AIPS MEMO NO. 35

AIPS GRIPES Procedures

Donald C. Wells, Eric W. Greisen, Nancy D. Wiener

National Radio Astronomy Observatory [*]
Edgemont Road, Charlottesville, VA 22903-2475
(804)296-0211, FTS=938-1271, TWX=910-997-0174

1 June 1985

Abstract

This memo documents the procedures used to operate the GRIPES system. It is divided into three parts: a discourse on gripes, a description of the system by which they are to be managed, and a detailed manual describing the tools provided to do the management.

CONTENTS

[*] The National Radio Astronomy Observatory is operated by Associated Universities, Inc., under contract with the U. S. National Science Foundation.

1 INTRODUCTION

The AIPS GRIPES system is the most reliable and efficient means for AIPS users to communicate with the AIPS programmers. It is intended to be used for software problem reports and for suggestions for improvement of the software. The automatic tools which have been constructed are intended to enable the AIPS group to cope with large numbers of gripes and the organized, machine readable form of the information facilitates systematic processing and information retrieval. This memo is intended to document the procedures used in Charlottesville for gripe collection, cleanup, typesetting, and distribution. Documentation of these procedures will facilitate transferring the responsibility for the operation of the system to new persons in the future.

2 THE AIPS GRIPES SYSTEM

2.1 Properties Of The Gripe Files

The gripe file on an AIPS system has the name GR100000. It is not a simple text file. Instead it contains a mixture of text and binary information. The format of this file is documented in the file DOCTXT:MV2C06GR. Its directory has a fixed size, which implies that the file holds a finite number of gripes. The number is 83 and, when it is reached, the gripe routines apologize and refuse to accept any more gripes. The gripe files should be cleaned out regularly to avoid this problem. Gripe files have been damaged during installation of new versions of AIPS; thus it is good practice to get the gripes before an installation.

2.2 AIPS Commands And User Programs

There are two programs in the AIPS system by which users may enter and examine gripes. They are AIPS itself and a stand-alone program GRIPR. On VAX systems, these programs are invoked by the procedures @[AIPS]AIPS and @[AIPS]GRIPE, respectively.

Gripes are entered on AIPS systems by using the verb "GRIPE" in either AIPS or GRIPR. The program responds by issuing instructions and prompting for needed information such as name, address, phone, and the gripe itself.

The current set of gripes on a system may be examined in two ways using either AIPS or GRIPR. To obtain an index enter the command

```
        GRINDEX                         (print index of gripes)
```

Note how many gripes there are. If you want to read them and are using AIPS, issue the following command

```
        FOR JOBNUM = 1 TO 7; GRLIST; END
```

(assuming there are 7 gripes) and they will be listed on the terminal. (The FOR loop is not available in GRIPR.) An individual gripe may be listed, in either program, by setting JOBNUM to the desired gripe number and issuing the GRLIST command. The information displayed on the terminal by GRINDEX and GRLIST may be printed using the verb PRTMSG.

2.3 Programs GRITP And GRTOTEX

There are two stand-alone programs to assist in processing gripes. GRITP is for use by AIPS Managers at sites not serviced regularly by NRAO (CV) personnel. It converts the contents of the GR100000 disk file into a FITS-like format on tape suitable for mailing to Charlottesville. The best time for doing this is when the Manager is returning a tape and shipping mailer to request a new version of AIPS. A blank is provided on the Order Form to tell us that there are gripes on the tape. (See Section 4.1.2 for an example of the use of GRITP.)

The other program is intended to be used on the NRAO VAXes to convert the local GR100000 file and tapes written by GRITP to VAX text files in a preliminary form suitable for editing and typesetting. This program is not written in a transportable fashion.

2.4 TEXsetting The Gripes

The gripes are TEXset using TEX. We use TEX for gripe files for two main reasons:

- All gripes are read by at least three people in CV; many are read by four people. Several of the people read them more than once. In addition, copies of all gripes are placed in the AIPS Caiges for users to read. Typesetting the gripes makes it easier for all of these people to read them.

- It is very useful to draw clear distinctions between the names of systems, names of programs, input typed into programs, and output emitted by programs. TEX assists in making these destinctions through the use of multiple fonts. In addition, the ability to use mathematical notation in discussions of algorithms is frequently helpful.

Furthermore, TEX macros allow us to alter the output form depending on the intended reader. Subsidiary, diagnostic information is provided for the person answering the gripe, but is suppressed when the gripe response is sent to the griper and placed in public notebooks.

2.5 Handling The Gripes

The process by which gripes are supposed to be managed is outlined in the next section and described in gory detail in following sections. In its simplest form, it goes as follows: Gripes are collected once a week from all NRAO computers and all tapes received during that week. They are typeset, read, and assigned to respondents promptly. A copy of the received gripe and any preliminary answer is mailed to the griper when the editing is reasonably complete. Answers to gripes are entered in the files as they are received by the gripe manager. At the next freeze date for AIPS, all gripes received during the quarter are printed out and a copy sent to the griper. Some of these will be complete, some partially answered including a "to be continued" ('tbc'), and some will, unfortunately, still have no answer. At the same time, any new answers to older gripes will also be sent to the gripers. All answers are reviewed twice before being mailed. After the quarter's answers have been sent, all gripe TEX files are combed and the tbc's separated from the completed gripes. All tbc's are then printed and redistributed to the people who are supposed to answer them.

In addition to mailing responses to the individual gripers, all gripes are displayed in notebooks kept in the AIPS Caiges in Charlottesville and at the VLA. (They make fascinating and informative reading.) We also add all people who have submitted a gripe to our main mailing list and send them quarterly copies of the AIPSLETTER. We do this since we take the submission of a gripe as an expression of interest in AIPS and its development.

3 GRIPES MANAGEMENT

In the following three sections, we outline a procedure by which we hope to manage the large volume of gripes in a responsive and responsible manner. At this writing, the system has not been tested fully and, hence, may require modifications.

3.1 On A Weekly Basis

1. Nancy to fetch gripes on Monday morning from:
   - Charlottesville Modcomp
   - Charlottesville VAX (CVAX::)
   - VLA VAX 1 (AIPS::)
   - VLA VAX 3 (VAX3::)
   - Tucson VAX (TUCVAX::)

   creating a GZiiii.TEX file for the week (where iiii is the first gripe number in the file). She then appends any gripes received on tape from other sites during the previous week and types in any received by mail.

2. Nancy begins TEX editing GZiiii.TEX.

3. When GZiiii.TEX will get through TEX, DVIQMS, and LAS at all, Nancy gives Eric a copy and continues to improve the editing.

4. Meanwhile, Eric reads everything, marking the assignments and keywords and trapping any especially important ones. Answers will not normally be generated at this time, but may be if they are short or if the problem is especially important.

5. Eric returns the marked copy to Nancy and she enters the assignments, keywords, and answers (if any) in the file and completes the typesetting.

6. Nancy runs 2 copies of each, mailing one to each griper and one to the person who is supposed to answer the gripe. Note that the one to the griper should be in the form generated by FINAL=T and the one to the respondent in the form generated by FINAL=F (see section 4.2.5.1).

7. Nancy appends GZiiii.TEX to the GXjjjj.TEX file being maintained for the current quarter (and changes \newgripe to \tbonew on all the newly appended gripes).

8. Nancy enters new gripers in WHOGETS.ADR.

3.2 On An Irregular Basis

1. Respondents write their answers out on the gripes they have been given and return them to Nancy.

2. Nancy enters the answers in the appropriate files. GSnnnn.TEX files are not changed except under very special circumstances. GXjjjj.TEX and the GTnnnn.TEX files may be changed freely (using CHKOUT, of course). When a gripe response in a GTnnnn.TEX file is changed, the string "\changed" must be added to the response. This will flag that gripe so that it may be sent to the griper at the next quarterly update. Whenever some answer is put into a gripe, the macros "\newgripe" and "\tbcnew" (which generate the standard "we have no real answer yet" responses) should be deleted.

3.3 On A Quarterly Basis

1. A bit before the magic date, Nancy reminds people to get their answers in to her and she enters all answers which she has received.

2. At, or a bit after, the magic date, Nancy extracts all gripes which have been modified during the quarter from each of the GTnnnn.TEX files and runs off a TEX copy of each.

3. Nancy then runs off a TEX copy of the current GXjjjj.TEX (with \finalrun{T}) and gives all the TEX outputs (steps 2 and 3) to Don.

4. Don reviews the answers promptly, marking the printout if needed, and then gives them to Eric who also reviews their contents and enters appropriate items into the Wishlist file.

5. Nancy makes the marked corrections to the GXjjjj.TEX file and the GTnnnn.TEX files. Then she runs GXjjjj.TEX through TEX with \finalrun{T}. She also TEXsets the changed gripes from each of the GTnnnn.TEX files.

6. Nancy makes two additional copies of the TEX outputs using the xerox machine. One copy is filed in the CV Aips Caige, one is sent to Pat Moore for the VLA Aips Caige, and one is sent to the griper.

7. The GXjjjj.TEX file is edited to substitute "\tbc" for "\tbcnew". Then, it is stripped into two files: one named GSjjjj.TEX containing completed gripes and the other named GTjjjj.TEX containing any still having "\tbc."

8. The GTnnnn.TEX files are edited to remove all "\changed" strings. Then all completed gripes are stripped from each GTnnnn.TEX file and inserted in the correct numerical order in the corresponding GSnnnn.TEX file.

9. Nancy runs all GTnnnn.TEX files through TEX, DVIQMS, and LAS (one copy). She sorts the output by the assigned respondent and distributes it. Nancy runs the indexing program and distributes copies of the output to all gripe respondents.

4 HOW THIS IS ALL DONE - THE DETAILS

Currently the gripes are to be retrieved from all AIPS machines, (CVAX, ModComp, AIPS, VAX3, and TUCVAX) once a week, preferably on Mondays, and special attention should be paid to the gripes toward the end of each 90-day cycle so that fixes can be made before the deadline.

4.1 Harvesting The Gripes

The general strategy is to SET DEFAULT to a suitable directory under the gripe manager's login, fetch the individual batches of gripes in with temporary names, and merge them into the larger batch being maintained for the quarter, using the EMACS text editor. Special EMACS "Mock Lisp" macros are provided to assist in cleanup of the gripes, numbering them, and editing typesetting commands into the files. This section concentrates on the commands used to fetch the gripes from the various places.

4.1.1 Copying Gripes From Nodes CVAX, AIPS, VAX3, TUCVAX

The same procedure works with all (VAX) sources of gripes. The idea is to login to an AIPS system and define a logical symbol TARGET into which GRTOTEX will try to write the gripes (using DECnet). In the example below, TARGET sends the gripes into Nancy's usual directory for gripes work. Please note that the syntax of the definition (quotes, colons, and [possibly] use of upper case) is important. An interesting fact is that, should you so desire, it is possible (for complete consistency) to SET HOST to the current machine!

```
        SET HOST hhhh           (CVAX, AIPS, VAX3, or TUCVAX)
        AIPS                    (to log in to account AIPS)
               (For TUCVAX, use Tucson's password)
               (For TUCVAX, say AIPS again to start up AIPS)
        TST                     (enter AIPS version)
        uuu                     (enter AIPS user number)
        GRINDEX                 (get a list of gripes in file)
        FOR JOBNUM = 1 TO u; GRLIST; END  (to list the gripes)
        EXIT                    (exit from AIPS)
        SET DEFAULT TST         (get into [AIPS.15mmm8y] - VMS-level)
        DEFINE TARGET "CVAX""NANCY <password>""::UMA0:[NANCY.GRIPES]"
        SHO LOGICAL TARGET      (to verify target definition)
        RUN [.LOAD]GRTOTEX
        0                       (Tape drive number {I1} 0 => disk)
        <CR>                    (Disk file name {6A4}
                                    <CR> => default: DA00:GR100000.;1)
        GZnhhhh                 (output file name: n=1,2,3,...;
                                    hhhh=CVAX,AIPS,...)
        Y                       (initialize the disk)
               (A password is required on some systems)
        DIRECTORY/SIZE/DATE TARGET:GZnhhhh  (a check, could use "GZ*")
        LO                      (log off, return control to CVAX::)
```

Some systems will not have a TST version available; in such a case, use NEW instead.

As GRTOTEX executes, a list of the gripes will appear on the terminal. If desired, the SET HOST to CVAX and final LO can be omitted for the harvesting of gripes on the main AIPS on CVAX node. If one does not want to initialize the disk just answer "N" to the final question from GRTOTEX. Later the GR file can be initialized with this recipe:

```
        RUN [.LOAD]GRITP
        0                       (Tape drive number {I1} 0 => disk)
        YES                     (Init the file? YES or NO {A4})
```

Initialization causes all current gripes to be deleted from the AIPS gripe file and hence should be done cautiously. However, if they are left for long, then the deferred initialization could cause some newly-added gripe to be lost completely. It is probably best to answer yes to GRTOTEX except when there has been something strange about its output while fetching the gripes.

Note that this procedure may also be run on non-NRAO VAXes if they are connected to CVAX by DECNET or other appropriate methods. CalTech's PHOBOS and DEIMOS VAXes are prime candidates, should the need eventually arise.

4.1.2 Copying Gripes From The ModComp -

The general procedure is to carry the gripes from the ModComp to CVAX node on a tape since there is no reasonable intercomputer link. The procedure for reading them into CVAX is given in the next section. The following recipe writes the tape on the ModComp:

```
                       (mount a scratch tape on Modcomp unit 1)
        JOB                     (for cleanup purposes)
        ASS 5 LO 6 LO
        EXEC GRITP,LMV
        1 (or 2)                (Tape drive number {I1}))
        WRIT                    (WRITe or VERIfy? {A4})
        YES                     (Do a verify pass? YES or NO {A4})
                                (At this point the tape should be
                                   dismounted and copied into CVAX
                                   using the procedure discussed in
                                   the next section. When it is safely
                                   inside the VAX the final question
                                   can be answered:)
        YES                     (Init the file? YES or NO {A4})
        JOB                     (cleanup)
```

If this fails, try: CTRL-A, \ON MT1, \V\R (for terminal 1 only).

4.1.3 Reading A Gripe Tape Into "CVAX" -

This procedure applies to gripe tapes received from all machines, including the ModComp and non-NRAO VAXes:

```
                         (check availability of tape drive(s))
        SHO DEV MTA0            (tape drive 1 - Computer Room)
        SHO DEV MMA0            (tape drive 2 - AIPS Caige)
                         (place tape on available unit)
        SET DEFAULT TST         (get into [AIPS.15mmm8y])
        @[-]MOUNT               (for MTA0:, or @[-]MOUNT2 for MMA0:)
                                ([-] to get from [AIPS.NEW] to [AIPS])
                (SHO LOG TST will display new (current) AIPS version)
        DEFINE TARGET "CVAX""NANCY ‹password›""::UMA0:[NANCY.GRIPES]"
        RUN [.LOAD]GRTOTEX
        1                     ({I1} 0 if disk, 1 if MTA0:, 2 if MMA0:)
        0                       (Files to advance +- {I3})
        GZnsssss                (output file name in "DOCGRIP" {A12}
                                   n=1,2,3...; sssss=site, e.g. MODC)
        DIRECTORY TARGET:GZnsssss        (a check)
        @[-]DISMOUNT            (or @[-]DISMOUNT2)
        DEALLOCATE MTA0:        (or MMA0:)
```

One should generate names which include the site/machine, such as GZ2MODC (ModComp), GZ2JDRLL (Jodrell Bank), GZ2PHOBOS (a CalTech machine), GZ2CVAX (CVAX::) etc. for the temporary GZ files.

4.1.4 Entering Gripes By Hand

Gripes which are submitted in the form of letters or telephone calls may be entered into the gripe files with the editor. The procedure is discussed below in the section on the EMACS editor.

4.2 Using The EMACS Editor On Gripe Files

The files produced by GRTOTEX all have names of the form "nnnnn.TEX", where "nnnnn" is generally of the form "Gnnnnn", as shown in the examples above. In the future, we hope to program the EMACS editor on CVAX node to trigger on such names and enable special support macros for the gripe files. At present, however, these macros must be loaded from the "DOCGRIP" directory immediately after accessing a gripe file via EMACS, before attempting to do any editing on the file. To do this, issue the command: "ESC-X: LOAD DOCGRIP:GRIPES.ML", followed by "ESC-X: GRIPE". (Note that the use of lower case for most commands is acceptable; however, upper case must be used for "GRIPE" when issuing "ESC-X: GRIPE".) This section describes the procedures used with EMACS and the support provided. We assume the usual level of familiarity with EMACS and don't describe EVERY detail.

4.2.1 Merging Gripe Files, Cleaning And Numbering -

The general strategy is to use EMACS to combine the individual gripe files into one, clean it up, and give the gripes successive numbers beginning with one more than the highest previous gripe. The usual steps are in the following sequence:

1. Start up EMACS.

2. Visit the highest numbered existing gripe file. Look at the end of the file to find the highest numbered gripe.

3. "Visit" a file with a name "GZnnnn.TEX" where "nnnn" is one larger than the previous highest number (actually the file is new and so it is not literally visited).

4. Insert the first new gripe file.

5. For each additional gripe file do the following steps:
   - Move to the end of the buffer.
   - Insert the next new gripe file (if any).

6. Move to the end of the file and add gripes from letters and verbal comments as described in the next section.

7. Issue "ESC-X: LOAD DOCGRIP:GRIPES.ML" and "ESC-X: GRIPE" (note upper case must be used for "GRIPE") in order to load the gripes macro. This step is temporary; in the future we hope to have the procedure autoloaded.

8. Issue "ESC-X: CLEANUP". This procedure edits the text extensively:
   - cleans up a several basic TEX commands
   - changes {\tbc} to {\secret{ }\newgripe}
   - deletes TABs
   - deletes leading blanks on lines
   - deletes NLs (new lines) before }
   - deletes NLs (new lines) after {
   - deletes redundant NLs
   - deletes blanks after { and before }
   - replaces VAX#1 with VAX1 and VAX#3 with VAX3.
   - query-replaces "&" with "\&" (you may actually want "\&\ " to get the spacing right)
   - query-replaces "%" with "\%" (or "\%\ ")
   - query-replaces "$" with "\$" (or "\$\ ")
   - query-replaces "#" with "\#" (or "\#\ ")
   - query-replaces "_" with "_" (or "_\ ")
   - query-replaces "^" with "\^" (or "\^\ ")
   - query-replaces "~" with "$\sim$"
   - deletes blanks before . , : ? !
   - query-replaces "<" with "$<$" and ">" with "$>$"
   - query-replaces "*" with "$\ast$".

   The replacement operators are designed to recognize special symbols that have already been cleaned up. They do this by temporarily transforming the patterns to something else which is not recognized by the query-replace patterns. This allows the "cleanup-gripes" command to be executed on a file more than once. The "query-replace" operations listed above allow the person to exercise judgement.

9. Move to the beginning of the file and issue "ESC-X: NUMBER". It will ask for a new starting number (use the "nnnn" from the file name) and will proceed to renumber the gripes.

10. For insurance, copy the edited gripe file, naming the copy GZnnnnbu.TEX, and use PUTBCK to put the file into the "DOCGRIP" directory. Then edit various typesetting features into the text as discussed in a subsequent section.

4.2.2 Entering Gripes By Hand -

Once the gripe files have been merged, additional gripes can be added to the end of the file (just before the "\end"). For each new gripe, insert (^x-^I) the gripe template which is in the file DOCGRIP:GTEMPLATE.TEX. Then fill in the empty TEX brackets with the relevant information. Use the "ESC-Y" keystroke to insert today's date. The template contains comments (% followed by text following the {}s) as a guide to filling in the information. These comments must be deleted after the information has been inserted.

4.2.3 Special Keys And Macros Supported By EMACS -

A gripe file is a special case of a TEX file (name of form G{S|T|X|Z}nxxxx.TEX). A variety of features are supported as a part of being in "TEX" mode. Others are supported for gripe files, which are in "GRIPE" mode. Useful keystrokes and macros which are a part of basic TEX support are:

1. Font selectors:
   - "ESC-I" encases a word with control symbols for "italic" font
   - "ESC-T" encases a word with control symbols for "teletype" font
   - "ESC-S" encases a word with control symbols for "script" font [used for system names].

2. Case selectors:
   - "ESC-U" makes a word be all upper case
   - "ESC-L" makes a word be all lower case
   - "ESC-C" makes a word be "capitalized" (first character upper case, rest all lower).

3. "ESC-A" deletes redundant blanks in neighborhood of cursor (leaves only one blank).

4. "ESC-X: BRACKETS" checks a file for matching brackets and parentheses. Any TEX file (and therefore any gripe file) which cannot pass this test is unlikely to operate correctly with TEX.

5. Additional features which are enabled in "GRIPE" mode are:
   - "ESC-N" positions "next" gripe on the window
   - "ESC-P" positions "previous" gripe on the window
   - "ESC-M" moves forward to next empty TEX bracket ("{}")
   - "ESC-Y" inserts a date string (use for dating answers)
   - "ESC-K" wrap word in \key{} for "keys" mechanism (see sect. 4.2.5.2)
   - "ESC-<line feed>" inserts "\fresh" (to make following text begin on a new line)

6. "EXC-X" (gives ":" prompt below the buffer), followed by "BIND-TO-KEY NAME: CASE-REGION-LOWER KEY: ESC-L" will allow one to convert a group of words to all lower case (useful when gripes have been submitted in all upper case). To use this feature, set the beginning point with "<control>-<space-bar>" or with "<control>-(shift)@" (depending upon the terminal being used) and then move the cursor to the desired end point and then use "ESC-L". (Note: when bound to CASE-REGION-LOWER, "ESC-L" will _not_ work on individual words as it usually does; it will change a large section of words preceding the current word to lower case if used inadvertently without setting the beginning and ending points. USE WITH CAUTION.

4.2.4 Typesetting Details -

Various special characters may be emitted in the "teletype" font with the following expression: "{\tt \char'ooo}", where "ooo" is the octal value of the ASCII code for the character:

1. dollar-sign = \char'044 (a "\$" should also work)
2. at-sign = \char'100 (a "\@" should also work)
3. backslash = \char'134

Several mathematical symbols are often useful in gripes (use these only inside math expressions, i.e., bracketed with dollar signs):

1. \times (multiplied by - a frequently used symbol)
2. \mapsto (maps to)
3. \rightarrow (converges to or approaches)
4. \gg (much greater than)
5. \ll (much less than)
6. \leq (less than or equal to)
7. \geq (greater than or equal to)
8. \pm (plus or minus)

We need a definition for "\neq". A definition for the proportionality symbol used by physicists would be nice too. Note that "512×512" often looks better as "512^2".

Several replacements are nice to execute (perhaps several of them should be in the "cleanup" macro):

1. iterative delete of characters before BS (in "cleanup" but doesn't work)
2. query-replace a "NL-NL" with "NL" (option to merge paragraphs).

4.2.5 Conventions And Features Of The TEX Macros -

The current macros are in the file "DOCGRIP:GRIPEMAC5.TEX", and GRTOTEX inserts into the gripes a command to input this file.

4.2.5.1 The "finalrun" Concept -

A TEX symbol called "\finalrun" is defined with value "T" or "F". The TEX macros for formatting the gripes test this value and produce different layouts depending on it. For example, the field which gives the name of the person assigned to answer each gripe is printed when finalrun is false, but is not printed when it is true. Also, the date when the answer was prepared is printed before the answer when false and after it when true. Certain other values are printed or not depending on \finalrun. The value is set at the beginning of each gripe file by a line of the form:

```
        \def\finalrun{F}
```

4.2.5.2 The KEYs Scheme -

The EMACS macro package defines "ESC-K" to insert "\key{ }" around the word currently under the cursor. If finalrun is F, a box is drawn around the word in the output. If it is T, no box is drawn (i.e., the "key" macro is a do-nothing). We have built a program called GRKEY to read the gripe files and extract the key strings in order to prepare a complete cross-reference listing of keywords in all gripes.

4.2.5.3 The SECRET Scheme -

The "\secret{ }" macro is used for programmer comments. When \finalrun is F, the text inside the { } is displayed prominently, but, when \finalrun is T, it is totally suppressed. Among matters for which \secret is useful are comments from Eric or Don to the respondent and questions from Nancy regarding typesetting, etc.

4.2.5.4 The CHANGED Scheme -

The "\changed" macro is used to indicate that a gripe response has been altered during the quarter. Since all GXjjjj.TEX gripes will be sent to the gripers at the end of the quarter, \changed is useful only in the GTnnnn.TEX files, from which only a few of the gripes will be mailed out in any given quarter. This macro produces no output on the printed page. During the execution of TEX, however, it produces the message "The following gripe number has had a change made during this quarter:" on the terminal just ahead of the gripe page number. The .LIS file produced by TEX may be printed to make a "permanent" record of those gripes which were changed.

4.3 Running TEX On Gripe Files

Here are the commands to run TEX on a gripe file:

```
@DBAO:[TEX.QMS]LOGIN        (if your LOGIN.COM hasn't already)
TEX gz5                         (process "gz5.tex")
DVIQMS gz5                      (convert to "BIT" file format)
LAS gz5.BIT                     (send to the QMS)
```

The DVIQMS command accepts several arguments:

1. DVIQMS gz5/START=1337/PAGES=5 (to print gripes 1337-1341)

2. DVIQMS gz5/COPIES=2 (2 copies of each gripe)
   This option could be given with /START and/or /PAGES.

If the QMS printer is totally fouled up one can press the RESET button on its electronics box and then issue the job control command "INITLAS". This will reset the machine and reload the default fonts for TEX.

4.4 Some Gripe Utility Programs

The source of these programs is in the "AIPPGM" directory and the subroutine package in GRISUB.FOR is in the "AIPSUB" directory. The executable images are in the "LOAD" directory and the procedures are in the "TST" directory. These programs are not coded in the usual AIPS standards.

4.4.1 Program GRCHEK

This program is used to check a gripe file for format and sequencing. The format checks are performed by subroutine GETGRI when it reads the next gripe. GRCHEK itself watches the sequencing and reports any discrepancies. It also gives a final report on the starting and ending gripe numbers in the file. Although its original motivation was just to test the GETGRI subroutine, it has proved to be a useful program in its own right because it performs a simple but effective consistency check on the file. To use:

```
      RUN LOAD:GRCHEK      it will ask for a file name
      GZ1234               it will pad the name with ".TEX" if
                                 you don't. If the file doesn't
                                 exist it will ask you again.
```

4.4.2 Program GRSORT -

This program sorts a gripe file into order of ascending gripe number. Because it uses subroutine GETGRI to read the input file it automatically also checks the formatting of the gripes (analogous to GRCHEK). It suppresses redundant copies of gripes which are found during the sort (they must be EXACTLY identical!). It also has an option to suppress non-identical duplicate gripe numbers; this is useful when concatenating files which have been changed.

GRSORT does a simple insertion sort in a big array in memory which has room for 500 gripes of up to 3000 characters each (these limits are Fortran-77 parameters in the program).

To execute:

```
      RUN LOAD:GRSORT
      GZ1234                  input file name (pads with ".TEX")
      GZ1234                  output file name (must enter!)
      F                       suppress ALL redundant gripes? T or F
```

4.4.3 Program GETTBC -

This program reads a gripe file and splits it into two files. The split test is made on a string which the user inputs. Just as for GRCHEK and GRSORT, the GETGRI subroutine reads the file and performs a consistency and format check on the input. The current version only implements one split string (per execution). To run it, enter:

```
RUN LOAD:GETTBC
GZ1234                   input file name (pads with ".TEX")
GQ1234                   regular output name (no split string)
GT1234                   "split" output file name
T                        \finalrun T or F for regular output file
F                        \finalrun T or F for "split" output file
\tbc                     split string, nonblank, up to 20 chars
                            (\changed would also be used)
```

4.4.4 Program GRKEY -

This program reads one or more gripe files and prepares sorted listing and TEX files. GRKEY requests first the "person name for a special listing" and interprets a blank answer to mean that no TEX output is desired. The string "Everyone" causes it to build a TEX file containing all gripes in the input files sorted first by the person to whom they have been assigned and secondarily by the \key strings in the gripes. Any other string is taken to restrict the TEX output to those gripes assigned to the person whose name is given by the string. Only "to-be-continued" gripes are written to the TEX file. The basic listing consists of any errors or semi-anomalous conditions found in the input files and the sorted list of selected gripes (including both final and tbc gripes) by person, key and number. Input flags allow the user to request additional printed output. The "complete" list is a sequential list of all gripes and of gripes separated by category (final, tbc, tbcnew and changed). The "All key" list is a list of all gripes sorted by the key string. The "site/griper" list is first a list of all gripes sorted by the string identifying the site at which the gripe was entered and then a list of all gripes sorted by the string identifying who entered the gripe. The user may specify, using VAX directory and wild-card conventions as needed, which gripe files are used as input. The default is "DOCGRIP:GT%%%%" with "GX%%%%" added when no TEX output is requested. Just as for GRCHEK, GRSORT, and GETTBC, the GETGRI subroutine reads the file and performs a consistency and format check on the input. To execute:

```
RUN LOAD:GRKEY
Everyone                 select TEX output (blank => none)
F                        request "complete" lising (F => no)
T                        request "All key" listing (T => yes)
T                        request "site/griper" listing
DOCGRIP:GT%%%%           input file specification
```

4.4.5 Procedures GRCHANGE, GXTBC, GRTBC, GTTEX -

The steps listed in Section 3.3 (GRIPES MANAGEMENT - On a Quarterly Basis) require a variety of potentially confusing operations to be performed on files with standard names and contents. To reduce the confusion, several procedures have been written to perform these standard operations. GRCHANGE is run to produce TEX copies of changed gripes to be mailed to the gripers. After the quarter's mailings are complete, GXTBC and GRTBC are run to rearrange the files. GTTEX is run to produce an overall index of gripes and to produce fresh copies of all to-be-continued gripes. All of the procedures clean up most of their scratch files and display a directory of any that must remain long enough for the print jobs to work.

The procedure GRCHANGE is designed to extract any gripe containing the \changed macro and produce a TEX copy of it with a user-selected value of \finalrun (having T as a default). The procedure assumes that a file GTnnnn.TEX resides in the default directory and contains some \changed gripes. (This can be determined via a string search with EMACS.) GRCHANGE produces a variety of temporary files, but deletes them all (except the .BIT file) before exiting. It is invoked by

```
                    @TST:GRCHANGE nnnn
```

The procedure GXTBC is designed to split the current quarter's GXnnnn.TEX file into new GSnnnn.TEX and GTnnnn.TEx files which will be stored in the "DOCGRIP" directory via PUTBCK. It assumes that GXnnnn.TEX resides in the default directory and that all \newgripe's and \tbcnew's have been converted (via EMACS) into \tbc's. GXTBC runs GETTBC to make a new GSnnnn.TEX file containing completed gripes and a new GTnnnn.TEX containing ones to be continued. It runs GRCHEK on the two output files and is invoked by

```
                    @TST:GXTBC nnnn
```

If " T" is given after the nnnn above, the procedure will also make a TEX copy (with \finalrun{F}) of the new GTnnnn.TEX. The various scratch files (except the .BIT plot file) are deleted by the procedure, but the original GXnnnn.TEX file is left alone.

The procedure GRTBC is designed to deal with old tbc files which contain newly-completed gripes. It assumes that both the GTnnnn.TEX and the corresponding GSnnnn.TEX files have been checked out from "DOCGRIP" using CHKOUT and now reside in the default directory. It extracts all completed gripes from the GT file, concatenates them with the GS file, resorts the GS file, and runs GRCHEK on the results. It is invoked by

```
                    @TST:GRTBC nnnn
```

If " T" is given after the nnnn above, the procedure will also make a TEX copy (with \finalrun{F}) of the new GTnnnn.TEX. The various scratch files are deleted by the procedure, but the several versions of the GSnnnn.TEX and GTnnnn.TEX files are not purged.

The procedure GTTEX is designed to produce sorted index listings of all gripes and to make new TEX copies of all outstanding gripes (sorted by respondent, key and number). It assumes that all gripe files have been restored to the "DOCGRIP" area. To make an index listing of all gripes enter:

```
                    @TST:GTTEX
```

which will produce a long terminal session followed by a print to the QMS laser printer. To make a new copy of all active gripes, sorted by respondent, enter

```
                              @TST:GTTEX   TEX
```

which will produce all the listings and run TEX, DVIQMS, and LAS on the sorted gripes. This procedure deletes its scratch files but leaves several output files. A directory listing of these is shown at the end of the procedure.

4.4.6 Subroutines In GRISUB

Subroutine GETGRI is the most important routine in this collection used in the various programs. Its function is to recognize the next gripe in the file and deliver it into a big string variable. It does this by collecting each group of non-blank lines terminated by a blank line and then checking that they constitute a gripe. In this process it notes and skips over "\input", "\finalrun", and "\end" commands and redundant blank lines before the gripe. A valid gripe begins with "\count0=" and contains the string "\gripe". It contains only one each of "count0" and "gripe". The current version also checks brackets!

Subroutine WRIGRI writes a gripe (which is a big string variable) onto the designated output file. Note that GETGRI compresses gripes into a big string by replacing all newlines with the string "|||" and WRIGRI undoes this replacement.

Subroutine PADTEX adds ".TEX" to file names if no file type is present.

4.5 Handling The Quarterly Operations

Every quarter the gripes are run through a sequence of operations which insure that all new gripe responses are sent to the gripers and that all gripe respondents are reminded of their outstanding problems. These operations are to be performed on, or soon after, January 15, April 15, July 15, and October 15 of each year. They divide into three major classes whose detailed steps are listed in the following subsections.

4.5.1 Gripe Preparation And Review -

1. A bit before the magic date, Nancy reminds people to get their answers in to her. She uses CHKOUT to get the appropriate GTnnnn.TEX files and the GXjjjj.TEX file from "DOCGRIP", uses EMACS to enter all answers which she has received, and uses PUTBCK to return the edited files to "DOCGRIP".

2. At, or a bit after, the magic date, and after all answers have been entered, Nancy uses CHKOUT to check out the current GXjjjj.TEX file. With EMACS she substitutes "\tbcnew" for

"\newgripe" which changes the wording of the "we have no real answer yet" answer. Nancy then runs off a TEX copy of the current GXjjjj.TEX (with \finalrun{T}) and gives all the TEX outputs to Don. Then she uses PUTBCK to put the file back into "DOCGRIP".

3. Nancy constructs a list of all GTnnnn.TEX files which have been modified during the quarter. To do this, she can use EMACS to visit each of these files (in "DOCGRIP") and simply search for the string "\changed". All operations described below on GTnnnn.TEX files apply only to those which are found to contain at least one such string.

4. Nancy copies the GTnnnn.TEX files to her area (CHKOUT should not be used). (A "COPY DOCGRIP:GT*.TEX *" will do if it is followed by a deletion of the unwanted files.) Using "@TST:GRCHANGE nnnn" on each of the GTnnnn.TEX files, she extracts all gripes which have been modified during the quarter (testing for "\changed") and runs off a TEX copy of each, giving the printed outputs to Don.

5. Nancy should check that her directory is back to normal at this point by (1) running OUTPRT to make sure that she has no files checked out and then (2) deleting all GT*.TEX and GX*.TEX files in her area.

4.5.2 Gripe Correction And Distribution -

1. Don reviews the answers promptly, marking the printout if needed, and then gives them to Eric who also reviews their contents and enters appropriate items into the Wishlist file.

2. Nancy uses CHKOUT to copy any of the GTnnnn.TEX files and the GXjjjj.TEX file for which Don and/or Eric have indicated changes. She employs EMACS to make these changes and returns the files to "DOCGRIP" with PUTBCK.

3. Nancy copies GXjjjj.TEX from "DOCGRIP" (CHKOUT should not be used) to her area, makes sure that \finalrun{T} is at the top of the file, and then runs it through TEX.

4. Nancy copies the GTnnnn.TEX files to her area (CHKOUT should not be used). (A "COPY DOCGRIP:GT*.TEX *" will do if it is followed by a deletion of the unwanted files.) Using "@TST:GRCHANGE nnnn" on each of the GTnnnn.TEX files, she extracts all gripes which have been modified during the quarter (testing for "\changed") and runs off a TEX copy of each.

5. Nancy makes two additional copies of the TEX outputs using the xerox machine. One copy is filed in the CV Aips Caige, one is sent to Pat Moore for the VLA Aips Caige, and one is sent to the griper.

6. Nancy should check that her directory is back to normal at this point by (1) running OUTPRT to make sure that she has no files checked out and then (2) deleting all GT*.TEX and GX*.TEX files in her area.

4.5.3 Gripe File Preparation For The Next Quarter

1. Nancy copies the GXjjjj.TEX file to her area and, with EMACS, substitutes "\tbc" for "\tbcnew". Then she does "@TST:GXTBC jjjj" to strip the file into two files: one named GSjjjj.TEX containing completed gripes and the other named GTjjjj.TEX containing any still having \tbc. Finally she does "PUTBCK DOCGRIP:GSjjjj.TEX", "PUTBCK DOCGRIP:GTjjjj.TEX", and "REMOVE DOCGRIP:GXjjjj.TEX" followed by a "DEL G*jjjj.TEX;*" to clean up her directory.

2. For each for the modified GTnnnn.TEX files she:

    1. Obtains the needed files with "CHKOUT DOCGRIP:GTnnnn.TEX" and "CHKOUT DOCGRIP:GSnnnn.TEX".

    2. Edits GTnnnn.TEX files to remove all "\changed" strings.

    3. Does "@TST:GRTBC nnnn". This procedure strips out all completed gripes, concatenates them with the corresponding GSnnnn.TEX file, and sorts the gripes in the GSnnnn.TEX file into numerical order. It also runs GRCHEK on the output files to make sure that nothing bad has happened.

    4. Puts the changed files back with "PUTBCK DOCGRIP:GSnnnn.TEX" and "PUTBCK DOCGRIP:GTnnnn.TEX".

    5. Cleans up with "CPURGE DOCGRIP:G*nnnn.TEX" and "DEL G*nnnn.TEX".

3. Nancy should check that her directory is back to normal at this point by (1) running OUTPRT to make sure that she has no files checked out and then (2) deleting all GT*.TEX and GX*.TEX files in her area.

4. Nancy does "@TST:GTTEX TEX". This procedure runs all GTnnnn.TEX (including the new GTjjjj.TEX) through TEX, DVIQMS, and LAS (one copy) and produces a listing which is an index of all outstanding gripes (presented in several orders). This procedure, by calling GRKEY, sorts the output by the assigned respondent and subject.

5. Nancy distributes the appropriate TEX output and copies of the index to all gripe respondents.

5 THINGS WE NEED

This is a list of action items for the improvement of the procedures, programs, EMACS macros, and TEX macros:

1. Finish the program to resort WHOGETS.ADR into alphabetical order. Retrieve mailing addresses of users from WHOGETS by using user number in gripes?

2. Fix EMACS to auto-execute and auto-load more.

3. Write the documentation file MV2C06GR.

4. The use of GRITP by system managers should be documented.

5. The "cleanup" macro needs more smarts including compressing gripe heading and name/date of response.

6. More smart keys:
   - "ESC-P" and "ESC-N" should position on "\count0" and should set _region_ on the current gripe. (May not be desirable - may be tried on an experimental basis.)
   - "ESC-G" to emit "text garbled in transmission for unknown reasons".

7. Gripe index program to execute from command language level like GRTOTEX and GRITP. Better yet: GRTOTEX and GRITP to be _tasks_.

8. Maybe we need a log file to record handling of the files (when received, when sent, etc.).