AIPSLETTER

Volume XV, Number 2: July 15, 1995

A newsletter for users of the NRAO
Astronomical Image Processing System

Written by a cast of *AIPS*

Edited by
Eric W. Greisen
National Radio Astronomy Observatory
520 Edgemont Road, Charlottesville, VA 22903–2475
804–296–0211, x209; TWX 910–997–0174; Fax 804–296–0278
aipsmail@nrao.edu

The Good News . . .

The 15JUL95 release of Classic *AIPS* is now available. It may be obtained via *anonymous* ftp or by contacting Ernie Allen at any of the addresses given in the masthead. As of this writing, 119 copies of the 15JAN95 release have been given out electronically (7 tar.Z, 22 tar.gz, and 24 binary over 7 operating systems) or on magnetic tape (39 8mm, 24 4mm, 3 QIC, 1 3.5-inch floppies and *no* 9-track). 80 of the 119 were of the full binary release. Outside of NRAO, the 80 sites receiving 15JAN95 indicated their plans to run *AIPS* on 147 SUN OS 4, 198 SUN Solaris, 18 IBM AIX, 36 HP-UX, 41 DEC Alpha, 17 SGI Irix, 14 PC Linux and 1 Convex computers. A total of 472!

The 15JUL95 release is the first release under a new system designed to protect NRAO's intellectual property rights, while making *AIPS* more readily available to both the astronomy and non-astronomy communities. All files are now copyrighted by Associated Universities, Inc., NRAO's parent corporation, but are made freely available under the GNU General Public License (GPL). This means that User Agreements are no longer required, that you may obtain copies via anonymous ftp without contacting Ernie Allen, and that you may redistribute (and/or modify) the software, under certain restrictions, if you so choose. You may *not* sell this software; it remains free to everyone. Details on this new way to get *AIPS* and the text of the GNU GPL appear later in this *AIPSLetter*.

This release contains the new task IMAGR, intended to replace all previous imaging and Cleaning tasks (*e.g.*, MX, HORUS, UVMAP, and WFCLN). It offers all capabilities of these tasks, with some corrections, plus a wide range of new data weighting, TV interaction, and wide-field and wide-bandwidth correction options. The iterative self-cal and imaging task SCMAP now offers many of the same imaging and interactive options. The *CookBook* chapters on imaging and deconvolution were combined into one chapter describing IMAGR and the VLBI data reduction chapter was completely revised. It will need to be revised further, however, to describe numerous improvements and additions to the arsenal of VLBI calibration tasks, some of which appear in the current release.

We expect to be able to offer a position in the Classic *AIPS* Group, to be based in Charlottesville, in the near future. If you are interested, please contact us at the addresses given above.

. . . and the Bad

On May 1, Phil Diamond was promoted to Deputy Assistant Director of the Socorro Array Operations Center with responsibilities for computing and array operations. While this may be good news for Phil, it is not for the *AIPS* group. We are now reduced to two general-purpose programmers, but see above.

While we are reduced in manpower, the official expectations on the longevity of the Classic *AIPS* project have been substantially increased. Phil's promotion came because Tim Cornwell was named Assistant Director for the AIPS++ project. Tim is reorganizing that project and has stated that he expects it will not replace Classic *AIPS* for about five years, *i.e.*, the year 2000. (Classic *AIPS* was first "frozen" by management in April 1991, but freon seems to be in short supply these days.)

The use of Sun OS 4.1.*x* is being phased out at the NRAO in favor of the Solaris operating system, currently at Solaris 2.4 (also called SunOS 5.4). This means that the `15JUL95` version will be the last to be tested extensively under the old Sun Berkeley-based operating system. We will keep one or more computers on the old system as long as we can and we do not anticipate major problems anytime soon, but it is inevitable that the quality of our support for the old OS will diminish with time.

CookBook Update Continues

The *AIPS CookBook* was last updated for the `15OCT90` release. Because a lot has changed in *AIPS* since then, we have decided to modernize the *CookBook*. We are doing this one chapter at a time and are making each chapter available via the World-Wide Web as soon as it is ready. For details of the Web, see the publications article in this *AIPSLetter*. The chapters changed so far are

- 1 — *Introduction* — Added new sections giving a project summary and a diagram of the structure of *AIPS*.
- 2 — *Starting Up AIPS* — Changed to describe workstation use, *AIPS* in networked environments, and managing the TV server `XAS`.
- 3 — *Basic AIPS Utilities* — Updated information about history files and disk allocation, added `ABOUT` and `APROPOS` to the help section, moved and updated tape mounting, and added a discussion on external disk files (Fits, text, ...).
- 4 — *Calibrating Interferometer Data* — With much help from Rick Perley and Alan Bridle, rearranged and corrected everything, adding a substantial discussion of when and how to edit and bringing the description of `TVFLG` up to date including a picture.
- 5 — *Making Images from Interferometer Data* — Rewrote old chapters 5 and 6 to describe the new `IMAGR` task rather than several old imaging tasks, to modernize the self-calibration description, and to replace the discussion of `IBLED` with one describing the current program.
- 6 — *Displaying Your Data* — Rewrote old chapters 7 and 8 to make a coherent, current, and complete description of printing, plotting, TV, and graphical data displays.
- 8 — *Spectral-Line Software* — Rewrote old chapter 10, replacing old outline format with a more coherent (and wordy) description of line analysis, emphasizing continuum subtraction and other more modern imaging techniques.
- 9 — *Reducing VLBI Data in AIPS* — Rewrote the old chapter to describe the nearly completely new software now available for the VLBA. This chapter will remain under active development for some time.
- 13 — *Current AIPS Software* — Replaced old lists with new ones produced for the `ABOUT` verb. Now current to the `15JUL95`version.
- A — *Summary of AIPS Continuum UV-data Calibration* — Inserted a new appendix giving an updated version of Glen Langston's outline of continuum calibration.
- B — *A Step-by-Step Guide to Spectral-Line Data Analysis in AIPS* — Inserted a new appendix by Andrea Cox and Daniel Puche giving *their* outline view of spectral-line data reduction in *AIPS*.
- Z — *System-Dependent AIPS Tips* — Replaced with whole new discussions including color printers, screen copying, film recorders, workstation environments. A method for people to have NRAO make slides for them is described.

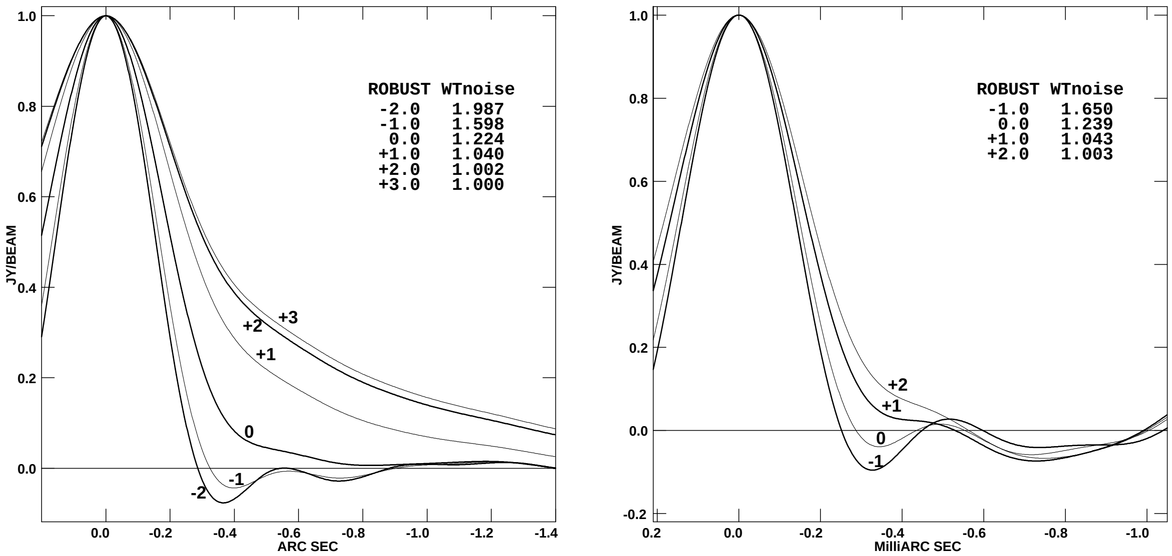
Improvements for Users in 15JUL95

Imaging

IMAGR

The new task **IMAGR** is intended to become the primary image making task in *AIPS*, replacing **UVMAP**, **HORUS**, **MX**, and **WFCLN**. It offers all of the capabilities of these tasks, including applying calibration to data from multi-source *uv* files. It does the Schwab-Cotton *uv*-plane subtraction form of the Clark Clean, in up to 16 fields, and can apply a variety of wide-field and wide-bandwidth corrections. The really new parts of the program lie in its ability to make 8192x8192 images, to sort the data (if needed), to use the TV display interactively, and to weight the data flexibly. Previous imaging tasks required the user to pre-sort the data, to accept poor forms of uniform weighting, and/or to put up with very inefficient multiple passes through the input data; **IMAGR** sorts the data if needed to avoid these things. The older tasks offered, at most, a TV display of one of the residual images and the option to terminate the Clean at the end of the current major cycle. **IMAGR** does its display before each major cycle, allowing you to interact with the dirty images or the current residual images. You may zoom and enhance the display and select both circular and rectangular Clean windows for each of the fields. The choice of field to display and window is made from a menu displayed on the TV. The menu offers numerous familiar functions including **CURVALUE**, **TVZOOM**, **TVPSEUDO**, **TVBOX**, and window setting as well.

IMAGR offers a large number of "knobs," in the form of adverbs, with which you may adjust the data weighting. To be honest, we must admit that we do not know what the optimum setting of the knobs might be, but we do know that they can make a significant difference in the signal-to-noise on images, can alter the synthesized beam width and sidelobe pattern, and can produce bad striping in the data when mildly wrong samples get substantially large weights. **IMAGR** allows the user to control the size of the "cells" in the *uv* plane used for counting samples in uniform weighting. It offers both circular and rectangular functions to control how a sample is counted as a function of distance from its location in the *uv* plane. It allows modification of the input data weights by various exponents for counting and/or weighting and performs the usual tapering. Finally, it offers a variation of Dan Briggs' "robust weighting" scheme to temper the wide divergence of weighting factors attempting to make the weights "uniform." The effect of this "**ROBUST**ness" parameter on synthesized beam patterns is illustrated in the accompanying Figure taken from the *AIPS CookBook*. **IMAGR** computes the effect of all of this weighting on the expected noise (compared to that expected from "natural" weighting) in the image and reports it in the image headers as parameter **WTNOISE**. The values of this parameter found for the beams in the Figure are shown in the accompanying tables.



Slices taken through the centers of synthesized beams for various values of the **ROBUST** parameter. Plot at left for a VLA A- and B-array data set, while the plot at right is for a VLBA data set. Do not assume that these plots apply to your data sets, however. Tables give noise increase over natural weighting ($\equiv$ large **ROBUST**).

To assist in the use of this new task, a new verb FILEBOX was written, TVBOX was revised to set circular as well as rectangular boxes, and the *CookBook* chapters on imaging and deconvolution were rewritten. FILEBOX assists the user to prepare a text file containing up to 500 (!!) rectangular and circular Clean boxes for each of up to 16 fields which can be fed to IMAGR as the initial selection of source-searching areas. It allows you to build up the list gradually, interacting with the TV display to prepare the list for the visible portion of the displayed field. All Clean box setting with the TV (verbs TVBOX, REBOX, FILEBOX and tasks IMAGR, SCMAP, PLAYR) uses the same subroutine which sets either kind of area and lets you modify existing Clean boxes and add new ones. Thus, NBOXES is now an *output* adverb from TVBOX and both an inputs and output adverb to REBOX. The new *CookBook* chapter combines old chapters 5 and 6 to describe IMAGR, SCMAP, and numerous other changes to imaging since the 1990 edition.

SCMAP

SCMAP is an OOP-based task intended to do imaging and Cleaning iterated with self-calibration. For this release, it has been improved by the addition of all of the new data-weighting and imaging options of IMAGR. SCMAP offers IMAGR's interactive TV options during Clean plus a similar interactive TV display before each self-calibration cycle. The latter will, some day, have an interactive data editing capability and a variety of other options to adjust the self-calibrations. At the moment, it is mostly just another way to adjust the Clean boxes. SCMAP has the ability to determine the basic SOLINT from the times found in the data set, but is currently restricted to a single SOLINT for all iterations.

Bugs and other worries

When doing Clark Cleans, programs have to make the decision about when to do another major cycle. Previously, all *AIPS* tasks chose not to start another major cycle when they were within 10% of the iteration limit. That is often a sensible choice, but is very wrong when the number of iterations in each major cycle previously has been very much less than 10% of the iteration limit. All tasks have been changed to extend a major cycle by no more than half of the previous major cycle.

Several bugs in the Clean component subtraction were corrected during the development of IMAGR. Subroutine ALGSUB did not compute the number of rows needed for bandwidth synthesis correctly and could, as a consequence, have failed to subtract the model from some channels of some *uv*-data samples. Subroutine ALGSTB, used for rotated or unsorted data sets, computed the maximum baseline wrongly and, as a consequence, put very large and erroneous fluxes in the residual data file. The gridded component subtraction routine replaced the correct (and simple) cell coordinate correction with an elaborate erroneous and non-linear correction. In some cases, *i.e.*, large fields near the North Pole, this caused error messages to appear and, more importantly, could cause the gridding of the components to be done erroneously. These bugs affected MX, UVSUB, WFCLN, CALIB, *et al.* and were capable of producing either subtle ripples in the Clean or completely erroneous (and hence obvious) outputs. Because these bugs had to be found by programmers in new tasks, it is believed that they did not affect very many people in a significant way. WFCLN had bugs which caused a scaling error when imaging with BIF $\neq 1$ and which could have caused addressing exceptions in bandwidth synthesis imaging.

Image analysis and display

Plot labeling

The task LWPLA has been using incorrect sizes for characters to try to correct for poor placement of the character strings by the plot tasks. These incorrect sizes led to misplacement of the strings in other ways. The solution was to go through all plot tasks and adopt a better and more standard way to place the labeling around the main plot areas. While doing so, we also added the adverb LTYPE to all plotting tasks which did not already have it and added new values for additional control over the labeling of plots. The verb EXTLIST was brought up to date for most, if not all, tasks.

KNTR, PCNTR

The task KNTR was changed to plot grey-scale images as well as contours. Either of the two input images can be used as the contour image and/or the grey-scale image. If one of the two images is a cube, multiple panels can be displayed. The other image can then be either a cube or a plane. The Clean beam may be plotted in a separate frame or in any of the four corners with various degrees of cross hatching. This makes CNTR and GREYS obsolete. KNTR can plot the edges between blanked and good pixels or leave them invisible under control of the new adverb DOBLANK.

PCNTR was improved to convert Stokes I, Q, and U on the fly rather than require the input of total polarization and angle images found by COMB. The latter are still accepted of course. The input images can now be polarization cubes or three separate images. The Clean beam may be plotted with various degrees of cross hatching in any of the four corners.

Displaying *uv* data

The display of *uv* data received some minor improvements and bug fixes for this release. UVHGM was brought into the modern world by teaching it to read compressed data, to do some Stokes conversion, and to plot more than one Stokes, channel, or IF on each plot. The meaning of the plot scaling adverbs was changed in UVPLT to give more freedom with partially self-scaled plots. The data binning mode in UVPLT was given the choice of weighted or unweighted averaging. VPLOT acquired greater flexibility in the selection of antennas and baselines to be plotted and became more understanding about strange values of the averaging interval. The averaging of frequency channels was corrected in CLPLT, the task which plots closure phases.

The task POSSM got a fair amount of attention. Various barriers to plotting single-dish "*uv*" data were removed. The scaling of multiple-frame plots was improved. It now appends to the output text file rather than replacing it, and will write multiple spectra to the file when the averaging interval SOLINT is set greater than zero. Spectrum reversals are now corrected in writing out the file. Labeling of reversed axes in channels was corrected, as was the computation of velocity labels.

LISTR got a new option. With OPTYPE = 'GAIN', setting STOKES = 'POLD' will get a display of the right versus left polarization gain ratio or their phase difference depending on the setting of DPARM(1). Other minor bugs in the gain listings were corrected, as were bugs in handling the list of antennas and baselines in the matrix mode.

TV improvements

The *AIPS* TV was changed in useful ways for the 15JUL95 release. Four more graphics overlay channels were added. These four are treated in a new way. The first four graphics channels actually require 15 colors so that places where they overlap will show in different colors. If we were to do that with four more planes, then all 256 levels would be used up and we could not display the images. Therefore, the second four graphics planes appear only if no lower-numbered plane is turned on at the pixel. Channel 8 is used for the black background surrounding lettering now, so there are four usable graphics channels with full coloring and three additional ones with partial capabilities. Users may need to change their .Xdefaults file; see HELP XAS for details. The image catalog files now need four more records for the new graphics planes, which will normally require them to be replaced.

The way in which XAS allocates its colors was also changed. If it cannot get enough colors in the system's default color table, then XAS allocates its own color table. When the cursor moves into the XAS window, this non-default table applies and the colors of the other windows on the workstation screen may change. In the new release, XAS takes its colors beginning at the top of the table. Then, if XAS does not use all 256 colors, the entries at the bottom levels of the table are not touched. This trick means that the first windows created as you log in — your basic xterms usually — will keep their colors when the cursor moves into the XAS window. In this way, you may actually get to see the instructions for using the TV while you are using the TV!

The way in which `AIPS` and tasks handle errors while talking to the TV was corrected. Previously, the close-down sequences could cause `AIPS` or the task to abort when trying to issue commands to a dead display. The *AIPS* startup procedures were modified to check for the environment variable `$AIPS_TV_BUFFERED` and to set the TV to a buffered mode if it has the value `YES`. This mode of talking to the TV is faster since it leaves out some of the error testing and other handshaking. The loss in reliability is probably insignificant, especially over slow network connections where any help with the speed is desirable.

VLBI data processing

Two-bit VLBA sampler corrections: ACCOR

The voltage thresholds of the two-bit VLBA samplers, which define the four states measured in this digitization scheme, may deviate slightly across the network and may also differ from the optimal theoretical values. The effect of errors in these level settings is to introduce amplitude scaling errors in the two-bit data, most generally showing up as an amplitude offset between the RR and LL cross-power data. These scaling errors can reach 10–20 percent in some cases. This effect can be removed by using the mean level of the autocorrelation data to determine the sampler scaling error (L. Kogan, VLBA Scientific Memo 9). The new task `ACCOR` in *AIPS* implements this method. `ACCOR` produces a solution (`SN`) table containing the amplitude scaling factors which can be applied using `CLCAL` in the standard manner. The sampler scaling errors for one-bit VLBA data are generally negligible. For further information contact L. Kogan (`lkogan@nrao.edu`).

Amplitude calibration with ANTAB and APCAL

Two new tasks have been introduced to improve amplitude calibration of VLBI data, replacing and expanding the functionality presently offered by `ANCAL`. The first task, `ANTAB`, reads a priori calibration text files containing system and/or antenna temperature data and gain curve information and updates the information in the system temperature (`TY`) and gain curve (`GC`) tables accordingly. Greater flexibility is offered in reading the calibration files, including improved handling of VLA calibration data, more flexible assignment of tabulated data to individual IF and polarization pairs, and an allowance for tabulated and IF dependent gains. The second task, `APCAL`, uses the system temperature and gain curve information to generate a solution (`SN`) table containing amplitude calibration information. `APCAL` allows a simultaneous solution for atmospheric opacity and the corresponding adjustment of the amplitude gains, a consideration which may be important for observations at 22 GHz and 43 GHz. The use of `ANTAB` and `APCAL` allows greater selection and control of primary amplitude calibrators. These tasks are the presently recommended route for a priori amplitude calibration of VLBI data. For further information contact Athol Kemball, `akemball@nrao.edu`.

VLBA delay decorrelation corrections

Further to the notice in the last *AIPSLetter*, development has continued regarding the correction of VLBA delay decorrelation losses within AIPS. The two major effects are spectral averaging decorrelation, which is caused by spectral pre-averaging in the correlator in the presence of unknown residual delays, and alignment or segmentation losses caused by the misalignment of FFT segments due to residual delay errors. The `15JAN95` release of *AIPS* allowed corrections for the more significant effect of spectral averaging decorrelation (typically a few percent). This release adds the smaller correction for FFT alignment losses (usually a few tenths of a percent).

To incorporate these corrections fully, we have added a new `CQ` table containing VLBA correlation parameters for each *AIPS* IF. The correction is now activated if the `AN` table array name keyword identifies the data as originating from the VLBA correlator and the `CQ` table is present. This more comprehensive solution supersedes the previous use of the `SPEC_AVG` keyword in the `AN` table as described in the last *AIPSLetter*. The `CQ` table is created by `FITLD` via adverb `DELCORR`. Task `FXVLB` will build a missing `CQ` table but must be used before any averaging or selection in frequency. The corrections are thereafter automatically made whenever delay corrections are applied. If the correction is not activated, a warning is given.

A full description of these corrections can be found in a forthcoming *AIPS* Memo (No. 90) by A. Kemball. For further information contact him at `akemball@nrao.edu`.

VLBA correlator digital corrections in FITLD

The *AIPS* data filling task `FITLD` has been upgraded to make digital corrections for data processed by the VLBA correlator. These include constant amplitude scaling corrections (b-factor) for cross-power data and full digitization corrections for autocorrelation data. Cross-power digitization corrections are performed in the frequency domain while autocorrelation data, which have higher correlation coefficients, are transformed to the lag domain before digitization corrections are applied. The autocorrelation digitization correction is only exact if zero-padding is used in the correlator due to the lag domain response of an FX correlator. Without zero-padding this correction becomes more exact as the autocorrelation data approach flat, continuum spectra. The autocorrelation correction is recommended even if zero-padding was not used as the digitization correction will generally exceed any errors due to incorrect zero-padding. The digital corrections are controlled using adverb `DIGICOR` in `FITLD`. Further details regarding digitization corrections can be found in VLBA Scientific Memo 6 by L. Kogan. For general information regarding `FITLD`, contact P. Diamond (`pdiamond@nrao.edu`).

Pulse calibration of VLBA data

VLBA pulse-calibration information can be loaded into the new *AIPS* pulse-calibration (`PC`) table using task `PCLOD`. This task, which will be used temporarily until this information is passed directly through `FITLD`, reads an external text file containing the VLBA pulse-cal data. Users are directed to their technical contact person for information on how to obtain the `PCLOD` text file for their experiment.

The task `PCCOR` uses the pulse-cal data in the `PC` table to generate a solution (`SN`) table which corrects the instrumental delay and phase offsets between individual baseband converters. This can be applied prior to fringe fitting using `CLCAL` in the standard manner. This is the first release of `PCCOR` and further development is expected. General information regarding VLBA pulse-calibration corrections can be found in VLBA Scientific Memo 8 by C. Walker. For further information contact L. Kogan (`lkogan@nrao.edu`) or A. Kemball (`akemball@nrao.edu`).

Polarization calibration tasks contributed

Two new polarization calibration tasks have been implemented in this *AIPS* release. These tasks were developed by Kari Leppanen as part of his thesis work and offer an alternative method of solving for instrumental feed polarization when the polarization calibrator is spatially resolved. As such, these methods are useful in processing polarization data taken at high frequencies. Further details of the algorithms can be found in Leppanen, Zensus and Diamond (Ap.J., in press). The tasks are `BLAVG`, which allows a more robust estimation of differential polarization delay offsets and `LPCAL`, a feed polarization calibration task which allows for spatial structure in the polarization calibrator. The use of these tasks and their interaction with other tasks in *AIPS* will be included in a forthcoming update to the VLB chapter of the *CookBook*. For further information contact A. Kemball (`akemball@nrao.edu`).

Other VLB-related changes

A variety of other changes and improvements were made in VLBI-related software. A bug in `FRING` affecting data sets for which the first IF was not at zero frequency was corrected, as was a bug in the handling of memory which led to failure messages mentioning `SOLINT`. `FRING` now also re-references the multi-band delays when appropriate. `MK3IN` was changed to allow up to 66 baselines and to accommodate data sets with a large amount of frequency switching. The handling of the reference date was corrected to prevent errors when the antenna tables were written to tape out of time order. `UVGLU` was given the smarts to glue together data sets that do not match exactly, but which are in time order. `FITLD` had a number of minor corrections made to stem the proliferation of `FQ` numbers and to deal with unusual sequencing of data sets. `SNSMO` had a number of changes including correcting for 2π ambiguities before direct smoothing of phases, fixing the sign of the phase in `VLMB`-type smoothing, and re-enabling clipping of fringe rates.

Ed Fomalont was also busy for this release. He added time range and IF selection to `MBDLY` and improved its testing for bad fringe fits. He corrected the geometric calculations in `CL2HF` for converting to Haystack-style delays and rates, changed some of the adverbs, and dropped observations below 0.5 degree elevation. He also made lots of changes to HF2SV including adding the OUTFILE adverb. These two tasks convert *AIPS*-style CL tables into files that can be read by the Goddard calc and solve programs. Ed also created a new task called HFPRT to print the contents of the HF extension file.

UV data calibration and manipulation

This area of the code received a fair amount of attention for this release, but mostly in the form of small bug fixes to be mentioned in the miscellaneous section below. A number of bugs in SPLIT were corrected. The header frequency was calculated erroneously when the first IF was not at the initial reference frequency and also when some, but not all, channels were averaged. BIF > 1 was ignored for all but the first source written. The routine used to average spectral channels was replaced with a more reliable one, already used in other tasks. This better routine was also put into AVSPC.

UVCOP was given a new option to delete data with weights below a user-specified threshold. This will be especially useful as a method for deleting bad data from VLBA data sets. The saga of corrections to UVCOP to account for data selection in the output tables continued for this release with corrections made to the handling of CL and IM tables.

Single-dish data in *AIPS*

A potentially serious bug was found in SDGRD, the task used to convert single-dish "*uv*" data into images. One non-existent sample from each buffer was gridded on each call to the gridding routine. If that portion of the memory was not suitably zero, then a square box the size of the convolving function support and of significant strength could appear at some location in the images of some, usually not all, spectral channels. The problem got worse if multiple passes through the data were required to image all spectral channels. Observers using the 12-meter on-the-fly imaging mode with SDGRD should consider remaking their images with the corrected task. Note that this bug affected only some of the images computed by SDGRD. A number of lesser bugs mostly having to do with allocation of memory were also corrected. See the article on patches for additional details.

SDGRD was also improved. The default weighting was changed to natural since uniform makes less sense in the single-dish case. An option to compute and put out an image of the expected noise (actually of $1/\sigma^2$) was added. This image may be used with the new task WTSUM to do weighted sums of images. This should allow single-dish users to image portions of their data, combining the output images, rather than having to image all of the data at one time (in very large files). Time smoothing was added to OTFUV, the task that converts 12m OTF data to *AIPS*, in another attempt to reduce the size of OTF data files.

Miscellaneous changes of interest to users

FITTP A bug caused FITTP to write an extra header-like record in place of table data when the table file was empty. The bug was fixed and the *AIPS* FITS readers changed to ignore this error. The readers were ignoring all following tables instead of the error previously.

SAD The source rejection logic was changed to reject some rather than all components in a multi-component island when some of them fail to meet the inclusion criteria.

GET4NAME New verbs to get and clear the fourth set of image name parameters were added.

PLAYR New task to enhance TV displays, blink two images, and the like chosen from a menu. It is a demonstration task for the *AIPS* OOP TV class, but may be useful for preparing color tables or looking at your images.

VTESS All of the *TESS tasks, especially UTESS and VTESS, had their use of buffers corrected and/or changed to employ appropriately larger sizes.

CONVL Messages about the convolving beam size and about any failures in its deconvolution were added.

calibration Calibrator source lists must be ignored when using SN tables generated on single-source data sets; a table parameter to describe the origin of the SN table was implemented.

flag tables Copied flag tables also can have entries incompatible with the current data set. Bugs, producing unpredictable flagging and other effects, were exterminated.

CVEL A bug causing data to be flagged excessively was found; see the patches report for details. Another (unpatched) addressing bug caused the time and antenna number to be passed into the shifting routines as though they were data. This produced subtle differences in the output spectra.

UVFLG A DTIMRANGE option was added for text-file input to control the amount by which the stated times are expanded before being written into the flag table.

FILLM The wrong reference channel was put in the header for bandwidth codes 8 and 9. It was set to the end rather than the center. FILLM also overstated the number of samples deleted for shadowing by doing that test before any other tests for data eligibility.

CLCOR A common error in correcting antenna positions (OPCODE 'ANTP') was to fail to set CLCORPRM(7) to indicate the phase convention (VLA versus VLB). The task was changed to check the antennas file for this information.

UVFIT Array sizes were increased to allow the program to work with the number of parameters it now attempts to fit.

KEYIN Modes in which the input text lines are echoed were enabled to assist in debugging large KEYIN-style input files used in ANCAL, ANTAB, APCAL, BLING, FETCH, MK3IN, PCLOD, SETAN, UVFLG, and VLBIN (at least).

Improvements Primarily for Programmers in 15JUL95

The area of OOP code and the TV display received a fair amount of attention for the 15JUL95 release. A new AP class was created, primarily so that each "AP" application subroutine could claim the pseudo-AP memory for itself without fearing that other subroutines would also expect free access to that memory. Since it is all one big COMMON these days, some "device" allocation scheme seemed like a good idea. To reserve the pseudo-AP memory, call APOBJ with an OPEN operation code and the current subroutine name. Be sure to close it as soon as your subroutine no longer depends on the contents of the AP memory.

The big change in this area was the creation of a TV device class and a TV utility library to make its use straightforward. At present, these are documented only in the $APLOOP files TVDEVICE.FOR and TVUTIL.FOR, but they will appear in the general OOP documentation in due course. This class provides access to virtually all TV functions including the display of, and interaction with, TV menus. To implement this class, a number of AIPS verb subroutines were restructured to make the TV functions appear in separate $YSUB subroutines. These subroutines were also generalized to handle more labeling options, line directions, and so forth. Although you may use these directly in *AIPS* programs, new tasks should consider using the OOP package with one or more instantiations of the TV class.

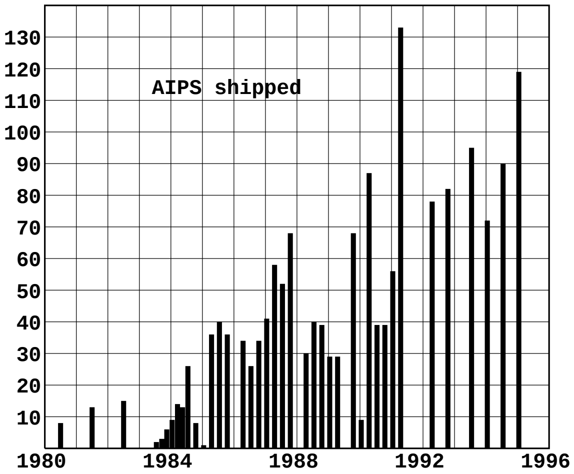
Other matters of interest to programmers in 15JUL95 include

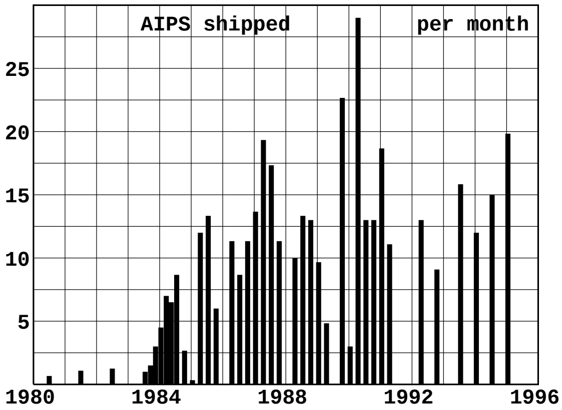
POPS A bug in the processing of the VERB and PSEUDOVB pseudo-verbs used in the NEWPARMS run file caused verb-like symbols to be re-compiled incorrectly and often caused more curious effects as well.

TKDEV The assignment of graphics device numbers was changed from the arbitrary 241–255 and the handling of the device assignment for "remote" graphics devices (namely the user's own terminal) was corrected.

GNU The GNU short copyleft statement was placed in all files in *AIPS*. It must be retained in every file and placed in, for example, the PostScript output files from TEX and dvips.

Perl To improve the speed of *AIPS* start-up scripts, versions of some of them have been written in Perl. They will be used if Perl is available on your system and can make quite a difference in the time it takes to get started.





release	tape	ftp	vms	unix	expfit	binary	total
ALL1980	8	0	8	0	0	0	8
ALL1981	13	0	13	0	0	0	13
ALL1982	15	0	13	2	0	0	15
15JUL83	2	0	2	0	0	0	2
15SEP83	3	0	3	0	0	0	3
15NOV83	6	0	6	0	0	0	6
15JAN84	9	0	9	0	0	0	9
15MAR84	14	0	14	0	0	0	14
15MAY84	13	0	13	0	0	0	13
15JUL84	26	0	14	12	0	0	26
15OCT84	8	0	8	0	0	0	8
15JAN85	1	0	1	0	0	0	1
15APR85	36	0	35	1	0	0	36
15JUL85	40	0	17	23	0	0	40
15OCT85	36	0	36	0	0	0	36
15APR86	34	0	32	0	2	0	34
15JUL86	26	0	25	1	0	0	26
15OCT86	34	0	20	14	0	0	34
15JAN87	41	0	29	11	1	0	41
15APR87	58	0	43	13	2	0	58
15JUL87	52	0	26	26	0	0	52
15OCT87	68	0	45	23	0	0	68
15APR88	30	0	10	19	1	0	30
15JUL88	40	0	20	20	0	0	40
15OCT88	39	0	17	22	0	0	39
15JAN89	29	0	11	18	0	0	29
15APR89	29	0	17	12	0	0	29
15OCT89	68	0	23	45	0	0	68
15JAN90	9	0	1	7	1	0	9
15APR90	87	0	35	52	0	0	87
15JUL90	39	0	12	27	0	0	39
15OCT90	39	0	13	26	0	0	39
15JAN91	56	0	16	40	0	0	56
15APR91	115	18	19	114	0	0	133
15APR92	37	41	0	78	0	0	78
15OCT92	24	58	0	82	0	0	82
15JUL93	33	62	0	95	0	0	95
15JAN94	16	56	0	72	0	0	72
15JUL94	48	42	0	90	0	37	90
15JAN95	66	53	0	119	0	80	119
total	1347	330	606	1064	7	117	1677

AIPS Distribution History

At a time when we are changing the method of distributing *AIPS*, it is good to look back and see the history of *AIPS*' distribution. Ernie Allen has prepared a list of the number of copies of *AIPS* given away by release. This list, which appears on the previous page, may be plotted as *AIPS* shipped for each release or as *AIPS* shipped per month. These plots also appear on the previous page and suggest that the demand for *AIPS* has been approximately constant in copies per month.

Note the rapid acceptance of ftp and binary forms of release and the gradual overtaking of VMS by Unix. We have not shipped a VMS system in rather a long time and shipped only a few in the last release in which VMS was fully supported. That same release was popular for Unix, mostly because it was around for a long time. More copies of `15JAN95` *AIPS* were shipped than has been usual for a 6-month release, perhaps because some sites took copies for more than one operating system.

AIPS Publications and the World-Wide Web

The *World-Wide Web* (WWW) is a method for sending and receiving hypertext over the Internet network and has been made easy to use by clients such as *NCSA Mosaic, Netscape, Arena,* and *Lynx*. NRAO is among the many institutions which now offer informative Web pages and networks of additional information. The NRAO "home" page is at the Universal Resource Locator (URL) address

`http://www.nrao.edu/`

The *AIPS* group home page may be found from the NRAO home page or addressed directly at URL

`http://www.cv.nrao.edu/aips/`

This page points at basic information, news items about *AIPS*, the PostScript text of recent *AIPSLetter*s, patch information for all releases after `15JAN91`, the latest *AIPS* benchmark data from various computer systems, copies of `CHANGE.DOC` for every release since `15JAN90`, *all* relevant *AIPS* Memos, *every* chapter of the *CookBook*, and all recent quarterly reports to the NSF. There is even a tool to let you brouse the `15JAN96` versions of all help/explain files. We recommend that you check this URL occasionally since it changes when new software patches, revised *CookBook* chapters, and new *AIPS* Memos are released.

Since there were no new *AIPS* Memos in the last six months (Number 90 is nearly ready), we will not repeat the usual information about the Memo series here. Since some Memos are not available electronically and others do not yet have computer readable figures, you may wish to write for a paper copy of these. To do so, use an *AIPS* order form or e-mail your request to `aipsmail@nrao.edu`. If you cannot use the Web, you can still use `ftp` to retrieve the Memos, *CookBook* chapters, etc.:

1. `ftp aips.nrao.edu` (currently on `192.33.115.103`)
2. Login under user name anonymous and use your e-mail address as a password (*yourname*`@` will do; ftp will fill in the machine you are using).
3. `cd pub/aips/TEXT/PUBL`
4. `get AAAREADME` and read it for lots more information.
5. `get AIPSMEMO.LIST` for a full list of *AIPS* Memos.

Patch Distribution

Since *AIPS* is now released only semi-annually, we make selected, important bug fixes and improvements available via *anonymous* `ftp` on the NRAO cpu `aips.nrao.edu` (now located on `baboon` which is `192.33.115.103`). Documentation about patches to a release is placed in the anonymous-ftp area `pub/aips/`*release-name* and the code is placed in suitable subdirectories below this. (The patches and their documentation are also available on-line via the World-Wide Web.) Reports of significant bugs in `15JAN95` *AIPS* were not numerous, so some of the patches were actually for new or improved code rather than bug fixes. The documentation file `pub/aips/15JAN95/README.15JAN95` mentions the following items:

UVINIT The low-level basic routine `UVINIT` had an error allowing it to wrongly conclude that it could do fast I/O. It then set its safety margin to 0, checked the buffer size, changed the I/O method, and left the safety margin *wrongly* at 0. This was triggered by a combination of circumstances starting with `NVIS` equal to an integer multiple of 256.

CVEL An error in the use of the same array in two parts of the task caused `CVEL` to flag more and more channels as it ran through the data. This would only occur for data from the VLBA correlator when bandpass calibration was requested.

SWPOL The `GEODLY` array in `SWPCAL` is not dimensioned to cope with the 6-term polynomials used for VLBA data. `SWPOL` is therefore likely to crash when working with VLBA polarization data.

AIPS `PSEUDO` made errors when handling the pseudoverbs `VERB` and `PSEUDOVB` for previously defined symbols. The errors would cause the procedure containing the pseudoverb to be declared a verb or pseudoverb with some verb number such as 4 (an `=` or) the one trying to be declared.

SETPAR messed up the setting of assigned users. The `SP` file only supports 15 disks for this and applies to disk 1 now. Changed it to limit the disk numbers to 1 through `MIN (NVOL, 15)`.

SYSETUP did not handle the symbolic linking of the gripes (`GR`) and password (`PW`) files due to a couple of missing quote marks.

patching The file `$SYSLOCAL/USESHARED` was inadvertently included in the binary distribution for SunOS and Solaris systems. This causes the `COMLNK`s to fail when any patch or rebuilding is attempted as `LINK` attempts to use shared libraries which are not included in the distribution.

AIPS REMOTE The functionality of being able to display graphics on one's Tektronix or compatible terminal was broken. In previous releases since `15APR92`, devices 241-255 were reserved for possible "`REMOTE`" tek devices for `AIPS` numbers 1 through 15. Now, it is assumed that the first 35 devices *beyond* the last configured TV device (as set in the `SP` file via `SETPAR` or `SETSP`) are reserved for these devices.

SDGRD failed to place the `STOKES` adverb into the `COMMON` used by the data reading routines. Therefore, it always did Stokes `'I'`.

SDGRD had three addressing bugs in the in-core gridding. One could have caused data to overwrite the gridding function in the "AP" memory. This should have produced obvious problems. Perhaps various round-downs kept the gridding routine from actually doing this. The second left very little room in the "AP" for data and could have hit limits where it tried to do many channels and then said there was no room for the data. The third was the most serious: an extra "data sample" was gridded for each channel in the group using whatever was in the AP in that range of addresses. In the case in which this was found, values of 2, 4, 6, and 8 were gridded in some of the channels and appeared as 5x5 blocks of cells. The AP memory being used was probably correctly used in doing uniform weighting, so these values are likely to be counts of samples. Thus other integer-like values can occur. The weight image showed very large values at the affected pixels since the gridding routine was called numerous times (every 63 samples in my case). A minor change was made in the test to decide whether to do a weight map (> 2 was changed to ≥ 2 in the code and inputs to match the help).

SDGRD Improved the task to offer the option of computing an image of $1/\sigma^2$ where σ is the correctly computed expected rms in the gridded image (given that the data weights are all the same constant divided ny rms^2 where rms is the data sample's expected rms).

WTSUM New task of interest to single-dish users (and perhaps others) particularly in conjunction with the `SDGRD` change above. It does a weighted sum of two images using two images of weights or of rms's.

OTFUV Added the capability to specify an averaging interval and an output interval. The main benefit is a reduction in disk needs.

Note that we do not revise the original release tapes or `tar` files for patches. No matter when you received your `15JAN95` "tape," you must fetch and install these patches if you require them. Information on patches and how to fetch and apply them is also available through the World-Wide Web pages for *AIPS*. As bugs in `15JUL95` are found, the patches will be placed in the `ftp`/Web area for `15JUL95`. No matter when you receive your `15JUL95` "tape," you must fetch and install these patches if you require them.

Obtaining *AIPS* and the GNU General Public License

We have decided to make *AIPS* available via anonymous ftp under the GNU General Public License, the meaning of which will be spelled out later in this section. The installation of *AIPS* will now proceed something like the following example:

We assume that you have created an account for *AIPS* with a root directory called `/AIPS`. Then do

```
home_prompt<601> cd /AIPS
home_prompt<602> ftp aips.nrao.edu
Connected to baboon.cv.nrao.edu.
220 baboon FTP server (Version wu-2.4(1) Fri Apr 15 12:08:14 EDT 1994) ready.
Name (aips.nrao.cv:egreisen): ftp
331 Guest login ok, send your complete e-mail address as password.
Password: egreisen@
230- This is the National Radio Astronomy Observatory ftp server for the
230- AIPS, AIPS++, and FIRST projects.  Your access from primate.cv.nrao.edu
230- has been logged, and all file transfers will be recorded.  If you do not
230- like this, type "quit" now.  Counting you there are 1 (max 20) ftp users.
230-
230- Current time in Charlottesville, Virginia is Mon Jul 17 10:18:46 1995.
230-
230-
230-Please read the file README
230-  it was last modified on Wed Mar  8 14:01:24 1995 - 131 days ago
230 Guest login ok, access restrictions apply.
ftp> cd aips/15JUL95
250 CWD command successful.
ftp> get README
200 PORT command successful.
150 Opening ASCII mode data connection for README (nnnn bytes).
226 Transfer complete.
local: README remote: README
nnnn bytes received in T seconds (5 Kbytes/s)
ftp> get INSTALL.PS
200 PORT command successful.
150 Opening ASCII mode data connection for INSTALL.PS (mmmmm bytes).
226 Transfer complete.
local: INSTALL.PS remote: INSTALL.PS
mmmmm bytes received in TT seconds (5 Kbytes/s)
ftp> binary
200 Type set to I.
ftp> hash
Hash mark printing on (8192 bytes/hash mark).
ftp> get 15JUL95.tar.gz
200 PORT command successful.
150 Opening ASCII mode data connection for 15JUL95.tar.gz ( bytes).
226 Transfer complete.
local: 15JUL95.tar.gz remote: 15JUL95.tar.gz
mmmmm bytes received in TTTTT seconds (5 Kbytes/s)
ftp> quit
221 Goodbye.
```

You should type in your name (not `egreisen`) followed by an `@` sign at the password prompt. The `hash` command is optional and may be inappropriate in some versions of ftp; it does give a useful indication of progress in the long `get` in most versions. If you do not have the GNU file compression code (`gzip`), you should `get 15JUL95.tar.Z` instead of the `gz` file.

At this point you should read the README file to review the latest changes, if any, affecting your installation of *AIPS*. You should print out the INSTALL.PS PostScript document and read, at least, its overview section. To create the rest of the /AIPS directory tree, and fill it with the *AIPS* source code

```
cd /AIPS
zcat 15JUL95.tar.gz | tar xvf -
              or
uncompress 15JUL95.tar.Z
tar xvf 15JUL95.tar
```

depending on which of the compressed source code files you fetched.

If you want to get the binary version(s) of *AIPS*, you should read the README file for further directions. They will tell you a procedure to run which will run a second ftp session to fetch the appropriate contents from the $LOAD, $LIBR, MEMORY, BIN, and DA00 areas. You may run this procedure more than once if you need to fetch binaries for more than one architecture. You may also have to run portions of this procedure "by hand" if you encounter reliability problems with the network.

You will then have to run the INSTEP1 procedure, as usual, to tell your *AIPS* about your computer environment. A new part of INSTEP1 is its offer to assist you in "registering" your copy of *AIPS*. It will help you complete a registration form and will even e-mail it to us if you want. When we get a registration request, we will enter your information in our user data base and reply with instructions and registration numeric "keys" which you may use to complete the registration process (using SETPAR and SETSP). This may seem cumbersome and onerous, but we have two reasons for doing this. The first reason is to provide us with information about the use of *AIPS*. This information is useful to us to justify, to management and funding agencies, our existence and our need for more employees or computers or disk or whatever. The second reason is a concern about excessive demands on our employees' limited time to provide assistance to sites in installing and running the software. If an excessive demand should arise, information from the registration process will allow us to set priorities among the different sites. This registration is entirely optional. We will use transaction logging in ftp and, hence, know which sites have fetched the code. We will assume that sites which do not register are not "serious" in their use of *AIPS* and we will be unable to provide any assistance to unregistered sites (except, of course, to help them register).

As of the 15JUL95 release, *AIPS* is available under the GNU General Public License. The short statement of this license is in every *AIPS* file, is available on-line via HELP GNU, and is given (once) in the *AIPSLetter* as follows.

```
Copyright (C) 1995
Associated Universities, Inc. Washington DC, USA.

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License as
published by the Free Software Foundation; either version 2 of
the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public
License along with this program; if not, write to the Free
Software Foundation, Inc., 675 Massachusetts Ave, Cambridge,
MA 02139, USA.

Correspondence concerning AIPS should be addressed as follows:
       Internet email: aipsmail@nrao.edu
       Postal address: AIPS Project Office
                       National Radio Astronomy Observatory
                       520 Edgemont Road
```

Charlottesville, VA 22903-2475 USA

You should have received the GNU General Public License from several sources, most notably GNU themselves with their `emacs`, `gcc`, and numerous other software products. Since *AIPS* now applies that license to itself — and intends to import and use other GNU-licensed routines — we also include the full license text on-line via **EXPLAIN GNU** and, once, in the *AIPSLetter*:

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for non-commercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY

GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
   'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items — whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.