IRAF Version 2.9 Revisions Summary

April 10, 1990

1. Introduction

This document summarizes the changes in IRAF version 2.9. This was primarily a development release intended to support applications software development, hence the major changes were in the programming environment, although there are important new features of interest to general users too. Since IRAF V2.9 is primarily a development release, it is not being released on all platforms, and it is expected that many sites will not need to upgrade until IRAF V2.10 is available. Sites interested in obtaining IRAF V2.9 should contact the IRAF project to determine if the release is available for a particular host system. At the present time, the release is being made available for all Sun systems, for VAX/VMS, and for the DECstation running Ultrix.

What follows is a brief description of some of the new features available in IRAF Version 2.9. This is not intended to be an exhaustive list, but rather a brief summary of the major changes since the last release of IRAF, Version 2.8, released in July 1989. More detailed revisions notes are available in the system notes file, `iraf$local/notes.v29`, as well as in the online revisions notes for the various packages.

Users looking for information on a particular new package should note that if the package is not mentioned in these release notes and therefore is not included in IRAF V2.9, that does not necessarily mean that it is not available. Most major reduction and analysis packages are now made available for testing as user installable layered packages before they are included in the standard distribution. For information on the available add-on packages, contact the IRAF group, or check the latest *IRAF Newsletter*.

This revisions summary is organized as follows:

2. IRAF System Revisions

2.1. IEEE to native floating point conversions

Support has been added to the programming interfaces (§4.2.3) for converting between the IEEE floating point and native floating point data formats, including both single and double precision. The FITS programs in DATAIO (§3.1.1) make use of this, allowing floating point data to be exchanged in FITS format without having to convert to type integer.

2.2. World coordinate system support

A major new VOS interface MWCS has been added to support general world coordinate systems (WCS) and transformations thereon (§4.2.1). This includes support for linear, piecewise linear or sampled WCS, and general nonlinear WCS such as the tangent plane or gnomonic projection.

If a FITS image is read into the system which has WCS information in the header, the WCS will be retained in the IRAF image header and can be used in coordinate transformations. The IMAGES tasks which move pixels around have been modified to edit the WCS to reflect the transformation (§3.1.2). The image i/o system will automatically propagate the WCS of an image to a new copy of the image, and will edit the WCS as necessary if an image section is copied (this applies to all IRAF tasks which operate upon images). The task RIMCURSOR in the LISTS package has been rewritten to add support for coordinate transformations (§3.1.3), and can be used, e.g., to read out the RA and DEC of objects on the image display using the image cursor, if the image has the necessary WCS information in the image header.

Full integration of the new world coordinate facilities into all the IRAF applications, e.g., the graphics tasks and the spectral reduction packages, will take a year or longer due to the amount of software involved. In V2.9 the IRAF spectral packages have not yet been converted to use MWCS, and if MWCS is enabled it could alter the normal behavior of these packages. *IRAF V2.9 is therefore shipped with MWCS disabled*. What "disabled" means is that WCS information in the image headers is not edited to reflect operations involving image sections, or geometric transformations of images. Tasks such as RIMCURSOR which use an already existing WCS will still work whether or not header editing is disabled. If the spectral tasks will not be used and it is desired that world coordinates be propagated correctly in image transformations, MWCS header editing can be enabled in either of the following ways.

The MWCS transformations are disabled by defining the variable "`nomwcs`" in the IRAF environment. To globally enable MWCS by default for everyone using the system, edit the file "`hlib$zzsetenv.def`" and comment out the following line as shown (you want to add the leading #, which will be missing in the distributed version):

```
#set    nomwcs          = yes
```

To enable MWCS header editing temporarily, for the current IRAF run:

```
cl> reset nomwcs = no
```

Detailed information on the coordinate systems defined by MWCS can be obtained in the online system with the command

```
cl> phelp mwcs$MWCS.hlp fi+
```

Additional information is also given in the help page for RIMCURSOR.

2.3. IMFORT changes

The IMFORT interface (host level Fortran or C interface to the IRAF image format) has undergone the following bug fixes and enhancements:

- A couple of bugs associated with the IMDIR (image pixel-file directory) feature introduced in IRAF V2.8 have been fixed.
- Image clobber checking has been added. By default, if you create a new image and another image with the same name already exists, the image create will now return an error code leaving the existing image unchanged. To override clobber checking in IMFORT programs, restoring the previous behavior of the interface, define "`clobber`" in your host environment.

- IMFORT will now perform a limited filename translation service using the IRAF VOS filename translation code. This should allow most IRAF filenames to be used as input to host level IMFORT programs. Full VOS filename mapping is not provided, but filenames containing upper case characters and multiple "." delimited fields should be translated as in IRAF programs.
- On systems with multiple architecture support (e.g., Sun, Convex) the FC task, used to compile and link IMFORT programs from within the IRAF environment, is now a script rather than a simple foreign task front end to XC. The purpose of the script is to see that all the necessary IRAF and host level command line switches and environment definitions (`IRAFARCH`, `FLOAT_OPTION`, etc.) are used. Previously, users had to make these environment definitions manually, and if they forgot the IMFORT program could fail to link or execute.
- On most UNIX/IRAF systems, the host library `-lU77` is now searched automatically by FC when an IMFORT program is linked. This library is not used by any of the IRAF code, but is required to link some Fortran programs that might want to use IMFORT.

Users are encouraged to use FC to link their IMFORT programs. It is possible to manually link against the IRAF libraries if you know what you are doing, but the location of the libraries and the required host level command line switches vary for different systems and for different architectures of a single system, and it is easy to make mistakes.

2.4. MKIRAF now copies login.cl to login.cl.OLD

On UNIX/IRAF systems, the MKIRAF command will now copy any existing `login.cl` file to `login.cl.OLD`, so that, for example, you can more easily merge any custom changes back in after running MKIRAF. On VMS/IRAF systems a new file version is created, as before.

2.5. Local additions to termcap/graphcap

The termcap and graphcap device capability files have been reorganized with a section at the top for local additions. It is recommended that any locally added entries be made in this area, to simplify future system updates. The local additions can then be simply transferred to the new version of the file when a new version of IRAF is installed (any entries which are modified versions of standard entries should always be checked to see if anything has changed in the distributed version).

2.6. BIN directories now smaller

On systems with multiple architecture support, the architecture save file `OBJS.arc` stored in the BIN directory for each architecture is now maintained as a compressed file. In a typical case this reduces the size of the file by about a factor of two, saving 1-2 Mb of disk space in each BIN directory.

2.7. Various system buffers increased in size

The layered software support in IRAF V2.8 (`extern.pkg` and all that) had a problem with very long `helpdb` environment strings, limiting the number of external packages which could be defined. To fix this problem, various buffers were increased in size all over the system. The maximum length of an environment variable such as `helpdb` is now 960 characters (12 80 character lines of text). String parameters to tasks can also be larger, and the system is more resistant to problems when size limits are exceeded. Foreign task commands, OS escapes, etc., can all be larger now. The current limit on such strings is about 1024 characters, and is defined at sysgen time by the new system parameter `SZ_COMMAND` in `hlib$iraf.h`.

2.8. Shared library versions

The Sun/IRAF shared library mechanism was modified to add support for shared library versions. The result is that when you install IRAF V2.9, which has a different shared library than V2.8, any local programs or other layered software linked under V2.8 will continue to run, because both the old V2.8 shared library and the new V2.9 shared library are included in V2.9 (with different version numbers). Although old programs will continue to run with V2.9, it is recommended that they be relinked eventually to take advantage of the many features and bug fixes provided by V2.9. In the case of very large packages, e.g., STSDAS 1.0, it may be wise to wait until the latest release can be obtained and installed before relinking, as the old version will not have been tested under IRAF V2.9 (which of course, didn't exist back then).

2.9. File pager enhancements

The system file pager, used in the PAGE task, the new PHELP task, and other places, has undergone the following enhancements.

- The N and P keys, used to move to the next or previous file when paging a list of files, now have a dual meaning: when paging a *single* file containing multiple formfeed delimited pages, the keys will move to the next or previous *page* in the file. This feature is used in the new PHELP task to page a large file containing, e.g., all the HELP pages for a package.
- A limited upscrolling capability is now supported, e.g., if you hit the 'k' key while in the pager, the screen will be scrolled up one line in the file being paged. This feature may not be supported for some terminals, in which case the entire screen is redrawn at the new file location.

2.10. STF image kernel enhancements

Extensive work has been done on the STF image kernel in this release (the STF kernel allows IRAF to access the Space Telescope image format directly). The changes included the following.

- Header file caching. STF images often have quite large FITS headers which can be time consuming to read. A header file caching scheme is now used to optimize the kernel in cases where the same imagefile is repeatedly accessed, e.g., when successively reading each element of a large group format image. By default up to 3 header files will be cached; this default should be fine for most applications. If necessary the number of cache slots can be changed by defining the integer variable "`stfcache`" in the IRAF environment (the builtin maximum is 5 cached headers per process).
- *The semantics of the kernel regarding header updates have changed.* STF images differ from other IRAF images in that they may consist of a group of images all in the same file, with each individual image having its own header (the group header), plus a single global FITS header shared by all images in the group. This is no problem in a read operation, but in a write or update operation there can be problems since parameters cannot be added to or deleted from the individual group headers. The new semantics regarding STF image header updates are as follows: 1) when updating the header of a multigroup image (not recommended) only the group header is updated, and attempts to add new parameters are ignored; 2) when updating the header of an image containing a single group, both the group header and the FITS header are updated.

  As a result of these changes, the behavior of a single group STF image is now identical to that of a regular IRAF image. It is recommended that multigroup STF images be treated as read only if possible, creating only single group images during interactive processing (except when running a program that is explicitly designed to create multigroup images).

- The kernel was modified to work with the new MWCS (world coordinate system) interface. The image section transformation is now performed by MWCS rather than by the STF kernel.
- A number of minor changes were made to the way the group parameter block (GPB) cards are maintained in the IRAF image descriptor. The comments on GPB definition cards are now preserved. Restrictions on the grouping of GPB cards in the header have been removed.
- A number of bugs were fixed and restrictions removed, e.g., the size of a header is no longer limited to 32767 characters (404 lines).

The IRAF core system and NOAO science applications were extensively tested with both single and multigroup STF images using the new kernel, and we now feel that it is safe to use the STF image format with these tasks, although the regular format is preferred if there is no special reason to use the STF format (the regular format is more efficient).

2.11. QPOE (event list image format) enhancements

The QPOE image kernel, used for event list data (photon counting detectors, e.g., X-ray satellites such as ROSAT) underwent the following changes.

- MWCS (world coordinate system) support has been added (§4.2.2). This provides a consistent coordinate system despite, e.g., the blocking factor, rect, or image section used to construct an image matrix from an event list.
- When opening a QPOE file as an IRAF image, the runtime filter expression used to create the image matrix is now saved in the parameter `QPFILT`*n* in the image header (multiple cards are used for long expressions).
- Region masks of arbitrary complexity and size can now be used to mask the event list when reading time-ordered or unordered (unindexed) event lists. This is done using the new PLRIO package (§4.2.5) which provides the capability to efficiently random access large image masks of arbitrary complexity.
- Unmatched brackets, braces, or parentheses are now reported as an error by the filter expression parser (this can occur even with a valid expression, e.g., due to truncation of the expression string). A reference to an undefined keyword, e.g., due to a spelling error, is now detected and reported as an error. Any errors occurring during expression parsing will now result in termination of the calling task, unless caught in an error handler.
- A number of bugs were fixed.

2.12. Changes affecting image display in VMS/IRAF

A new version of Nigel Sharp's UISDISPLAY program, for image display on VMS systems running UIS, has been installed in "`iraf$vms/uis`". An executable for an early version of the SAOIMAGE display program for the X window system, written by Mike VanHilst (SAO), and ported to VMS by Jay Travisano (STScI) has been placed in the directory "`iraf$vms/x11`". An executable for a VMS version of XTERM (the X window terminal emulator, ported to VMS by Stephan Jansen), is also in this directory. We wanted our VMS users to have access to these programs, although more development work and testing is needed before we can offer good support for X window based image display and graphics on VMS. A more comprehensive package providing enhanced capabilities should be available as an add-on later this year.

3. IRAF Package Revisions

The most notable changes to the tasks in the IRAF packages are summarized below. Further information may be obtained by reading the help page for each task, or by paging the revisions file for a particular package. For example, to page the revisions for the DATAIO package:

```
cl> phelp dataio.revisions op=sys
```

3.1. Changes to the System Packages

3.1.1. Modifications to tasks in the DATAIO package

- The RFITS and WFITS tasks have been modified to add support for the IEEE floating point format. The "bitpix" parameter in WFITS can be set to -32 or -64 to specify real or double precision IEEE floating numbers on output. RFITS recognizes these same values in the bitpix keyword in the FITS header on input and converts the data accordingly. Note that this option must be selected by the user as the defaults for writing a FITS tape have not changed. The user is cautioned that support for the IEEE floating formats is a new feature of FITS and may not be supported by all FITS readers.
- RFITS was modified so that the "iraf_file" parameter can be a list of output images or a image root name.

3.1.2. Modifications to tasks in the IMAGES package

- MWCS (world coordinate system) support was added to those tasks in the IMAGES package which change the geometry of an image, i.e., IMSHIFT, SHIFTLINES, MAGNIFY, IMTRANSPOSE, IMCOPY, BLKREP, BLKAVG, ROTATE, IMLINTRAN, REGISTER, and GEOTRAN (REGISTER and GEOTRAN only support simple linear transformations). If one of these tasks is used to linearly transform an image, the world coordinate system (WCS) in the image header will be updated to reflect the transformation. Note that MWCS is disabled by default in IRAF V2.9, and must be explicitly enabled to allow these tasks to edit the image header to update the WCS (see §2.2).
- The IMSTATISTICS task was modified. The "verbose" parameter was renamed "format" with the default being set to "yes" (fixed format with column labels). Otherwise the fields are printed in free format with 2 blanks separating the fields. The name of the median field has been changed to "midpt".
- The IMHISTOGRAM task has a new parameter called "hist_type" that gives the user the option of plotting the integral, first derivative, or second derivative of the histogram instead of the normal histogram.

3.1.3. Modifications to tasks in the LISTS package

- The RIMCURSOR task in the LISTS package was completely rewritten to add MWCS support, so that coordinates may be output in any user specified coordinate system defined by the WCS information in the image header of the reference image. For example, if an image with a TAN projection WCS is loaded into the image display, RIMCURSOR may be used to print the right ascension and declination at the location defined by the image cursor. Refer to the help page for details.

3.1.4. Modifications to tasks in the PLOT package

- A new graphics kernel task IMDKERN (written by Zolt Levay at STScI) has been added to the PLOT package. The new graphics kernel allows the graphics output of any task to be plotted as a graphics overlay on the image display. As with the other graphics kernels, this may be done by calling the IMDKERN task directly, but is more often done by specifying the image display (e.g., device "`imd`") as the output device when running a graphics task. Refer to the help page for details.

- The CONTOUR task was modified so that it could be used with IMDKERN to overlay contour plots on the image display. If the parameters `fill=yes` and `perimeter=no` are set the contour plot is scaled to fill the entire device viewport and all axis and plot labeling is disabled. If the image being displayed also fills the entire device viewport (display frame) then the contour plot will be drawn to the same scale as the displayed image. Refer to the help page for details.
- Several tasks in the PLOT package were modified to allow use with image specifications containing brackets, e.g., group format images, QPOE filter expressions, and image sections. The tasks modified were PROW, PROWS, PCOL, PCOLS, SURFACE, and CONTOUR.
- An option was added to the PVECTOR task to output the vector (cut through the image at an arbitrary angle and center) as a text file or image, rather than plotting the vector.

3.1.5. Modifications to tasks in the SYSTEM package

- A new task PHELP (paged help) was added to the SYSTEM package. PHELP is a script task front end to HELP which collects the output of HELP in a scratch file and pages it with the system pager, allowing one to randomly skip around to read the help text. Note that paging of all the help pages in a package is supported, e.g.,

  ```
  cl> phelp images.*
  ```

  would page all the help files for the IMAGES package.
- The NEWS task was completely rewritten, and is now used to page the revisions summary for the current and previous releases. In other words, one can now type NEWS to find out what is new in the current release.
- The GRIPES task was modified to send mail to `iraf@noao.edu` or `5355::iraf`. The IRAF site administrator may want to check this script for compatibility with the local mail system.

3.2. Glossary of New Tasks in the IRAF System Packages

Task		Package		Description
imdkern	-	plot	-	Image display device (IMD) graphics kernel
news	-	system	-	Summarize what is new in the current release
phelp	-	system	-	Paged HELP: collects and pages the output of HELP
rimcursor	-	lists	-	Read image cursor position in world coordinates

3.3. Changes to the NOAO Packages

3.3.1. New NOAO Packages

A new package ARTDATA, used to generate artificial data, has been added to the NOAO packages. ARTDATA includes tasks for the generation of star fields, optionally containing galaxies, and one and two dimensional spectra as well as simple test pattern images. The tasks GALLIST and STARLIST provide many options for producing lists of galaxies or stars that can then be used by the task MKOBJECTS to produce output images. The tasks MK1DSPEC and MK2DSPEC provide tools for making artificial spectral data. The task MKNOISE allows the user to add readout noise, poisson noise and/or cosmic ray events to new or already existing images. The task MKPATTERN allows the user to make images from a choice of patterns.

3.3.2. Modifications to Existing NOAO Packages

3.3.2.1. The ASTUTIL package

- The task SETAIRMASS in the ASTUTIL package was modified so that it now precesses the coordinates to the epoch of the observation.

3.3.2.2. The DIGIPHOT.APPHOT package

- A new task APTEST was added to the DIGIPHOT.APPHOT package that tests the execution of the package. Output files are generated that the user can review.
- Two new parameters were added to DATAPARS, "datamin" and "datamax". Pixels outside this range are rejected from the sky fitting algorithms and from the non-linear least square fits in FITPSF and RADPROF.
- An "update" parameter was added to all of the APPHOT tasks. If the "verify" parameter is set to "yes" and the task is run in noninteractive mode `update=yes` will update the critical parameters in their respective parameter sets.
- Four new parameters, "airmass", "xairmass", "filter", and "ifilter", were added to the DATAPARS task. These parameters provide the user the option of having the filter and airmass quantities from the image headers to be carried over into the APPHOT database files for later transmission to calibration programs.
- A new algorithm "mean" was added to the sky fitting options.
- A setup menu mode was added to all the APPHOT tasks. When the user types "i" in interactive mode a setup menu is presented rather than a fixed set of predefined commands.

3.3.2.3. The IMRED.IRRED package

- The APSELECT task (from the APPHOT package) has been made visible.
- The image i/o for IRMOSAIC, IRALIGN, IRMATCH1D, and IRMATCH2D has been optimized so things should run much faster. There is now an option to trim each section before insertion into the output image. The actions of these tasks can now optionally be output to the terminal.

3.3.2.4. The IMRED.MSRED package

- A task called MSBPLOT was added to the IMRED.MSRED package. This task allows the user to plot a range of lines in multispec images in batch mode.

3.3.2.5. The ONEDSPEC package

- Several modifications were made to the ONEDSPEC package. These changes affect all of the IMRED packages that include these tasks as well.
- The equivalent width measurement using the "e" keystroke in SPLOT is now computed using the ratio of the spectrum to the continuum. The previous approximation is included in the logfile for comparison.
- The DISPERSION task will now add CD*i_j* (CD matrix) keywords to the image header as an alternative way of expressing the dispersion function. If the keywords W0 and WPC or CRVALn and CDELTn are not in the image header the tasks reading this information for setting the wavelength (IDENTIFY, SENSFUNC, SPLOT, and SPECPLOT) will look for the CD*i_j* keywords. This change should have no affect on the NOAO applications but provides compatibility with STSDAS applications using the new MWCS interface provided with IRAF version 2.9.

- The call to the CALIBRATE task in the script task BATCHRED was modified so that the "extinct" parameter is always set to "yes". Since CALIBRATE checks to be sure the data has not been previously extinction corrected this simple change provides more flexibility.

3.3.2.6. The PROTO package

- Two new tasks, IMALIGN and IMCENTROID, were added to the package. IMCENTROID computes a set of relative shifts required to register a set of images. The task IMALIGN both computes the shifts and aligns the images.
- The JOIN task (previously a simple script) has been replaced by a compiled version which removes many of the restrictions of the previous version.
- The IR tasks have been modified as mentioned above under the IMRED.IRRED section (§3.3.2.3).
- The TVMARK task was modified to permit deletion (the "u" key) as well as addition of objects to the coordinate file. Another cursor keystroke, the "f" key, was added allowing the user to draw a filled rectangle.

3.3.2.7. The TWODSPEC.LONGSLIT package

- Tasks in the TWODSPEC.LONGSLIT package that are used for setting wavelength information (EXTINCTION, FLUXCALIB, and TRANSFORM) were modified for the CD*i_j* keywords as outlined above for ONEDSPEC.

3.4. Modifications and Additions to Calibration Data

The calibration data used by some of the tasks in the TWODSPEC, ONEDSPEC, and many of the IMRED packages are kept in a directory called ONEDSTDS in `noao$lib`. The current contents of this directory are best summarized by paging through its README file, e.g.,

```
cl> page noao$lib/onedstds/README
```

A new directory spec50redcal in "noao$lib/onedstds" has been added containing flux information for standard stars. The data in this list are from Massey and Gronwall, Ap. J., July 20, 1990.

3.5. Glossary of New Tasks in the NOAO Packages

Task		Package		Description
aptest	-	apphot	-	Run basic tests on the apphot package tasks
gallist	-	artdata	-	Make an artificial galaxies list
imalign	-	proto	-	Register and shift a list of images
imcentroid	-	proto	-	Compute relative shifts for a list of images
mk1dspec	-	artdata	-	Make/add artificial 1D spectra
mk2dspec	-	artdata	-	Make/add artificial 2D spectra using 1D spectra templates
mknoise	-	artdata	-	Make/add noise and cosmic rays to 1D/2D images
mkobjects	-	artdata	-	Make/add artificial stars and galaxies to 2D images
mkpattern	-	artdata	-	Make/add patterns to images
msbplot	-	msred	-	Batch plots of multispec spectra using SPLOT
starlist	-	artdata	-	Make an artificial star list

4. Programming Environment Revisions

4.1. Changes to the Programming Utilities

4.1.1. MKPKG changes

The MKPKG utility can now substitute the contents of a file back into the input stream, as a special case of the macro replacement syntax. For example, the sequence

```
abc$(@file)def
```

would be translated as

```
abc10def
```

if the file "*file*" contained the string "10". The replacement is performed by inserting the contents of the file back into the input stream, replacing sequences of newlines, spaces, or tabs by a single space, and omitting any trailing whitespace.

The "-p <pkg>" argument to MKPKG, XC, and so on loads the environment of the named package *pkg*, to define the package environment variables, load the mkpkg special file list, define the directories to be searched for global include files and libraries, and so on. Multiple "-p" arguments may be given to load multiple package environments. What is new is that if `pkglibs` is defined in the environment of a package to list the package library directories to be searched (the usual case), and multiple package environments are loaded, successive redefinitions of `pkglibs` will *add* to the list of directories to be searched, rather than redefining the old list as each new package environment is loaded. For example, if two package environments are loaded, and each defines its own library, both libraries will be searched.

4.1.2. Generic preprocessor

A minor change was made to the generic preprocessor which affects how strings such as "FOO_PIXEL" are translated. In the usual case, the preprocessor replaces all occurrences of "PIXEL" by "int", "real", or whatever the actual datatype is. The translation is now context sensitive. Rather than translating "FOO_PIXEL" as "FOO_int" (e.g., "MII_PIXEL" -> "MII_int"), the type name will now be output in upper case if the rest of the name in which it occurs is upper case. Hence, a string such as "MII_PIXEL" will now be translated as "MII_INT". This allows the use of generic constructs to symbolize SPP macros.

4.1.3. SPP changes

The language constant ARB, formerly defined as 32767, is now treated differently depending upon how it is used. In a declaration of an array argument, ARB is replaced in the output Fortran by a "*", e.g., "`int data[ARB]`" becomes "`INTEGER DATA(*)`". In an executable statement, ARB is replaced by a very large ("arbitrarily" large) integer value, e.g., to define a DO-loop which is to loop an arbitrary number of times. If ARB is mistakenly used to dimension an array which is a local variable rather than an argument, the SPP translator will now detect and report the error.

4.1.4. Interactive development and the process cache

Whenever a CL task is run and the process containing the task is already idling in the CL process cache, the CL will now check to see if the modify date on the process executable has changed, and restart the process if the executable has been modified. For example, when doing software development from within the CL and a process is alternately relinked and tested, the CL will now automatically detect that the process has been relinked and will run the new process, without any need to manually flush the process cache.

4.2. Programming Interface Changes

4.2.1. New MWCS interface (world coordinate system support)

A major new VOS interface MWCS, providing general facilities for linear and nonlinear world coordinate systems, has been added to the programming environment and is used in IRAF V2.9 in IMIO, IMAGES, and other parts of the system. MWCS is intended for use in scientific applications as well as in system code such as IMIO, hence is of potential interest to anyone developing software within the IRAF environment. The source directory is "`mwcs`" and the interface is documented in the file "`mwcs$MWCS.hlp`". Users should be aware that, although the new interface addresses the general WCS problem and has been carefully designed, a second version of the interface is planned and the current interface is not yet a "frozen" interface.

4.2.2. QPOE interface changes

The QPOE (event list image) interface has been extended to add routines to store MWCS objects in the QPOE header. By default, there is one MWCS per QPOE file, stored encoded in a machine independent binary format in a variable length array *qpwcs* of type *opaque*. The new routines are as follows:

```
  mw = qp_loadwcs (qp)
       qp_savewcs (qp, mw)
mw = qpio_loadwcs (io)
```

The routines *qp_savewcs* and *qp_loadwcs* merely save a MWCS in the QPOE header, or load a previously saved one. The QPIO (event i/o) routine *qpio_loadwcs* is like *qp_loadwcs*, except that it will also modify the Lterm of the MWCS to reflect any blocking factor or "rect" specified in the filtering expression when the event list was opened. The new routine is called automatically by QPF and IMIO whenever a QPOE event list is opened under image i/o, making the physical coordinate system of the image matrix the same as physical event coordinates.

The calling sequences of the *qp_add* and *qp_astr* routines, used to conditionally add or update header parameters, have been changed (so far as we could determine very few programs exist yet which use these routines, so we decided to risk an interface change). The change made was to add a *comment* argument. This change was motivated by the observation that people would not use the routines but would instead use lower level routines, in order to be able to set the comment field if the parameter has to be added to the header.

4.2.3. IEEE support routines added

Routines for IEEE floating to native floating conversions have been added to the MII and OSB interfaces. The new MII routines are as follows:

```
nelem = miiread[rd] (fd, spp, maxelem)
       miiwrite[rd] (fd, spp, nelem)
         miipak[rd] (spp, mii, nelems, spp_datatype)
         miiupk[rd] (mii, spp, nelems, spp_datatype)
```

The *miiread* and miiwrite routines are like their FIO counterparts, except that they are used only with data of the indicated type, and perform the IEEE to native floating conversion (or vice versa) as part of the i/o operation. The *miipak* and *miiupk* routines pack (native→IEEE) and unpack (IEEE→native) arrays of the indicated type.

The lowest level conversion routines are the OSB routines, which are what the MII routines use to perform the lowest level translation.

```
        ieepak[rd] (datum)
        ieeupk[rd] (datum)
       ieevpak[rd] (native, ieee, nelem)
       ieevupk[rd] (ieee, native, nelem)
    iee[sg]nan[rd] (NaN)
```

The *ieepak* and *ieeupk* routines transform a single scalar value in place, while the *ieevpak* and *ieevupk* routines transform vectors (note that the package prefix is "iee", not "ieee"). In-place vector conversions are permitted. Since IRAF does not support the IEEE not-a-number formats, *NaN*, *Inf* etc. values are converted to a legal native floating value on input. The native floating value to which *NaN*s are mapped (default zero) may be globally set with *ieesnan*.

On some systems, e.g., the VAX, the low level conversion routines may be written in assembler or machine dependent C. If so, the source file actually used by the system will be found in the "`host$as`" directory.

4.2.4. New routine GETLLINE added to FIO

A new routine *getlline* (get long line) has been added to FIO. This is similar to *getline*, except that it will reconstruct arbitrarily long newline delimited lines of text, whereas *getline* returns at most SZ_LINE characters.

```
nchars = getlline (fd, outstr, maxch)
```

The new routine should not be confused with the old routine *getlongline*, a higher level routine which performs a similar function, but which also ignores comment lines and help blocks, and maintains a line counter.

4.2.5. Modifications to PLIO/PMIO

A new routine *p[lm]_sectnotconst* has been added to PLIO and PMIO (the pixel list and image mask interfaces). As the name suggests, the routine tests whether a given rectangular section of the mask is all at the same value, and if so returns the mask value as an output argument.

```
bool = pl_sectnotconst (pl_src, v1, v2, ndim, mval)
```

A new subpackage PLRIO has been added. This is used to efficiently random access any 2D plane of an existing pixel list or image mask.

```
 plr = plr_open (pl, plane, buflimit)
       plr_setrect (plr, x1,y1, x2,y2)
mval = plr_getpix (plr, x, y)
       plr_getlut (plr, bufp, xsize,ysize, xblock,yblock)
       plr_close (plr)
```

The mask is opened for random access on a special descriptor which incorporates a scaled, active 2D lookup table. Most subsequent *plr_getpix* calls will return the given mask value directly from the table with very little overhead; only if the referenced pixel occurs in a region too complex to be described by a single table entry is the value computed by direct evaluation of the mask. A special 2D binary recursive algorithm (using *pl_sectnotconst* above) with log2(N) performance is used to calculate the scaled lookup table. These algorithms provide efficient table generation and random mask pixel access even for very large masks.