Contents

1. **Introduction** 1

2. **Installing DECstation/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 3
     - 2.2.1. Distribution tape format 4
     - 2.2.2. Installing the main system 4
     - 2.2.3. Configuring the BIN directories 5
       - 2.2.3.1. BIN directories under the IRAF root. 5
       - 2.2.3.2. BIN directories outside the IRAF root. 6
     - 2.2.4. Network software distributions 6
   - 2.3. Merge local revisions back into the new system 7
   - 2.4. Run the INSTALL Script 8

3. **System Checkout** 8

Appendix A. A Complete Example 10

DECstation/IRAF Installation Guide

Doug Tody

Central Computer Services
National Optical Astronomy Observatories†
July 1989
(revised May 1990)

ABSTRACT

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

1. Introduction

Before installing DECstation/IRAF, one must 1) obtain an appropriate DECstation/IRAF distribution from the IRAF project, 2) select the machine on which the system is to be installed, and arrange for sufficient disk space to hold the system (approximately 65 Mb for DECstation/IRAF), 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.

This Installation Guide is intended primarily for sites installing IRAF on a Decstation running Ultrix, although it may be used as a general guide when doing ports to BSD-like systems. Other popular UNIX systems for which IRAF is available, e.g. SunOS and Ultrix, have their own system specific IRAF installation guides.

The device and system configuration tables in the standard IRAF distribution come configured for the NOAO systems on which the distribution tapes were made, and will have to be modified once the system is installed. These modifications are discussed in detail in the companion document *UNIX/IRAF Site Manager's Guide*. To simplify the installation process as well as future upgrades, we have tried to isolate the site dependent files to the minimum number of directories, i.e., `dev`, `hlib` (a subdirectory of `$iraf/unix`), and `local`. The remainder of the system should not require any modifications.

In IRAF V2.8 local additions to the system are no longer installed directly in IRAF; the *layered software* enhancements allow maintainance of a custom LOCAL package outside IRAF. Sites which had previously maintained their own IRAF software in `$iraf/local` will have a one-time conversion job, discussed in the Site Manager's Guide; in future releases only a single file will need to be edited to reinstall the local additions.

IRAF HOTLINE

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

telephone	`(602) 323-4160`	
internet	`iraf@noao.edu`	
span/hepnet	`noao::iraf`	(noao = 5355)
uucp	`{arizona,decvax,ncar}!noao!iraf` or	
uucp	`uunet!noao.edu!iraf`	
bitnet	`iraf@noao.edu` (through a gateway)	

Issues such as interfacing new graphics terminals, plotters, or image displays are also described in the Site Manager's Guide. Help is available via the IRAF Hotline if any problems should arise while installing the system or interfacing new devices.

2. Installing DECstation/IRAF

Installing DECstation/IRAF on an actual DECstation system is straightforward. 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), 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. A complete example for the simplest type of installation is provided in Appendix A.

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 *
```

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 as described above 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 most BSD 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.

It is not necessary to place the entire system on the same disk; the binary files and external packages like `noao` may be located on a separate disk from the core system if desired; see §2.2.1 for the sizes of the different components.

---

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

Do not worry about configuring the `.login` or other 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 might be, for example, a 1600† or 6250 bpi 9 track tape, or a TK50 cartridge tape.

If you are installing IRAF on a system which has a local tape drive you can skip what follows and 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. If you have NFS, do not use it 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 as in §2.2.1, then

```
% 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 execute `dd` remotely 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) 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 name required for your site may vary from that shown):

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

This should cause a TOC file similar to the following to be listed for a normal DECstation distribution; the sizes of the files will change for different releases:

```
0    Table of Contents
1    AS.DSUX.GEN     44.9Mb   IRAF, NOAO packages and DECstation sources
2    IB.DSUX.MIP     21.0Mb   IRAF system binaries for DECstation
3    NB.DSUX.MIP     18.0Mb   NOAO packages binaries for DECstation
```

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 DSUX host system interface), the **IB** file, or IRAF core system binaries, and the **NB** file, or NOAO binaries. The NOAO sources are

†Distributions for a 1600 bpi tape require two tapes. Instructions for reading in IRAF from two tapes are contained in a separate cover letter.

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.

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
```

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.DSUX.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.

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

The executables for a software product such as the IRAF core system or the NOAO packages are contained in a single directory, the so-called BIN directory. In some cases the system object files and libraries may also reside in the BIN, e.g., to support software development for multiple architectures. In the rest of this document, we will assume a single IRAF system, supporting only the DSUX architecture both for the runnable system and for software development. A system configured for multiple architecture support will have multiple BIN directories, one for each architecture. Further information on multiple architecture support is given in the *UNIX/IRAF Site Manager's Guide*.

For the default DSUX-only configuration, two BIN directories will be required: one for the core system, and one for the NOAO packages. Since a BIN can be fairly large one may want 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. A BIN may be located either in the root directory of the system to which it belongs, or in an external directory, replacing the entry in the package root directory by a symbolic link. The procedures for configuring the BINs in each case are outlined below.

2.2.3.1. BIN directories under the IRAF root.

Go to §2.2.3.2 if you want the binaries outside the IRAF root. If the executable files are to reside in the same directory tree as the rest of IRAF, it is necessary first to remove the `bin.mips` link just created when the `AS` file was read in, create a new `bin.mips` subdirectory, enter it and read the BIN contents from tape. Assuming the mips BIN is file 2 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.DSUX.MIP`) onto disk:

```
% cd $iraf
% rm bin.mips                    # remove old symbolic link
% mkdir bin.mips                 # create real bin directory
% cd bin.mips
% mt -f /dev/nrmt8 fsf 1         # position tape to IB.DSUX.MIP
% tar -xpf /dev/nrmt8            # read iraf binaries
```

Alternatively we could have rewound the tape and done an `fsf 2` to get to tape file 2. Now read in the NOAO binaries.

```
% cd $iraf/noao
% rm bin.mips                    # remove old symbolic link
% mkdir bin.mips                 # create real bin directory
% cd bin.mips
% mt -f /dev/nrmt8 fsf 1         # position tape to NB.DSUX.MIP
% tar -xpf /dev/nrmt8            # read iraf binaries
```

You are now finished reading the tape and may proceed to §2.3 (or 2.4 for a new IRAF installation).

2.2.3.2. BIN directories outside the IRAF root.

Let's assume we have a directory `/u3` with sufficient space for our 10 Mb mips BIN. Assuming the mips BIN is file 2 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.DSUX.MIP`) onto disk:

```
% mkdir /u3/bin.mips
% cd /u3/bin.mips
% mt -f /dev/nrmt8 fsf 1
% tar -xpf /dev/nrmt8
```

Alternatively we could have rewound the tape and done an `fsf 2` to get to tape file 2. Now read in the NOAO binaries.

```
% mkdir /u3/noao.bin.mips
% cd /u3/noao.bin.mips
% mt -f /dev/nrmt8 fsf 1
% tar -xpf /dev/nrmt8
```

The next step is to tell IRAF where the new BIN directories are:

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

2.2.4. 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 outside the HSI 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.DSUX.GEN	The host system interface (HSI) for DECstation Ultrix, including the HSI binaries.
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. 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 DSUX 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.DSUX.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 should then configure an empty mips BIN directory and start a SYSGEN. This should be done from the IRAF account. The following assumes that the BIN is to be placed in a subdirectory rather than being a link to a remote directory (see §2.2.3).

```
% cd $iraf
% mkdir bin.mips
% ln -s bin.mips bin
% mkpkg >& spool
```

This would compile all the binaries. To do the same for the NOAO packages, one could configure the empty bin and then compile the system as follows.

```
% cd $iraf/noao
% mkdir bin.mips
% ln -s bin.mips bin
% 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.

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 not to 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 its 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 *UNIX/IRAF Site Manager's Guide.*

2.4. Run the INSTALL Script

Once all of the IRAF files have been restored to disk the IRAF install script (`hlib/install`) 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.

The exchange with the install script will be along the lines of the following:

```
% ./install -n
new iraf root directory (/iraf/iraf): /usr/iraf
default root image storage directory (/tmp2/iraf):
local unix commands directory (/local/bin): /usr/local/bin
install iraf for machine type mips
old iraf root = /iraf/iraf, old imdir = /tmp2/iraf
installing iraf at /usr/iraf, imdir=/tmp2/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.

---

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

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.

```
% mkiraf
Initialize uparm? (y|n):
Terminal types: gterm=ttysw+graphics,vt640=(vt100+retrographics),etc.
Enter terminal type: vt640
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 default terminal type option (`vt640` in the example) is very site dependent and you will probably want to enter a different value from that shown (e.g., "xterm"). Look in the file `$iraf/dev/termcap` to see what terminals are supported. Instructions for interfacing new terminals are given in the *Site Manager's Guide.* The `stty` task may be used to display or change the terminal type after logging into the CL. A graphics terminal must be specified to be able to run IRAF tasks which use interactive graphics.

Once `mkiraf` has been run to initialize the IRAF environment the CL may be started, e.g.:

```
% cl                          # startup IRAF
```

This should startup the CL, which will clear the screen and print out a startup message. 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

Here we present a complete sample IRAF installation for a DECstation running DEC Ultrix. This is the simplest possible installation, i.e. it is a new installation, the BIN directories are located under the IRAF root, and the archives are read sequentially off a 9 track distribution tape from device `/dev/nrmt8` (the actual device is bound to be different).

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`. The directory `/usr/iraf` should exist, but that is all that is needed. 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/
% cd $iraf
%
% whoami
iraf
%
% mt -f /dev/nrmt8 rew
% cat /dev/nrmt8

0  Table of Contents
1  AS.DSUX.GEN     44.9Mb   IRAF, NOAO packages and DECstation sources
2  IB.DSUX.MIP     21.0Mb   IRAF system binaries for DECstation
3  NB.DSUX.MIP     18.0Mb   NOAO packages binaries for DECstation

%
% tar -xpf /dev/nrmt8           # unpack AS.DSUX.GEN
%
% rm bin.mips                   # remove old symbolic link
% mkdir bin.mips                # create actual directory
% cd bin.mips
% mt -f /dev/nrmt8 fsf
% tar -xpf /dev/nrmt8           # unpack IB.DSUX.MIP
%
% cd $iraf/noao
% rm bin.mips                   # remove old symbolic link
% mkdir bin.mips                # create actual directory
% cd bin.mips
% mt -f /dev/nrmt8 fsf
% tar -xpf /dev/nrmt8           # unpack NB.DSUX.MIP
%
% cd $iraf/unix/hlib            # run the INSTALL script
% source irafuser.csh           # pick up environment defs for install
% ./install -n
% su
# ./install
# exit
%
% cd; pwd
/usr/iraf/local
% source .login                # read .login now that we have one
% rehash                       # pick up new iraf commands
% mkiraf                       # initialize iraf environment
%
% cl                           # verify that the CL runs
```