.

Because SciAn is distributed as source code, you must also have access to a C compiler to install. A FORTRAN compiler is also required to install certain features such as the PLOT-3D file reader. A single set of source files is used to produce all variations of SciAn.

SciAn may also optionally be installed using the Hierarchical Data Format (HDF) library developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. This library is available via anonymous `ftp` from `ftp.ncsa.uiuc.edu`. We strongly recommend that you obtain this library, as it provides a good data format.

Obtaining SciAn

SciAn is normally distributed via anonymous ftp from `ftp.scri.fsu.edu`. If you do not have access to anonymous ftp, send us mail, and we'll try to figure out some other way to get you the program. It is much easier to get the program using `ftp`, however, and it's certainly easier to get updates that way.

Connecting to `ftp.scri.fsu.edu`

Before you connect to the SciAn distribution machine, make sure that you are in a directory on your machine where you want the SciAn source to reside. You will need a few megabytes to keep the files during the installation process.

To connect to `ftp.scri.fsu.edu`, start up `ftp` on your unix machine like this:

```
ftp ftp.scri.fsu.edu
```

When asked for a user name, enter `anonymous`. When asked for a password, enter your network electronic mail address.

The SciAn program and documentation are located in the SciAn subdirectory of the pub directory. To get into that directory, enter

```
cd pub/SciAn
```

In that directory, you will find a `README` file and several subdirectories.

The `release` subdirectory contains the release versions of SciAn. Obtaining SciAn from this directory is described in the next section.

The `beta` subdirectory contains versions of SciAn for our beta testers. These versions are not as thoroughly tested as the release versions. When a version passes beta test, it is moved to the `release` directory.

The `patches` subdirectory contains minor patches to particular versions of SciAn which do not require the bother of recompiling the entire source. You will receive notice of patches through the SciAn mailing list.

The `documentation` subdirectory contains documentation for SciAn. Obtaining and printing it is described in “Documentation” below.

The `technotes` subdirectory contains technical notes for using SciAn. These are text files that contain information about topics that we haven’t had time to put in the manual.

Downloading the SciAn Package

In the `release` subdirectory (enter `cd release` to get to the directory) you will find several files. They will all have names of the form `scianXXX.tar.Z`, where `XXX` is the version number of SciAn. For example, the file `scian060.tar.Z` contains SciAn version 0.60.

It is usually best to get the latest version of SciAn, unless you have received a note to use an older version.

You must download the file in binary mode. Most versions of `ftp` figure out that the target machine is running UNIX and go into binary mode automatically. Just to make sure, enter

```
image
```

to put `ftp` in binary mode. Now it is time to download SciAn. Let’s assume that you want to download SciAn version 0.60. Enter

```
get scian060.tar.Z
```

After the file has been transferred, you can get out of `ftp` by entering `bye` or `quit`, depending on the version of `ftp` you have.

The SciAn distribution is a compressed `tar` file. The first step in getting the file ready is to uncompress it. Enter

```
uncompress scian060.tar
```

When the uncompress command finishes, you will have a file called `scian060.tar`. To extract the individual files from the `tar` file, enter

```
tar –xvof scian060.tar
```

If you get an error message, try

```
tar –xvf scian060.tar
```

Many files will be extracted from the SciAn `tar` file. Files ending in `.c`, `.h`, and `.f` are the source files of SciAn. The `Makefile` and files with `.make` in their names are for the `make` program. There is also a directory called `demo` which contains demonstration files for SciAn. There may also be a file called `RELEASE.NOTES`. Look in this file to determine special features of the version you have just downloaded.

Installing SciAn

Before SciAn can be installed, it must be configured to your machine. In order to understand the process of configuration, you must first understand how the SciAn source is structured.

One copy of the SciAn source is used to produce all variations of SciAn. Which variation is being compiled depends on constants defined in the file `machine.h`. At compile time, the file tests constants to see if it is running on an IBM or a Silicon Graphics machine and adjusts automatically. However, there are several options that you need to set by hand.

There are five basic steps in installing SciAn: Configure SciAn; check the installation options; assign font mappings for special text characters; compile and link SciAn; and release SciAn. These steps are described in the next five sections.

Configuring SciAn

The `machine.h` file controls how SciAn is installed. All it contains are preprocessor directives such as `#define` which have the effect of defining constants that control the installation of SciAn. There are plenty of comments in the file to help you understand exactly what is going on.

The installation process has been automated so that you probably won't have to change anything in this file. However, it is a good idea to be familiar with the file in case problems arise.

At the beginning of the file is some code which tries to figure out what kind of machine is compiling SciAn. This always seems to work properly and should not need to be changed. Nearly all of the installation options are set automatically based on the machine, but there are a few described later that you will need to know about.

Next there are some comments that explain what each one of the defined constants does. Next is some code which defines constants based on which machine SciAn is being compiled on. Finally there are the defines for the user configurable options.

One important option is the HDF file format option, which is controlled by the `HDFDEF` constant. You can choose to compile SciAn with or without the HDF library. By default, the distribution is set up to compile without HDF. However, we strongly recommend that you obtain and install HDF, as SciAn is much more useful with it. Information on how to get HDF can be found in "Obtaining HDF" in Chapter 4. The HDF library needs to be installed and placed in `/usr/local/lib` before you begin installing SciAn. The `make INSTALL` process will produce a file called `machine.hdf.h` which will declare whether the HDF library is to be installed or not.

If you are compiling on an IBM RS/6000 machine, you may need to define the `MENUSFROM0` constant. The numbering of menu items in GL is supposed to start at 1. However, some versions of the IBM GL emulation, such as with AIX 3.1.5, start at 0. There isn't really any way to tell this beforehand, but if you compile SciAn and notice that the wrong menu items are getting highlighted, you will need to go back and change this.

Checking the configuration

After you have set up `machine.h` the way you want it, you need to check the configuration. This is an extremely important step, and if you forget to do it, you may get link errors later. Enter

```
make INSTALL
```

A program called `ScianPreInstall` will go through the options defined in `machine.h` and check to make sure that the libraries that are needed are present on your system. When `ScianPreInstall` runs, it will ask you questions and print some messages to the console. *Read everything it prints!* `ScianPreInstall` will tell you if there is anything that needs to be changed in the libraries or in `machine.h`. Do not proceed to the next step until `ScianPreInstall` tells you that it is OK to do so.

`ScianPreInstall` also writes two very important files into the directory: `flags.make`, which contains compiler flags, and `lfiles.make`, which contains link files. The way it produces these files is by reading a template file specific to the kind of machine you are compiling on. (To see all the template files, enter `ls *.*.make`.) Then it adds optional link files based on the options in `machine.h`. If, for some reason, you need to change the link files because your system is unusual, do not edit `lfiles.make`, because it will be overwritten the next time `make INSTALL` is done. Instead, edit the appropriate template file.

If and only if `make INSTALL` has given you the go-ahead to compile, you can go on to the next step.

Assigning font mappings

The next step in the installation is to assign font mappings. SciAn uses some characters in the Silicon Graphics font library which are outside the normal range of ASCII characters. Unfortunately, which characters are which has changed as new versions of the operating system and the font manager have come out. It is necessary for you to tell SciAn what the mappings are. To do this, enter

```
make FONTS
```

at the console. If you are running on a Silicon Graphics workstation, you *must* be at a graphics console when doing this. On all other machines this will just produce a message and generate a file, and you can go on to the next step. On the Silicon Graphics, this will run a program that will put up a window which will ask you to find the various characters. Follow the directions given to you by the program, and then you will be able to go on to compiling SciAn.

This step really only needs to be done when you install SciAn for the first time, or when you do an upgrade of the operating system which changes the characters in the fonts. If you are having any trouble with special characters within SciAn, e.g., the copyright C in a circle doesn't appear on the Copyright help screen, then you must do this step and recompile SciAn.

Compiling and linking SciAn

To make the SciAn executable file, enter

```
make scian
```

If you are making on a Silicon Graphics machine with more than one processor, you can do a parallel make instead by entering

```
pmake scian
```

The `makefile` will compile and link SciAn resulting in an executable named `scian`. If you get any kind of error, make sure that the configuration is set up correctly as described in "Configuring SciAn" and "Checking the configuration" above. If that doesn't help, read through the section "Common problems and solutions" below. If you still cannot figure out what is wrong, send mail to us at `scian-bugs@scri.fsu.edu`.

When `scian` has been made, test it out by typing `./scian`. The user's manual has a brief tour, but if you don't have a copy of the manual, you can get on-line help by clicking the mouse in the title window.

Releasing SciAn

The final step in installing SciAn is to release it to the users of your machine.

First copy the `scian` executable file to a place where users can reach it. One good place to put it is in `/usr/local/bin`.

Then copy or move the `demo` directory to some directory where users can reach it and tell your users where it is. The user manual refers to this directory, so your users need to know its location.

The executable of the Silicon Graphics version of SciAn should run on any Silicon Graphics workstation. However, the IBM RS/6000 version will probably only run on workstations configured in a similar way to the workstation where it was compiled. If you are running in a heterogeneous environment with several different versions of AIX, you may need to keep several executables of SciAn.

Obtaining the documentation

Documentation can be downloaded in two forms: PostScript files and Macintosh files for Microsoft Word 5.

PostScript format

PostScript files for the manuals are located in the `documentation` subdirectory. The name of the user manual is of the form `userXXX.ps.tar.Z`, and the name of the reference manual is of the form `refXXX.ps.tar.Z`, where `XXX` is the version of SciAn. For example, for version 0.80 of SciAn, there is a user manual named `user080.ps.tar.Z` and a reference manual named `ref080.ps.tar.Z`. There may also be a file of the form `colorXXX.ps.tar.Z`, which contains a color plate that can be printed out on a color PostScript printer.

Each file is a compressed `tar` file, similar to the file containing the SciAn software, as described in “Obtaining SciAn” above. Make sure that `ftp` is using the binary format by typing `image` and then download the files. After you exit `ftp`, uncompress the files using the `uncompress` command. Then use the `tar` command to extract the files from the archive. For example:

```
uncompress user080.ps.tar
tar xvf user080.ps.tar
```

The `tar` command will produce a number of PostScript files, all ending in the `.ps` extension. You can print out these PostScript files according to the rules for the printers at your site.

Microsoft Word format

Microsoft Word files are located in the `documentation` subdirectory. The name of the user manual is of the form `userXXX.msw.sit.hqx`, and the name of the reference manual is of the form `refXXX.msw.sit.hqx`, where `XXX` is the version of SciAn. For example, version 0.80 of SciAn has a user manual named `user080.msw.sit.hqx` and a reference manual named `ref080.msw.sit.hqx`.

There may also be a file of the form `color080.msw.sit.hqx`, which contains a color plate that can be printed out on a color printer.

The manual is in Microsoft Word 5 format and has been compressed and encoded with `Stuffit`. Use `BinHex4` or `StuffIt` to decode the file and then use `StuffIt` to extract the archive. When printing the manual, be sure to check Color/Grayscale in the print dialog. Use background printing. Printing the manual requires a lot of memory, so it may be necessary to exit all other applications while the manual is printing. If this fails, you can also print the manual in sections.

Getting help

We hope you enjoy SciAn. If you have difficulty installing it, please first check through this chapter to make sure that you are doing everything correctly. Be sure to look through “Common problems and solutions” below. If you still have problems, send a message to `scian-bugs@scri.fsu.edu`.

Using the technical notes

When we need to release information about some aspect of SciAn that cannot wait for the long time delay in releasing a new version of the manual, we will put a technical note in the technotes directory. A file called INDEX will give summaries of all the technical notes in the directory.

Technical notes are not guaranteed to be easy to understand and may cover esoteric topics.

Common problems and solutions

This section contains some problems that a few people have had during installation. Please check this section if you have a problem.

PROBLEM: I get link errors.
SOLUTION: About 95% of the time this is due to forgetting to do `make INSTALL` before doing `make scian`. `make INSTALL` is a very, very important part of the installation process. It will set up the link files to link properly if it can and will tell you what to do to fix it if it can’t.

PROBLEM: When I link, I see a message saying that `ScianFontMapper.h` can’t be found.
SOLUTION: This indicates that you haven’t gone through the `make FONTS` process to generate the font mappings. See “Assigning font mappings” above.

PROBLEM: When I link, I get a message saying that `libdf.a` can’t be found.
SOLUTION: The `libdf.a` file is the file that contains the HDF library. Review “Configuraing SciAn” above to find out what to do about it. Also, be sure to do `make INSTALL` before doing `make scian`.

PROBLEM: When I link, I get `DFSDgetmaxmin` or `DFSDgetrange` undefined.
SOLUTION: Between versions 3.1 and 3.2 of the NCSA HDF library, they changed the name of one of the routines from `DFSDgetmaxmin` to `DFSDgetrange`. SciAn can link using either, but you have to specify in `machine.h`. Look for constants beginning with the letters `HDF`. The `make INSTALL` process will tell you if SciAn is correctly configured for the right version of HDF.

PROBLEM: SciAn compiles fine on the IBM, but when I run it, I get an error and it exits.
SOLUTION: This one can mean a wide range of things, from trivial to disastrous. In ascending order of difficulty, these are some of the things we have run into:

- When a user brings up X, some versions of AIX assume that the user "owns" the GL emulator and does not allow anyone else to use it. Make sure that you are the user that ran `xinit`.
- Binaries compiled under different versions of the operating system will not run on other versions. (Also see `MENUSFROM0` in `machine.h`.) Make sure that you have compiled and run on the same machine.
- There are some software configurations that prevent you from using the GL emulator from within X. Check with your system manager to see if this is the case.
- If you have multiple versions of the X library, the search path specified may get the wrong one. If this is the case, change the order of the directories in `lfiles.ibm6k.make` and rerun `make INSTALL`. *Do not* change `lfiles.make`; it is written by `make INSTALL`.
- SciAn does not run on every IBM system, only on those with graphics accelerators that can do 3-D GL emulation, have Z-buffers, and can do RGB color.

PROBLEM: When I try to run it on the IBM, the wrong menu items appear to highlight, and the fonts don't work in text boxes on the screen.
SOLUTION: This probably means that you have compiled under AIX 3.1.5 without defining `MENUSFROM0` in `machine.h`. Take a look at `machine.h`, run `make INSTALL` again, and try to make again. This problem is further explained in "Configuraing SciAn" above.

2 Command Line Arguments

This chapter describes the `scian` command and gives the meaning of all the command line switches accepted by SciAn.

Summary

`scian` [`-s` *script*] [`-l` *log*] [`-o` *directory*] [`-CdSv`] [`-1c`] [`-1l`] [`-1o`] [`-1p`] [`-1r`]
[`-f` *format*] [*filename*...]

The SciAn command line consists of the name of the program (`scian`) plus zero or more arguments.

Arguments beginning with a minus sign (−) are switches. Some switches take an additional argument; some do not. Switches that take an additional argument such as a file name must immediately be followed by the argument, separated by one or more spaces.

Alphanumeric arguments which are not arguments for switches are considered names of files to be read into SciAn. The entire file name, including the extension, must be included. If an `-f` switch has not appeared yet, the types of files will be determined by their extension.

Command line arguments are read and processed from left to right.

Switches

Here the switches recognized by SciAn:

`-C` Normally, SciAn will abort whenever it encounters an error in a script. The `-C` switch causes SciAn to continue reading past script errors. Be careful when using this option, as the effect of errors within scripts can be cumulative.

`-d` Runs SciAn in demo mode. In this mode, text is just a little bit bigger and blacker so that it can be more easily seen by somebody looking on.

`-f` *format* Sets the current data format to *format*, a short string indicating the file format. The format will be used to read all subsequent files on the command line, regardless of their filename extensions.

`-l` *name* Causes SciAn to log all operations to a SciAn script in a file called *name*. Both the `-s` flag and the `-l` flag can appear on a single command line, but the file names must be different. The `-l` switch and associated file name can appear any number of times on the command line. If *name* is a single hyphen (-), the log will go to standard output. This is sometimes handy for debugging scripts.

`-o` *name* Opens the directory named *name* in a new file browser. You can use this switch more than once on the command line to open up multiple directories.

`-s` *name* Causes SciAn to read a script called *name*. If present, this should be the first argument on the command line. This is because subsequent switches may open windows, and SciAn needs to know it is reading a script to do automatic window placement. None of the lines in the script are actually processed until the entire command line has been read.

`-S` Normally, SciAn scripts do not contain commands to select and deselect objects. This is normal for movies, where you don't want to see selection marks in the final product. However, if you are using scripts for some other purpose, such as to demonstrate SciAn, include the `-S` switch to see the selections. The `-S` switch affects both logging using `-l` and script reading using `-s`.

`-v` Enables video recording. Normally, all recording commands just print messages to the console for debugging. This is the default so that it is harder to make a mistake. The `-v` switch on the command line enables all the recorder drivers.

`-1c` Causes SciAn to use only one clock for all spaces created. Normally, each new space is given a new clock. Note that the first character in this switch and the following four switches is the numeral one, not the letter l.

`-1l` Causes SciAn to use only one set of lights for all spaces created. Normally, each new space is given a new set of lights.

`-1o` Causes SciAn to use only one observer for all spaces created. Normally, each new space is given a new observer.

`-1p` Causes SciAn to use only one color palette for all datasets created. Normally, each new dataset is given a new palette.

`-1r` Causes SciAn to use only one renderer for all spaces created. Normally, each new space is given a new renderer.

Examples

```
scian -d terrain.hdf
```

Runs SciAn in demo mode and opens file `terrain.hdf`. The file format is determined from the file extension.

```
scian -s script1.log -l script2.log
```

Runs SciAn to first read `script1.log`, saving the results in `script2.log`.

```
scian -o demo
```

Runs SciAn automatically opening the demo directory in a window.

```
scian -s movie.log -v
```

Runs SciAn reading an animation script from `movie.log` and actually producing the movie.

3 Menus

This chapter contains descriptions of the menu items within SciAn.

The Main menu

Press with the right mouse button anywhere in the interior of a window to bring up the Main menu. The contents of the Main menu may vary depending on the window.

All windows have at least three items. The Help item brings up the help window. The Preferences item brings up the Preferences window as described in Chapter 10 of the user's manual. The Quit SciAn item first confirms that you really want to quit using a dialog window and then quits.

The remaining items in the Main menu are submenus, which are described in the next few sections.

Hint *Don't bring up a menu too close to the right side of the screen. This makes submenus awkward and difficult to use.*

Some common functions are available through buttons on the screen as well as menus. In general, all functions are available through menus, while buttons show the most common functions for selected objects in that window.

The File menu

The File menu contains items for working with files.

The Open item opens the selected file icons and reads the datasets they contain. It has the same function as the Open button on the file browser window. If any folder icons are selected, SciAn will open new file browser windows for them.

The Set Format item brings up a dialog window which allows you to set the file format of the selected files. It has the same effect as the Set Format button.

The Show Info item brings up a dialog window which shows information on the selected files. It has the same effect as the Show Info button.

New File Browser asks for a directory and then brings up a new file browser window for that directory.

The Show File Readers item brings up the file readers window, which contains icons for all the file readers.

The Save Palette item is only available within the control panel of a color palette. It saves the settings of a palette to a file in the directory where the dataset was read. After a palette has been saved, any dataset with the same name in that directory will use that palette automatically.

The Datasets menu

The Datasets menu contains items for working with datasets.

The Visualize item visualizes the selected datasets using their default visualizations together in a new visualization window.

The Visualize As item determines all the possible ways of visualizing the selected datasets. Then it brings up a window containing templates for the visualizations. You can select and visualize these templates or drag them into existing visualization windows.

The Modify item brings up a window with all possible modifications of the selected datasets. This item really isn't very useful yet, as Visualize As automatically does all important modifications.

The Show Datasets item shows the window containing icons for all the datasets and brings it to the front.

The Object menu

The Object menu contains items for working with the currently selected objects.

The Arrange submenu contains items which control the placement of drawing objects on space panels. The Bring to Top item brings selected drawings on space panels to the top. The Push to Bottom item pushes selected drawings to the bottom. The Move to Front Panel item moves selected drawings from the back to the front space panel. The Move to Back Panel item moves selected drawings from the front to the back space panel.

The Pick Up item picks up the selected objects. Once a group of objects are picked up, click in a window to drop the objects. Pick up and drop are provided to support systems where dragging between windows does not work. If it does work, it is much easier to use than pick up and drop.

The Show Controls item shows the control panels of the selected objects.

The Turn On and Turn Off items turn the selected objects on and off. This is most useful for visualization objects and lights in visualization windows. It affects all windows where the objects are present.

The Duplicate item duplicates the selected objects in the window.

The Make Local Copy item applies to shared objects. It makes the selected shared objects in the window into local copies. The names of the icons may change to reflect the new identities, but the locations will remain the same.

The Edit Palette item brings up a control panel to edit the color palette of the selected data set.

The Activate item applies only to recorder drivers. This item activates the selected recorder driver for use in all subsequent recording.

The Delete item deletes the selected objects from the window.

The Select All item selects all the selectable objects in the window.

The Deselect All item deselects all objects.

The Text menu

The Text menu contains items for working with text.

Cut Text, Copy Text, and Paste Text work with the selected text in the active text box. Cut Text removes the selected text from the text box and places it on the text clipboard. Copy Text copies the selected text from the text box onto the clipboard without removing it from the text box. Paste Text pastes the contents of the clipboard into the text box at the insertion point, replacing any text that was selected. The text clipboard only works within SciAn.

The Text Font submenu contains fonts for annotations, clock displays, etc. Some fonts which have more than one style have submenus for the style. Choose a font and all of the text in all selected text boxes will switch to that font.

The Text Size submenu contains several font sizes. Choose a size and all of the text in all the selected text boxes will switch to that font size. Because of the way fonts are implemented, there may not be a copy of the font in the exact size you want. In this case, the font will be scaled or the closest available size will be used.

The Text Alignment submenu contains three items for specifying text alignment. Align Left causes text to be aligned flush on the left side of the text box. Align Center causes text to be centered around the middle of the text box. Align Right causes text to be aligned flush on the right side of the text box.

The Animation menu

The Animation menu only contains one item: Show Recorder Drivers. This menu item shows a window containing icons for all the video recorder drivers in SciAn.

The Window menu

The Window menu contains items for working with a window.

The Window Location submenu contains items for locating and sizing the window on the screen. Full Screen makes the interior of the window occupy the entire screen. Video Screen makes the interior of the window occupy an area at the lower left of the screen just the right size for a Silicon Graphics IRIS in NTSC mode. Double Video Screen makes the interior of the window twice as large as the video screen. This is good when recording a video using 2-to-1 reduction in the renderer. You do lose about a dozen pixels on the right of the image when you do this. The Previous Location item snaps the window back to the location before the last window change.

The Tile Full Screen and Tile Video Screen items tile all the visualization windows in the full screen or the video screen, respectively.

PHSCologram Screen makes the interior of the window occupy an area the right size and shape to produce images for a PHSCologram, a kind of 3-D picture produced by the Art^n Institute in Chicago, IL. [Right now this isn't very friendly, but eventually it will be a single operation to produce a PHSCologram. -EMP]

The Show Front Panel Controls and Show Back Panel Controls items only appear on visualization windows. They show the control panels for the front and back space panels respectively.

The Show Space Controls item only appears on visualization windows. It shows the control panel for the space within the window.

The Show Control Panel and Hide Control Panel items show and hide the controls of a visualization window.

The Show Window Frame and Hide Window Frame items show and hide the frame of the window. They only appear if your version of the operating system supports hiding window frames.

The Close item closes the window. On some systems, such as the IBM RS/6000, this is the only safe way to close a window. On other systems, such as the Silicon Graphics machines, it is safe to use the Close item from the Motif window menu. [The IBM RS/6000 appears to make the Close item exit the program instead of sending an event to the GL event queue. I don't know why. The handling of the interface between the X queue and the GL queue is sloppy in other ways as well. We intend to strip out the GL queue routines and plop in X routines to try to get around this problem. -EMP]

The Color Menu

The Color menu is only present in a Palette control panel.

The Save Palette item saves the settings of the control panel in a special file.

The Keep Changes item keeps the changes you have made to the palette so far, but does not save them to a file.

The Revert to Original item puts the palette back to its state when you last chose Keep Changes.

The Color Model submenu allows you to select among the RGB, YIQ, HSV, and HLS color models. (See Chapter 5 in the user's manual for a discussion of color models.)

The Special Effects submenu contains items which do special effects on the currently selected range of colors in the palette control panel. The Ramp item generates a smooth ramp of colors between the colors at the two ends of the selected range. The Reverse item reverses the selected range of colors. The Ruffle item reverses short sections of the selected colors, producing a contour-like effect. The Smooth item smoothes out abrupt changes in the range of colors. The Sharpen item exaggerates changes in the selected color range.