Digital Unix IRAF Installation Guide

Doug Tody

Mike Fitzpatrick

IRAF Group

Revised November 1997

ABSTRACT

This document describes how to install or update IRAF on a DEC Alpha running Digital Unix V4.x (OSF1 V4.x). Both standalone and network installations are described. Only those issues which one must understand to install DUNX/IRAF are discussed here; a companion document, the *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.

December 27, 1997

Contents

Digital Unix IRAF Installation Guide

Doug Tody

Mike Fitzpatrick

IRAF Group

Revised November 1997

1. Introduction

Before installing DUNX/IRAF on a DEC Alpha system, one must 1) obtain an appropriate DUNX/IRAF distribution, e.g., from the IRAF network archive on `iraf.noao.edu` (or one of it's mirror sites, or by ordering a mailed (tape or CD-ROM) distribution from NOAO), 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 the basic installation should only take a half hour or so. Additional time may be required to customize the system, to configure the local tape drives and other devices, set up IRAF networking, and so on.

This Installation Guide is intended for sites installing IRAF on a DEC Alpha system running Digital Unix V4.0 or later. The system is expected to work on OSF1 V3 platforms although there may be problems compiling external packages or local IMFORT programs; sites are urged to upgrade if at all possible. Other popular UNIX systems for which IRAF is available, e.g. AIX, HPUX, IRIX, SunOS, Solaris, etc. have their own system specific IRAF installation guides. There are also separate installation guides for the DECstation or VAXstation running Ultrix. The IRAF distribution for the DECstation is DSUX and that for the VAXstation is VXUX. The VMS/IRAF distribution is known as VMS5 and is intended for VAX hardware (an OpenVMS V2.11 version of IRAF for the DEC Alpha is still pending as of this writing). The DEC Alpha Digital Unix IRAF distribution described here is known as DUNX (Digital Unix) in the IRAF archives.

2. Installing DUNX/IRAF

Although the details of how DUNX/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 the distribution files, 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.

System managers familiar with the installation of typical UNIX programs should beware that IRAF, being a large system in its own right and not a UNIX program, does not always follow the usual conventions. It is wise to read and adhere to the installation instructions to avoid problems.

```
# Prepare the root IRAF directory.
if new installation
    create iraf account
else if updating an old installation
    save locally modified files; delete old system

# Install the files.
login as iraf
unpack the core system distribution
configure the BIN directories

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

run the iraf install script to complete the installation
checkout the new system
```

If problems should arise during the installation help is available by posting a query to one of the *adass* newsgroups on USENET, by sending email to `iraf@noao.edu`, or via the IRAF HOTLINE (520-318-8160).

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 something like the following.

```
% cd $iraf†
% tar -cf /scr0/oiraf.tar local dev unix/hlib
% /bin/rm -rf *
```

There are many possible variations on this, e.g., you could use *mv* to move the above directories to someplace outside the main IRAF directory tree. Although it is probably simplest and safest to save entire directories as in the example, in practice only a few files are likely to have been modified. These are the following.

```
dev/graphcap
dev/hosts
dev/tapecap
dev/termcap
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. It is important to delete the old system first to avoid creating junk files or directories when the new system is installed (due to file or directory name changes or deletions). Once the new system has been restored to disk, do *not* merely restore the files saved above, as you will need to carefully merge local changes into the versions of the files supplied with the new IRAF release (more on this later).

† ↵$iraf symbolizes the UNIX pathname of the root IRAF directory. If no "iraf" environment variable is defined just supply the actual pathname.

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 user '`iraf`'. This is necessary for IRAF system management, which should always be done from the IRAF account. The IRAF account has special login files which set up a custom UNIX environment for IRAF system management. Having an IRAF account provides a convenient place (the IRAF system manager's login directory) to keep scratch files created during system configuration.

The location of the IRAF root directory is arbitrary. Our practice here is to locate the software in a system file storage area separate from the operating system files (to simplify Digital Unix upgrades), and then use a symbolic link such as /iraf to point to the actual root directory. This makes life simpler if IRAF is NFS mounted on several machines and it is later necessary to move the IRAF files. Try to keep the path to the physical IRAF root directory short to avoid filename truncation problems when IRAF is run.

In whichever directory you choose to install IRAF you will need to create several directories into which you will unpack the IRAF distribution files. For simplicity we will assume that you use /iraf (or use /iraf as a symbolic link to some other directory), if so you need to create an *iraf* directory for the source tree (the `as.dunx.gen` files discussed below), and an *irafbin* directory for the core system and NOAO package binaries. Within *irafbin* create a *bin.alpha* directory for the core binaries (the `ib.dunx.alp` distribution file) and a *noao.bin.alpha* directory for the NOAO package binaries (the `nb.dunx.alp` distribution file). Using this suggested directory structure the IRAF directory tree should look something like:

```
        /iraf
         /  \
        /    \
    /iraf   /irafbin
              /   \
             /     \
       bin.alpha  noao.bin.alpha
```

Note that with this setup, the *iraf root directory*, i.e. $iraf, is then /iraf/iraf.

The login directory for the iraf account should be $iraf/local (e.g., /iraf/iraf/local, which will be created when unpacking the sources), rather than the IRAF root directory $iraf as one might expect. This is done to provide a work area for local files separate from the main IRAF directory tree, to simplify updates and make it easier to keep track of what has been locally added and what is standard IRAF. In any case, make sure that when the IRAF account is set up the login directory is set correctly, or the IRAF environment will not be set up properly, and later problems are sure to result.

A typical IRAF installation consists of the main IRAF release, a number of BIN directories (the IRAF binaries), and additional directories for layered software such as STSDAS, PROS, and so on. If sufficient disk space is available to keep everything in one area the following directory structure is recommended.

```
/iraf/iraf                      # iraf root directory ($iraf)
/iraf/iraf/local                # iraf login directory (˜iraf)
/iraf/irafbin                   # iraf BIN directories
/iraf/irafbin/bin.alpha         # DEC Alpha binaries - core system
/iraf/irafbin/noao.bin.alpha    # DEC Alpha binaries - NOAO packages
/iraf/extern/stsdas             # layered package
/iraf/extern/tables             # layered package
/iraf/extern/xray               # layered package
/iraf/extern/ftools             # layered package
  (etc.)
```

For the purpose of this example we assume that the IRAF files are stored in /iraf; as we say this might be a link and the actual directory is arbitrary. Given this directory the IRAF root $iraf would be "/iraf/iraf/" and the login directory for the IRAF account would be /iraf/iraf/local. The alpha (Digital Unix) binaries for the core IRAF system would be in /iraf/irafbin/bin.alpha, with a link $iraf/bin.alpha pointing to this directory (more on this later).

Given the above directory structure the `passwd` file entry for the IRAF account would be something like the following.

```
iraf:abcdef:312:112:IRAF system login:/iraf/iraf/local:/bin/csh
```

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 are installing from tape skip forward to §2.2.2. If you are installing from a network distribution (i.e., from disk) continue with the next section.

2.2.1. Installing from a network distribution

DUNX/IRAF is available over the network via anonymous ftp from the node `iraf.noao.edu`, in the subdirectory `iraf/v`*nnn*`/DUNX`, where "`v`*nnn*" is the IRAF version number, e.g., subdirectory `iraf/v211/DUNX` for V2.11 DUNX/IRAF.

If IRAF is being installed from a network distribution all the architecture independent IRAF files for both the core IRAF system and the NOAO packages will be in the distribution file `as.dunx.gen`. This "file" is stored in the network archive as a directory wherein the large distribution file has been split into a number of smaller pieces, e.g.,

```
% ls as.dunx.gen
CHECKSUMS           as.dunx.gen.Z.12    as.dunx.gen.Z.26
FILES.Z             as.dunx.gen.Z.13    as.dunx.gen.Z.27
as.dunx.gen.Z.00    as.dunx.gen.Z.14    as.dunx.gen.Z.28
as.dunx.gen.Z.01    as.dunx.gen.Z.15    as.dunx.gen.Z.29
as.dunx.gen.Z.02    as.dunx.gen.Z.16    as.dunx.gen.Z.30
as.dunx.gen.Z.03    as.dunx.gen.Z.17    as.dunx.gen.Z.31
    (etc.)
```

Assume that the directory `as.dunx.gen` as shown above has been recreated somewhere on the machine on which IRAF is to be installed. We can restore the main IRAF source tree as follows.

```
% whoami
iraf
% cd $iraf
% cat /path/as.dunx.gen/as.* | uncompress | tar -xf -
```

After the above finishes the root IRAF directory should appear as follows (this is for V2.11).

```
HS.DUNX.GEN  bin.alpha    doc          math         pkg          unix
IS.PORT.GEN  bin.generic  lib          mkpkg        sys
bin          dev          local        noao         tags
```

The file `bin.alpha` is a link to the IRAF BIN directory for the "alpha" architecture (for binary executables), which probably does not contain anything yet. Configuring the BIN directories is discussed in section §2.2.3.

2.2.2. Installing from tape

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 may be a DAT tape, an Exabyte, or whatever.

IRAF 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 actual device name will likely be different than that shown in the examples).

```
% mt -f /dev/nrmt0h rew;  cat /dev/nrmt0h
```

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 actual file sizes may differ from what is shown). The example below is for a distribution of DUNX/IRAF for the DEC Alpha, including all available binaries. The file sizes are only examples in this case.

```
0   Table of Contents

1   AS.DUNX.GEN   52.2Mb   All sources, includes HSI binaries
2   IB.DUNX.ALP   30.3Mb   IRAF binaries (core system, DU4 compilers)
3   NB.DUNX.ALP   18.9Mb   NOAO binaries (NOAO packages, DU4 compilers)
```

Here, the first column is the file number on the tape, the TOC file being file zero (the first distribution file is number one), the second column is the name of the tape file, the third column is the file size in megabytes of the uncompressed distribution file (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 (the core IRAF system, NOAO packages, and the Digital Unix host system interface), the IB files, or IRAF core system binaries, one for each architecture, 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 this 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. The distribution files may be compressed if this was necessary to fit all the files on a tape.

In the above example, the DUNX in the file names indicates that these files are for a DEC Alpha running Digital Unix. A DEC Ultrix distribution is indicated by a DSUX in the file names, a Sun Solaris distribution by SSOL, an IBM AIX distribution by AIX3, and so on. In principle a given distribution tape may contain any combination of these files.

The V2.11 DUNX/IRAF distribution for the DEC Alpha supports a single architecture **alpha**, which is for DEC's 64 bit Alpha processor. The alpha binaries were generated with the DEC compiler V4.1-6 on a Digital Unix V4.0 system.

The following commands would suffice to restore the main IRAF system to disk, given the distribution tape described by the TOC file in our example above. Once again, the tape device file and possibly the block size will likely have to be changed to whatever is needed for the actual tape device used.

```
% whoami
iraf
% cd $iraf
% mt -f /dev/nrmt0h rew;  mt -f /dev/nrmt0h fsf 1
% tar -xf /dev/nrmt0h
```

After the above tar file read operation, 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 some tape drives, it is safest 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 IRAF all the files specific to any particular hardware or software architecture, e.g., for the Alpha chip with the Digital Unix compilers in the case of the DEC Alpha, are contained in a single directory called the BIN, or "binary", directory. To run IRAF you must install not only the AS (all-sources) directory tree, but the BIN directory for each desired architecture. The IRAF core system and the NOAO packages have separate BIN directories.

The BIN directories for the IRAF core system or a layered package (such as NOAO) are located, logically or physically, in the root directory of the IRAF core system or layered package. Every layered package has its own set of BIN directories. In the distributed V2.11 system you will find the following BIN files (directories or symbolic links) at the IRAF root.

```
link         bin -> bin.generic
directory    bin.generic
link         bin.alpha  -> ../irafbin/bin.alpha
link         noao/bin.alpha  -> ../../irafbin/noao.bin.alpha
```

If the IRAF directory structure is set up as described in §2.1.2, with $iraf located at /iraf/iraf and the BIN directories stored in /iraf/irafbin, then these links will not have to be modified. If a different directory structure is used you will have to modify the links accordingly.

The *bin* link and the *bin.generic* directory are required for the correct operation of the IRAF system software (*mkpkg*) and are maintained automatically by the IRAF software management utilities. *Under no circumstances should "bin" or "bin.generic" be modified or deleted.* It is a very common error to manually delete the bin link and manually set it to bin.alpha or some other architecture (usually after noticing that the library links in lib$ point to "../bin"). The bin.alpha link can be modified as desired but bin and bin.generic should be left alone.

Assume that the bin.alpha directory has been created somewhere, e.g. in the /iraf/irafbin directory, and that the `ib.dunx.alp` distribution files for the core IRAF system VAX binaries have been downloaded from the network archive. We can restore the alpha binaries with the following commands.

```
% cd $iraf/bin.alpha
% cat /path/ib.dunx.alp/ib.* | uncompress | tar -xf -
```

Similarly, to restore the NOAO package alpha binaries:

```
% cd $iraf/noao/bin.alpha
% cat /path/nb.dunx.alp/nb.* | uncompress | tar -xf -
```

This process is repeated for each architecture, restoring two BIN directories per architecture.

The procedure for restoring a BIN directory from a tape distribution is similar to that described in §2.2.2 for the core system. For example,

```
% cd $iraf/bin.alpha
% mt -f /dev/nrmt0h rew;  mt -f /dev/nrmt0h fsf 2
% tar -xf /dev/nrmt0h
```

would restore the core system bin.alpha directory from tape unit zero containing an uncompressed `ib.dunx.alp` as file 2 on the tape.

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 `graphcap/termcap/tapecap` additions together at the head of the file, so that they may merely be transferred to the new file with the editor. This approach preserves local additions or overrides while ensuring that any changes made to the body of the standard distributed file are kept. The task of propagating revisions will also be easier if notes have been kept of all revisions made since the the last release was installed. This should be done in the iraf account directory, i.e. $iraf/local.

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

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

only a *template-local* to be copied and used as the starting point for a custom LOCAL layered package. 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 DUNX/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/        # trailing '/' is required
% cd $iraf/unix/hlib
% source irafuser.csh
% ./install -n
```

and answer the questions. The use of "./install" versus just "install" is recommended to be sure the IRAF install script is executed and not the system /usr/bin/install command. 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.

Installing IRAF requires a few changes to be made to system directories outside the IRAF directory tree. Two fifo device entries are made in /dev. A symbolic link "iraf.h" is created in /usr/include. A number of links (cl, mkiraf, etc.) are made in /usr/local/bin or some similar "local bin" directory which most users can be expected to have in their search path. The tape allocation task alloc.e is made suid root (there are no known security loopholes, although we cannot make any guarantees). A symbolic link imtoolrc is created in /usr/local/lib (but only if this directory already exists. Since the imtoolrc file is needed for image display it is a good idea to create this directory prior to running the install to avoid problems later on.)

Following one or more trial "no execute" ("-n") runs to see what the install script will do, 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.

It is necessary to run the install script separately on each node from which IRAF will be used. If a single version of IRAF is installed on a server and NFS mounted on one or more clients, the install script must be run first on the server and then on *each* client (when installing on a client there will be warnings about insufficient permission to make changes to files on the NFS mounted partitions, which can be ignored). Having a /iraf link on each system means the iraf path with be the same on all nodes and will hide any NFS mount points, simplifying the systems all around. To install IRAF on a diskless client it may be necessary to run the install script *on the server* to do the install for the client, since the client's /usr/include and /dev directories may only be writable by root on the server. On some systems /usr is mounted read-only, and must be unmounted and remounted 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.

```
% ./install -n
new iraf root directory (/iraf/iraf):
default root image storage directory (/data0/irafimdirs):
local unix commands directory (/usr/local/bin):
directory for local shared libraries (/usr/local/lib):
install iraf for machine type alpha
old iraf root = /iraf/iraf, old imdir = /data0/irafimdirs
installing iraf at /iraf/iraf, imdir=/data0/irafimdirs,
lbindir=/usr/local/bin
proceed with installation? (yes):
```

The "iraf root directory" is the value of $iraf (minus the trailing '/'in this case). 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. The "local unix command directory" is where the UNIX callable IRAF startup commands (e.g. `cl`, `mkiraf`, etc) 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. Lastly, the "directory for local shared libraries" is where the IRAF shared library, `libiraf.so` is installed. This directory should normally be in the user's LD_LIBRARY_PATH environment path, it will first default to some local lib directory if it exists or will otherwise use a system library directory.

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 (IRAF command language) from the iraf account. You should be able to login as IRAF and type "cl" to start IRAF, using the login files which come with the distributed system.

```
% login iraf
% cl
```

To more thoroughly test the installation it is a good idea to test IRAF from a user account. To do this you login to a user account and run the *mkiraf* task to set up the IRAF login files. This will create or initialize the user's `uparm` (user parameter) directory, and create a new `login.cl` file. It may also be desirable to edit the user's `.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, in our case *alpha*.

```
% mkiraf
Initialize uparm? (y|n): y
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc."
Enter terminal type: xgterm
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* command should now start up 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* (or available from our network archive as `/iraf/docs/testproc.ps.Z`) should be run to verify the installation.

Appendix A. A Complete Example

Assume we are installing IRAF for the first time on a DEC Alpha. The IRAF directories will be located at /u3/iraf using a symbolic link /iraf to point to this location. We will configure only alpha binaries, locating the BIN directories in the directory /iraf/irafbin. The local user commands will be placed in /usr/local/bin. We will be installing from a network distribution with the distribution files located in /scr0.

The first step is for the superuser to create an account for the fictitious user '`iraf`', with home directory /iraf/iraf/local and shell /bin/csh. The /u3/iraf directory is created owned by IRAF, and pointed to by the link /iraf. We then login as IRAF (a warning message will be printed since there is no login directory) and proceed as follows.

```
% whoami
iraf
%
% setenv iraf /iraf/iraf/                # set root directory
% mkdir /iraf/iraf
%
% cd $iraf                               # unpack main IRAF distribution
% cat /scr0/as.dunx.gen/as.* | uncompress | tar -xf -
%
% cd /iraf                               # create BIN directories
% mkdir irafbin
% mkdir irafbin/bin.alpha
% mkdir irafbin/noao.bin.alpha
%
% cd $iraf/bin.alpha                     # unpack core bin.alpha
% cat /scr0/ib.dunx.alp/ib.* | uncompress | tar -xf -
%
% cd $iraf/noao/bin.alpha                # unpack NOAO bin.alpha
% cat /scr0/nb.dunx.alp/nb.* | uncompress | tar -xf -
%
% cd $iraf/unix/hlib                     # run the INSTALL script
% source irafuser.csh
% ./install -n
% su
# ./install
# exit
%
% cd
% source .login                         # read new .login
% rehash                                 # pick up new iraf commands
% cl                                     # verify that the CL runs
```

This will fully install IRAF on a server or a standalone system. If this version of IRAF will be accessed via NFS by client nodes then the IRAF install script must be run on each client node as well. Installing IRAF does not allow one to access local tape drives, printers, and so on. Refer to the *UNIX/IRAF Site Manager's Guide* for information on how to configure IRAF for the local site.