Sun/IRAF Installation Guide

Doug Tody

Central Computer Services
National Optical Astronomy Observatories†
June 1989

ABSTRACT

This document describes how to install IRAF on a Sun workstation, or update an existing installation. Both standalone and networked, multiple architecture configurations are described. Only those issues which one must understand to install Sun/IRAF are discussed here; a companion document, *Sun/IRAF Site Manager's Guide*, deals with other issues such as interfacing new devices, configuring the IRAF networking system, Sun/IRAF shared libraries, adding layered software, and so on.

April 12, 1990

†Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

Contents

1. **Introduction** 1

2. **Installing Sun/IRAF** 2
   - 2.1. Prepare the root IRAF directory 2
     - 2.1.1. If updating an existing IRAF installation 2
     - 2.1.2. If installing IRAF for the first time 3
   - 2.2. Install the files 4
     - 2.2.1. Distribution tape format 4
     - 2.2.2. Installing the main system 5
     - 2.2.3. Configuring the BIN directories 5
     - 2.2.4. Deleting unused HSI binaries 7
     - 2.2.5. Network software distributions 7
   - 2.3. Merge local revisions back into the new system 8
   - 2.4. Run the INSTALL Script 8

3. **System Checkout** 9

Appendix A. A Complete Example 11

Sun/IRAF Installation Guide

Doug Tody

Central Computer Services
National Optical Astronomy Observatories†
June 1989

1. Introduction

Before installing Sun/IRAF, one must 1) obtain an appropriate Sun/IRAF distribution from the IRAF project, 2) select the server or node on which the system is to be installed, and arrange for sufficient disk space to hold the system, and 3) set aside sufficient time to do the installation. If these directions are followed carefully and mistakes are avoided 1-2 hours should suffice to do the installation.

At the present time there are three main distributions of Sun/IRAF, representing the range of workstations currently marketed by Sun, all of which are directly supported by the IRAF project. These are the following:

- Sun-3 workstations running SunOS-3.
- All Sun-3 and Sun-4 workstations running SunOS-4.
- The 386i (Roadrunner), running SunOS-4.

These different distributions are all very similar at the source level (they are identical except for several files) but very different binaries are needed for a "load-and-go" installation on the different systems. Sun/IRAF V2.8 will be the last release supporting OS-3. Sun/IRAF can also be made to run on the older Sun-2 workstations, although this is not directly supported by IRAF since we don't have access to any Sun-2 systems.

The new SPARCstation class of machines is supported by the Sun-4 OS-4 binary distribution, which will run on the SPARCstation, although the current version of Sun/IRAF cannot yet be compiled under the new version of SunOS (4.0.3) provided with the SPARCstation. For normal runtime usage the 386i can be supported as an architecture (BIN) under the OS-4 distribution, e.g. from a central Sun-4 server, but for full programming functionality a separate 386i based system is required.

The amount of disk space required to install IRAF depends upon the system configuration, primarily the number of architectures (SPARC, 386i, Sun-3 with f68881, Sun-3 with fpa, etc.) one needs to support. The main system, including both the core system and NOAO package sources, requires about 36 Mb, less if stripped after installation. Each core system binary requires about 9 Mb and each NOAO package binary requires about 7 Mb (assuming OS-4 and shared libraries). The actual numbers will vary slightly depending upon the architecture.

---

†Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

2. Installing Sun/IRAF

Although the details of how Sun/IRAF is installed or updated depend upon the type of distribution and the desired local system configuration, the basic procedure is always the same. First one obtains a distribution, usually by writing to NOAO and requesting the normal tape distribution, then one follows the procedure outlined below to install the system. Most of these steps should be performed while logged in as 'iraf'; superuser permission is required in the final stages of the installation, to run the `install` script.

```
# Prepare the root IRAF directory.
if (new installation) then
    create 'iraf' account
else if (updating an old installation) then
    save locally modified files
    delete old version of iraf
endif

# Install the files.
use the 'tar' program to unpack the distribution files
configure the BIN directory for each supported architecture

# Merge local revisions into new system.
if (updating an old installation) then
    merge locally modified files back into new system
endif

# Run the INSTALL script (as superuser).
# Checkout the new system.
```

It is important to realize before beginning the installation that IRAF is not an isolated program or collection of programs, but a complex system in its own right, providing a full programming environment, support for the addition of layered software (including locally added software), support for multiple architectures, and so on. Someone who is familiar with the usual installation procedures for UNIX add-on programs will get tripped up if they try to follow similar procedures for installing IRAF, without first reading these installation instructions carefully.

If problems should arise during the installation help is available by calling the IRAF HOTLINE (602-323-4160), or by sending electronic mail to `iraf@noao.edu` (internet) or `5355::iraf` (SPAN).

2.1. Prepare the root IRAF directory

2.1.1. If updating an existing IRAF installation...

If you are updating an existing IRAF installation then you will be replacing IRAF by the new version, and IRAF should already have an account and root directory on the desired host system. You should save any locally modified files and delete the old system, e.g., login as '`iraf`' and enter:

```
% cd $iraf†
% tar -cf /tmp/SAVE.tar local dev unix/hlib
% /bin/rm -rf *
```

† $iraf symbolizes the UNIX pathname of the root IRAF directory.

There are many variants on this, e.g., you could run `find` to determine which files need to be saved and later merged back in, and you could copy these files to some other directory, rather than making a full tar backup. Although we suggest saving the entire directories listed above, in practice only a few files are likely to have been modified, e.g.,

dev/devices
dev/hosts
dev/termcap
dev/graphcap
hlib/extern.pkg
hlib/login.cl
hlib/zzsetenv.def
local/.login

Once the old system has been deleted you are ready to install the new one, as described in §2.2. Note that it is essential to delete the old system to avoid creating junk files or directories when the new system is installed (due to file or directory name changes or deletions).

2.1.2. If installing IRAF for the first time...

If you are installing IRAF for the first time then the first step is to set up a new account for the fictitious user '`iraf`'. This is unconventional but is recommended for the following reasons:

- All IRAF system management should be performed using some derivative of the environment provided by the "." files in the `iraf` login directory. If this is not done important environment definitions may be missing which are required for the correct execution of the software (this affects only IRAF system management, not normal runtime usage).
- Multiple people may need to be IRAF system manager. Having a separate account avoids the need for one user to know another user's password. Even if there is only one site manager at your site, it may be necessary to give login information to the IRAF HOTLINE personnel to allow them to investigate a problem.
- Having IRAF owned by root is not a good solution as then anyone who needs to serve as IRAF site manager would require the root password.

The common practice on Sun-3 and Sun-4 systems is to locate the IRAF root at `/usr/iraf`, although any other directory would do (try to keep the path to the root short to avoid later filename truncation when IRAF is run). Note that the *login* directory for the iraf account should be `$iraf/local` (e.g., `/usr/iraf/local`), rather than the more conventional `$iraf` or root directory, as we want to keep all the locally modified files in subdirectories off the iraf root, to simplify site management. If this point is missed the iraf environment will not be set up properly, and later problems are sure to result.

On 386i systems Sun recommends placing exportable layered products such as IRAF in the 386i-specific directory `/files/vol`. Hence the recommended root directory for IRAF on the 386i is `/files/vol/iraf`. Due to the mandatory yellow pages feature of the 386i, the easiest and most reliable way to create a new user account is with SNAP. If SNAP is used to create the IRAF account the home directory will be set to `/home/iraf`.

To create the iraf root and login directories and make them the default for the iraf account, do the following (as superuser):

```
# rm -rf /home/iraf
# mkdir /files/vol/iraf /files/vol/iraf/local
# /etc/chown -R iraf /files/vol/iraf
# ln -s /files/vol/iraf/local /home/iraf
```

Do not worry about configuring the environment files for the new account as these will be created when the iraf system is later restored to disk.

2.2. Install the files

If you have not already done so, log into the iraf account so that the files when restored will belong to iraf. Mount the distribution tape, which should be either a 6250 bpi 9 track tape, or a Sun cartridge tape.

If you are installing IRAF on a system which has a local tape drive you can skip what follows go to §2.2.1.

If the tape drive is on a remote node connected via the network then it is simplest to copy the files to a temporary disk on the remote node, e.g., with the unix utility `dd`, then use `rsh` and `cat` to pipe the remote file into the standard input of `tar` to unpack it on the local node. Do not use NFS to access the remote file on disk directly, as NFS is not an error corrected transfer protocol and data corruption can result if there are any problems with the networking interfaces on your systems (using `rsh` is also more efficient).

For example, if

```
% tar -xpf /dev/nrmt8
```

would be used to unpack a tarfile from tape to disk on the local node,

```
% dd if=/dev/nrmt8 of=file.tar bs=10240       # remote node
% rsh node "cat file.tar" | tar -xpf -         # local node
```

will accomplish the same thing using the network and an intermediate disk file (the block size shown is for a standard tar file on a 9 track tape). It is also possible to remotely execute `dd` to read directly from the tape, eliminating the disk file, if you are certain of the current file position of the tape.

2.2.1. Distribution tape format

Beginning with IRAF version 2.8, distribution tapes consist of multiple files separated by tape marks, with a TOC (table of contents) file as the first file on the tape. To find out what is on the tape, rewind it and read out the TOC file as follows (the device names required for your site may vary from those shown).

For a 9 track drive:

```
% mt -f /dev/nrmt8 rew;  cat /dev/nrmt8
```

For a cassette drive:

```
% mt -f /dev/nrst0 rew;  cat /dev/nrst0
```

This should cause a TOC file to be listed similar to the following, except for the file names which will vary depending upon what type of distribution you have. The example below is for a distribution of Sun/IRAF for SunOS-4, with the f68881, ffpa, and sparc binaries.

```
0   Table of Contents

1   AS.SOS4.GEN   36.2Mb   IRAF, NOAO packages and Sun/OS4 sources
2   IB.SOS4.F68    8.5Mb   f68881 binaries for IRAF system - Sun/OS4
3   IB.SOS4.FPA    9.1Mb   ffpa binaries for IRAF system - Sun/OS4
4   IB.SOS4.SPC   11.3Mb   sparc binaries for IRAF system - Sun/OS4
5   NB.SOS4.F68    6.9Mb   f68881 binaries for NOAO packages - Sun/OS4
6   NB.SOS4.FPA    7.4Mb   ffpa binaries for NOAO packages - Sun/OS4
7   NB.SOS4.SPC    8.7Mb   sparc binaries for NOAO packages - Sun/OS4
```

Here, the first column is the file number on the tape, the TOC file being file zero, the second column is the name of the tape file, the third column is the file size in megabytes (this tells you how much space will be needed to unpack the file on disk), and the last column is a description of the file contents.

There are three types of tape files in the example shown: the **AS** file, which is all the IRAF sources (core system, NOAO packages, and SunOS-4 host system interface), the **IB** files, or IRAF core system binaries, one for each architecture (machine cpu type or floating point accelerator option), and the **NB** files, or NOAO package binaries. The NOAO package sources are included in the AS file since most people requesting IRAF are expected to want the astronomical reduction software, although IRAF can be configured without these if desired. All of the file objects are UNIX `tar` format files, with the exception of the TOC file which is a simple text file.

In the above example, the `SOS4` in the file names indicates that these files should be installed on a workstation running SunOS-4. A SunOS-3 distribution is indicated by a `SOS3` in the file names, and a 386i distribution is indicated by a `S386`. In principle a given distribution tape may contain any combination of these files.

2.2.2. Installing the main system

To install the main IRAF system, login as iraf, set the current directory to `$iraf`, and read and unpack the `AS` file from the tape, e.g., for a nine track tape, given the example TOC file shown above, where the `AS` file is file 1 on the tape:

```
% mt -f /dev/nrmt8 rew;  mt -f /dev/nrmt8 fsf 1
% tar -xpf /dev/nrmt8
```

In the case of a cartridge tape the block size should also be specified, and the command becomes:

```
% mt -f /dev/nrst0 rew;  mt -f /dev/nrst0 fsf 1
% tar -xpbf 126 /dev/nrst0
```

If the last operation performed on the tape was to read the TOC file the tape will already be positioned to file 1 (which is the `AS` file in our example), and the rewind/forward-skip step can be omitted. After reading and unpacking the tape file the current directory should be listed to verify that the correct tape file was read. If the correct tape file was read, the tape file name (e.g., `AS.SOS4.GEN`) will appear as a zero length file in the current directory after the unpack operation.

After either of the above tar file read operations, the tape is left positioned to *just before the EOF of the file just read*, since `tar` stops reading the file data before reading the physical EOF. Hence, an `mt fsf` will be required to position to the next file on the tape. Any combination of `fsf` (forward skip file) or `bsf` (backward skip file) operations may be used to position to a file on a 9 track tape. On a cartridge tape, it is best to plan things so that only forward file skips are used, using a rewind and forward skip if it is necessary to position to an earlier file on the tape.

Once the main system, containing only sources, is installed it is possible to create one or more empty BIN directories for the executables, then compile and link the full system. More commonly one will merely read the precompiled executables off the distribution tape, as we discuss in the next section.

2.2.3. Configuring the BIN directories

In Sun/IRAF V2.8 and subsequent versions, all binaries (object files, object libraries, and executables) for a software product are contained in a single directory, the so-called BIN directory. Each supported architecture, e.g., sparc, 386i, Sun-3 f68881, etc., has its own BIN directory containing all the binaries for that architecture. To be able to run IRAF, or use IRAF for software development, on a machine of a particular architecture one must first configure the BIN for that architecture. Often a single copy of IRAF will be installed on a server and NFS mounted on client nodes of various architectures, and it is the use of multiple BIN directories which makes such sharing of a single copy of IRAF possible. Each major IRAF based software product, e.g., the IRAF core system or an external layered product such as the NOAO packages, STSDAS, LOCAL, etc., has its own independent set of BIN directories.

Continuing with our example, suppose we want to configure IRAF and the NOAO packages with the sparc binaries so that we can run the system on a Sun-4 or SPARCstation. Two BIN directories will be required, one for the core system, and one for the NOAO packages. If we later decide to add support for another architecture two more BIN directories would be required, and so on for each supported architecture. Since a BIN can be fairly large it is common to locate the BIN directory somewhere outside the IRAF directory tree, to provide maximum flexibility in allocating the remaining free space in the available disk partitions.

Having decided upon the architecture to be configured and where the BIN will be located, two steps are required to configure the BIN:

[1] Create the BIN directory and read the binaries from the distribution tape into the new directory.

[2] In the root directory of the package tree to which the BIN is being added (e.g., `$iraf` for the core system or `$iraf/noao` for the NOAO packages), create a symbolic link `bin.`*arch* pointing to the BIN, where *arch* is the name of the architecture, e.g., `f68881`, `ffpa`, `sparc`, or `i386` (*not* `386i`). If the directory created in step [1] is already located in the package root directory and has the right name, no link is required and this step can be skipped.

In our example, let's assume we have a directory `/u3` with sufficient space for our 11 Mb sparc BIN. Assuming the sparc BIN is file 4 on the 9 track distribution tape and we have just unpacked tar file 1, leaving the tape positioned to just before file 2, the following commands would suffice to read the BIN (tape file `IB.SOS4.SPC`) onto disk:

```
% mkdir /u3/bin.sparc
% cd /u3/bin.sparc
% mt -f /dev/nrmt8 fsf 3
% tar -xpf /dev/nrmt8
```

Alternatively we could have rewound the tape and done an `fsf 4` to get to file 4 on the tape. The next step is to tell IRAF where the new BIN directory is:

```
% cd $iraf
% rm bin.sparc                       # remove old link, if any
% ln -s /u3/bin.sparc bin.sparc
```

The procedure for configuring the sparc BIN for the NOAO packages (or any other layered product) is very similar. In that case we might create a BIN directory `/u3/noao.bin.sparc` and link it to a `bin.sparc` in the NOAO package root directory (normally `$iraf/noao`).

If we were to continue on in this vein and configure the IRAF core system for use with the `sparc`, `f68881`, and `ffpa` architectures, the IRAF root directory might look as follows when we were done.

```
HS.SOS4.GEN   bin.ffpa      doc         mkpkg       unix
IS.PORT.GEN   bin.generic   lib         noao
bin           bin.sparc     local       pkg
bin.f68881    dev           math        sys
```

Here, `bin` and `bin.generic` are preexisting files which should be left alone, and the remaining `bin.` files are links to external BIN directories added by the IRAF site manager.

2.2.4. Deleting unused HSI binaries

Unlike the main IRAF system and external packages like NOAO, the host system interface (HSI) comes with its binaries pre-installed. The `SOS3` (SunOS-3) and `S386` (Sun 386i) HSIs each come with only a single set of HSI binaries which you will surely need if you are using one of these systems. The `SOS4` (SunOS-4) HSI, on the other hand, comes with pre-installed `sparc` and `mc68020` (Sun-3) HSI binaries, each of which is about 1.3-1.5 Mb in size. These binaries are stored in the obvious `bin` subdirectories in `$iraf/unix`.

If you will not be needing either of these HSI BINs (because you don't have both Sun-3 and Sun-4 machines) you may wish to delete the contents of one or the other of these directories to save disk space. For example, if IRAF is installed on a standalone Sun-3 system only the binaries in `bin.mc68020` will be needed, and the contents of `$iraf/unix/bin.sparc` can be deleted.

2.2.5. Network software distributions

Although most IRAF installations or updates will be made from a distribution tape, it is also possible to install IRAF from compressed disk tar files acquired via FTP from the IRAF network archive. The procedure followed is very similar to installing IRAF from a tape, except that the file objects are stored in the FTP archive rather than on tape, and the content of the distribution files is slightly different. Most significantly, the binaries are omitted hence if a network installation is attempted it will be necessary to recompile the full system.

The main difference between the distribution tape and the network archive is that the `AS` (all sources) file object is gone, being replaced by the following files, which one has to manually combine to produce the equivalent of the `AS`.

`HS.`*sys*`.GEN`	The host system interface (HSI) for a particular host system.
`IS.PORT.GEN`	The IRAF core system sources (for any system).
`NS.PORT.GEN`	The NOAO package sources (for any system).

The `IS` and `NS` (core system and NOAO sources) are portable and may be combined with the `HS` for any host machine to produce an IRAF for that host. For example, to configure IRAF for a Sun-3 or Sun-4 under SunOS-4, one would use `HS.SOS4.GEN`. To configure both the `SOS4` and `S386` (386i) systems one would fetch the former three files plus the 386i HSI, `HS.S386.GEN`. These files are stored in the IRAF network archive in compressed form, hence the actual file names will have a `.Z` appended and will have to be uncompressed with the UNIX program `uncompress` before being unpacked with `tar`.

To build, for example, the `SOS4` IRAF using the network file objects, starting from an empty root directory belonging to IRAF, with the compressed archive files stored in `/tmp`:

```
% cd $iraf
% uncompress < /tmp/IS.PORT.GEN.Z | tar -xpf -
% uncompress < /tmp/HS.SOS4.GEN.Z | tar -xpf -
% mkdir noao; cd noao
% uncompress < /tmp/NS.PORT.GEN.Z | tar -xpf -
```

After running the INSTALL script to configure the programming environment (see §2.4), one might then configure, e.g., an empty `sparc` BIN directory in the usual fashion, then enter

```
% cd $iraf
% mkpkg sparc
% setarch sparc
% mkpkg >& spool
```

which would configure the core system for the `sparc` architecture and compile all the binaries (`setarch` is an alias defined in the `.login` for the iraf account). To do the same for the NOAO packages, one would configure the empty bin and then compile the system as follows:

```
% cd $iraf/noao
% mkpkg sparc
% setarch sparc
% mkpkg -p noao >& spool
```

As we see, the commands are the same except for the root directory and the additional argument required to tell mkpkg the name of the non-core system package being compiled. To prepare BINs for other architectures one would repeat the sequence of creating an empty BIN and doing a `mkpkg` to compile the system for each architecture.

At the present time, anyone wishing to access files from the IRAF network archive should first contact the IRAF group to determine the status of the archive and how to access it. In addition to the standard release products, various updates, bug fixes, and add-on packages may be retrieved from the archive without having to wait for a major release of the full system.

2.3. Merge local revisions back into the new system

If this is a new IRAF installation this step can be skipped. Otherwise, once the new system has been restored to disk any local revisions made to the previous IRAF installation should be merged back into the new system. See §2.1.1 for a list of the files most likely to be affected. When propagating revisions made to these files, be sure to not replace the entire file with your saved version, as the version of the file in the new release of IRAF will often contain important additions or changes which must be preserved. It is best to merge your revisions into the version of the file which comes with the new IRAF.† This task will be easier if the revisions have been localized as far as possible, e.g., keep all `termcap` additions together at the head of the file, so that they may merely be transferred to the new file with the editor. The task of propagating revisions will also be much easier if detailed notes have been kept of all revisions made since the the last release was installed.

Beginning with IRAF version 2.8, one should no longer install locally added software in the core system LOCAL package. This significantly complicates updates and is no longer necessary as, due to the layered software enhancements introduced in V2.8 IRAF, it is now straightforward for each site to maintain their own custom LOCAL package external to the core IRAF system. The core system LOCAL is now only a **template-local** to be copied and used as the starting point for a custom LOCAL. The layered software enhancements, and the procedure for building a custom LOCAL, are discussed further in the *Sun/IRAF Site Manager's Guide*.

---

†The UNIX utility `diff` is useful for comparing files to see what has changed.

2.4. Run the INSTALL Script

Once all of the IRAF files have been restored to disk the Sun/IRAF installation script must be run to complete the system installation. The install script modifies the system as necessary to reflect the new root directory and new default image storage and local bin directories, checks the mode and ownership of a number of files, installs a small set of IRAF commands in UNIX, and so on.

To make a trial run of the install script, enter the following commands:

```
% setenv iraf /path/iraf/
% cd $iraf/unix/hlib
% source irafuser.csh
% ./install -n
```

and answer the questions. The "-n" argument tells install to go through the motions without actually doing anything, so that one can see what will be done before committing to it.

Following one or more trial "no execute" ("-n") runs, the install script should be run without the "-n" to complete the installation. This must be done by the superuser as superuser permission is required to carry out the necessary additions to UNIX. Note that it is necessary to run the install script separately on each node from which IRAF will be used. Also, `/usr`, which is normally mounted read-only on a 386i, must be mounted read-write before doing the installation to allow an entry to be made in `/usr/include`. Once the installation is complete the default mount access mode may be restored.

The exchange with the install script will be along the lines of the following (this example is for a 386i):

```
% ./install -n
new iraf root directory (/files/vol/iraf):
default root image storage directory (/tmp3/iraf):
local unix commands directory (/usr/local/bin):
install iraf for machine type i386
old iraf root = /files/vol/iraf, old imdir = /tmp3/iraf
installing iraf at /files/vol/iraf, imdir=/tmp3/iraf, lbindir=/usr/local/bin
proceed with installation? (yes):
```

The "iraf root directory" is the value of `$iraf`. The "root image storage directory" is the default place to put image data for users; the program may prompt with `/tmp` if it cannot find any likely looking data storage areas on your system, but `/tmp` is not a good place to put image data as the contents are deleted whenever the system reboots. The value entered should be the path to a public iraf subdirectory of a designated data or scratch disk on your system. Lastly, the "local unix command directory" is where the UNIX callable IRAF startup commands will be defined. This should be a UNIX directory which is in the default path of anyone who might want to use IRAF; `/usr/local/bin` is the most common value.

After answering with "yes" or hitting return in response to the "proceed with installation" query, the script will issue a series of messages as it checks the system and performs the installation, possibly answering additional questions in the process.

3. System Checkout

The basic IRAF system should be usable once the files have been restored to disk, the binaries have been configured or generated, and the install script has been run. To verify that the basic system comes up and runs successfully, login as iraf and startup the CL (command language) from the iraf account.

Before starting up IRAF from the iraf account, following the installation of a new system, it will be necessary to execute the `mkiraf` task. This will (optionally) initialize the `uparm`

directory for the iraf account, and create a new `login.cl` file. It may also be necessary to edit the `.login` file to modify the way the environment variable `IRAFARCH` is defined. This variable, required for software development but optional for merely using IRAF, must be set to the name of the desired machine architecture, e.g., `sparc`, `f68881`, etc. If it is set to the name of an architecture for which there are no binaries, e.g., `generic`, the CL will not run.

```
% mkiraf
Initialize uparm? (y|n):
Terminal types: gterm=ttysw+graphics,vt640=(vt100+retrographics),etc.
Enter terminal type: gterm
A new LOGIN.CL file has been created in the current directory.
You may wish to review and edit this file to change the defaults.
```

The CL should then run. For Sun/IRAF the CL is normally run from a GTERM window under SunView, e.g.:

```
% sun                    # startup SunView
% cl                     # startup IRAF (in GTERM window)
```

This should startup the CL, which will clear the screen and print out a startup message. The alias *sun* will startup SunView with the text, graphics, and image display windows normally used to run IRAF on a Sun workstation. The standard test procedure included in Volume 1A of the *IRAF User Handbook* should be run to verify the installation.

Appendix A. A Complete Example

Assume we are installing IRAF for the first time on a Sun-3. The IRAF root will be `/usr/iraf`. We will configure both the f68881 and ffpa binaries, locating the BIN directories in the directory `/u2`. The local user commands will be placed in `/usr/local/bin`. We will be reading a 9 track tape distribution from device `/dev/nrmt8`.

The first step is for the superuser to create an account for the fictitious user '`iraf`', with home directory `/usr/iraf/local` and shell `/bin/csh`. We then login as iraf (a warning message will be printed since there is no login directory) and proceed as follows:

```
% setenv iraf /usr/iraf/
% source $iraf/unix/hlib/irafuser.csh
% cd $iraf
%
% whoami
iraf
%
% mt -f /dev/nrmt8 rew
% cat /dev/nrmt8
0  Table of Contents
1  AS.SOS4.GEN  36.2Mb  IRAF, NOAO packages and Sun/OS4 sources
2  IB.SOS4.F68   8.5Mb  f68881 binaries for IRAF system - Sun/OS4
3  IB.SOS4.FPA   9.1Mb  ffpa binaries for IRAF system - Sun/OS4
4  NB.SOS4.F68   6.9Mb  f68881 binaries for NOAO packages - Sun/OS4
5  NB.SOS4.FPA   7.4Mb  ffpa binaries for NOAO packages - Sun/OS4
%
% tar -xpf /dev/nrmt8           # unpack AS.SOS4.GEN
%
% mkdir /u2/bin.f68881
% cd /u2/bin.f68881
% mt -f /dev/nrmt8 fsf
% tar -xpf /dev/nrmt8           # unpack IB.SOS4.F68
%
% mkdir /u2/bin.ffpa
% cd /u2/bin.ffpa
% mt -f /dev/nrmt8 fsf
% tar -xpf /dev/nrmt8           # unpack IB.SOS4.FPA
%
% mkdir /u2/noao.bin.f68881
% cd /u2/noao.bin.f68881
% mt -f /dev/nrmt8 fsf
% tar -xpf /dev/nrmt8           # unpack NB.SOS4.F68
%
% mkdir /u2/noao.bin.ffpa
% cd /u2/noao.bin.ffpa
% mt -f /dev/nrmt8 fsf
% tar -xpf /dev/nrmt8           # unpack NB.SOS4.FPA
%
% cd $iraf                      # install IRAF BINs
% rm bin.f68881
% ln -s /u2/bin.f68881 bin.f68881
% rm bin.ffpa
% ln -s /u2/bin.ffpa bin.ffpa
%
% cd $iraf/noao                 # install NOAO BINs
% rm bin.f68881
% ln -s /u2/noao.bin.f68881 bin.f68881
% rm bin.ffpa
% ln -s /u2/noao.bin.ffpa bin.ffpa
%
% cd $iraf/unix/hlib            # run the INSTALL script
% ./install -n
% su
# ./install
# exit
%
% cd
% source .login                # read .login now that we have one
% rehash                       # pick up new iraf commands
% mkiraf                       # initialize iraf environment
%
% setarch f68881               # run f68881 version of system
% sun                          # fire up SunView
% cl                           # verify that the CL runs
```