

A Programmer's Guide to The *AIPS* Gripes Database

W. D. Cotton, Dean Schlemmer¹

20 February 1992

Abstract

The *AIPS* Gripes database is an **emacs**-based tool for maintaining all of the many user complaints and suggestions received by the project over its lifetime. The present document is an in-depth guide for programmers and the Gripe Manager to maintain the Gripes Database and to maintain the database tools. It assumes a working knowledge of the User's Introduction to the *AIPS* Gripes Database and of **emacs**.

7 Introduction

The *AIPS* Gripes Database is built within the **emacs** editor using the **lq--text** string search system to aid in the selection of files. Most of the functions are implemented as **emacs** Lisp routines with **lq--text** routines being invoked as subprocesses to **emacs**. The two purposes of this document are (1) to describe how to enter new gripes, modify or answer gripes and the distribution of gripes and (2) to document the software system used for the Gripes system.

8 Access to the gripes database

Programmer/Gripe manager access to the gripes database is through the account (username) **aipgripe** on **zia**. (If you wish only to *use* the database (read-only privilege), you may also log into virtual machine **gripe**.) The gripes database system is started using the procedure **viewgripe**. Programmers/Gripes Managers then have the full user functionality available, as well as a variety of management tools which are not documented in the on-line on-line help and are only available when logged into account **aipgripe**.

9 Structure of the database

The Gripes database system consists of a number of parts: the text of the gripes, user address information, and **lq--text** information for both of these types of information. The gripe and user information is kept in a collection of files, one per gripe or user. These files consist of coded keywords, starting with **_** (underscore) and ending with **:** (colon) and containing only upper-case letters and **-** (minus). This allows these files to be decomposed into a list of keyword-value pairs which can then be manipulated by lisp functions. Gripe

¹TeXset and updated to 1994 by Eric W. Greisen.

files have names of the form `gripe $xxxx$.grp` where $xxxx$ is the gripe number and live in `~aipgripe/gripe` or `~gripe/gripe`, or `/aips1/gripe/gripe`. User information files have names of the form `username.info` and live in `~gripe/adr`. `lq-text` files for rapid searches of either of these databases are kept in `~gripe/gripedir` and `~gripe/adrdir` for gripes and user information, respectively.

10 Entering new gripes

Gripes may be entered into this system through three possible routes: conversion of old TeX-format gripes, conversion of new *AIPS* GR files, or by interactively entering new gripes in `viewgripe`. Each of these are executed using lisp functions in `~gripe/lisp/newgripes.el` which is set to auto load. Since the inclusion of a new gripe involves assigning a new, unique gripe number (using function `gripe-number`), no other routes are allowed. (Most of the gripes lisp functions have documentation available through the `emacs describe-function` feature.) The functions for entering new gripes are described in the following:

(A) The function `convert-texgripes` reads the older TeX-based gripe files (`GZ`, `GX`, `GS`, and `GT` files — harvested and typeset) and extracts gripes and user information and enters them into the new system. A user information file will only be added (created) if none exists for a given user and will need to be edited to put needed linefeeds into the addresses. To use this function, open a window on the input file (`^x-4 ^f`, then enter the name of the file — default location `/aips1/gripe`), position to the top of the file and execute `convert-texgripes` (`ESC-x convert-texgripes`). The routine should run to completion with the gripe and user files added.

(B) The function `convert-AIPS-gripes` reads the *AIPS* binary gripe (`GR`) files and extracts the gripes and corresponding user information and enters them as individual files in the new system. A user information file will only be added if none exists for the user and will need to be edited to put needed linefeeds into the addresses. To use this function, open a window on the input file (`^x-4 ^f`, then enter the name of the file — default location `/aips1/gripe`), position to the top of the file and execute `convert-AIPS-gripes` (`ESC-x convert-AIPS-gripes`).

(C) The function `init-gripe-form` allows interactive input of a gripe. First, it initializes a new buffer containing a “fill-in the blanks” form and automatically assigns a new gripe number and the current date and time to it. To use this function, hit `ESC-x init-gripe-form`. The buffer is then edited and saved with a `^X^S` — this will correct the status if an answer is given and cause the `lq--text` files to be updated. If you wish to stop and return to the Selection/Options form, `^x-b AIPS-GRIBE` will switch buffers. Other buffer-switch options can be displayed by hitting `^x-b ?`.

(D) The function `init-user-info` allows input/editing of user information files. It will first prompt you for a user name and, if there is no information for that user, a new buffer is initialized for interactively entering the data. If an information file for the named user already exists, this file is displayed in a window for possible editing. A user information file may have multiple aliases by using several `_USER:` name pairs. An important point to remember when entering an address is that each line after the first should begin with a blank space. This will assure correct pagination when printing the mailing labels. The user information files are then saved with a `^X^S` command, which will also update the `lq--text` files. Again, `^x-b SPC` will show you all possible buffers to which you can switch.

11 Managing lq-text files

When gripe or user information files are updated they are re-indexed automatically in the `lq--text` system. In addition, there is the function `rebuild-lqtext` to re-build the `lq--text` system. This should be run occasionally to insure that the files are up to date. `rebuild-lqtext` will prompt for the file type to be rebuilt; answer `gripe` for gripes and `adr` for user information files.

12 Editing existing gripes/user information files

If a large number of gripes need editing, such as when assigning keywords to new gripes, use Display option **edit** in the main menu of **viewgripe**. Each of the selected gripes will be displayed in gripe number order in a window for editing. The files can be saved and indexed using the **^X^S** command. The command **ESC ^C** will then advance to the next gripe. Selecting by status **new** and a beginning gripe number is usually appropriate.

Gripes or user information files, displayed in any of the ways described in the user documentation, can be switched to edit mode by typing a lower case **e** when the cursor is in the window. This file can then be saved and indexed using **^X^S** and the edit terminated with **ESC ^C**. The status of any gripe being given an answer will automatically be updated.

12.1 A few words about gripe selection

In order to accomodate this new gripe system, slight modifications have been made to “pre-database” gripes. One important modification is the status of the gripes. Answered gripes prior to about #4563 (those entered before about the second week of November, 1990) will have a status of **answered**, as opposed to **answer**, **answer-new**, or **answer-sent**. You must keep this in mind, particularly when selecting a group of gripes which may have had some entered before *and* after the transition to the current database system. Thus, it may be necessary to run your query twice — once for each “epoch” of entries.

If, at any time, you wish to select only one gripe, you may do so by first entering the number of that gripe at the **Beginning Gripe Number** prompt of the Selection/Options form and then entering **one** at the **Display Options** prompt. This will quickly display the specified gripe.

12.2 A word about printing gripes

If, at any time, you wish to obtain a hardcopy of a gripe (or gripes), selecting the **print** option from the Selection/Options form will invoke a prompt as to whether to send the output to a printer or to write it to a file. If the **file** option is selected, the output is written in file **/tmp/gripestoprint**. Any answer other than **file** will be taken as **printer**. You will note that **one** and **print** cannot be selected simultaneously. Therefore, if you wish to print only one gripe, your other selection criteria must be sufficiently unique such that only one will actually be selected by it. We recommend using the **Arbitrary String in Gripe** where possible.

13 Answering gripes

To insert an answer to a gripe, first select the appropriate gripe(s) in an index display and then display the gripe to be answered. Typing a lower case **e** will switch to **edit** mode. Enter your name as the person responsible for the answer, the current date, and the answer in the appropriate places in the form. A **^x^s** will then automatically correct the status, save the file, and re-index it in the **lq--text** system. At this point it would be friendly to mail the griper the answered gripe. There are two ways to do this. From the edit mode, you can immediately send the gripe to the user (or to any e-mail address) by hitting **ESC-x mail-user-answer** (you will be always be prompted for a valid address and also a return address in case the mail message bounces). The subject heading for the mail message will read **Response to AIP Gripe**. If, within the editor, you need to verify an address, you may use the lookup utility by hitting **ESC-x lookup-user**.

You can also mail a gripe from outside the editor by first returning to the gripe index with **ESC^c**. If you don't know the e-mail address you can look it up by typing a lower case **a** at this point. Then re-select the gripe by hitting **RETURN**. At this stage, a lower case **m** will cause the gripe to be mailed, prompting you for an

address. (Note: this assumes that you did not enter an e-mail address on the top level form. If you did, then the gripe file will be mailed to the address entered in the form and you will not be asked for the address.)

14 Sending answered gripes

If you would like to prepare the selected gripes for mailing, you should enter **send** at the Display Options prompt of the Selection/Options form. The **Status** in the Selection Criteria section of the Selection/Options form should also be filled out as **answer-new** (*i.e.*, gripe answered but not yet sent). The procedure called by this entry will make two files with the selection criteria you specified: one is a file containing the gripes which have been filtered for printing and mailing (`/tmp/gripestomail`), and the other is a file containing the corresponding griper's addresses in a form suitable for printing labels on the line printer (`/tmp/gripeaddresses`). This procedure will also change the status of these gripes from **answer-new** to **answer-sent**. Once the **send** option has been run from the Selection/Options form, it cannot be run from there again, whether or not you restart **viewgripe**, since the contents of the gripe files will have been changed.

If something goes wrong with the above procedure and you need to make the gripe print and address files over again, exit **viewgripe** and restart it. When the Selection/Options form appears, issue the command **sendgripes** (invoked with **ESC-x sendgripes**). This procedure will recognize that its input list (**gripe-by-user**) is empty and will recover the list it last used from file **savegripelist** which lives in the **gripe** directory.

15 Buffers used by the gripe system

There are a number of buffers used in the **viewgripe** system whose names don't usually appear in the mode line. It is frequently desirable to switch to one of these buffers when something goes wrong. (The **switch-to-buffer** command is invoked by `~X-b` with name completion supported). These buffers are:

AIPS-GRYPE — buffer contains the main Selections/Options form. Hitting **RETURN** will invoke whatever function is selected on the form.

GRYPE-INDEX — buffer contains the index of selected gripes and is usually created even if it is not displayed. The usual functions can be invoked when visiting this buffer.

LQTEXT-ERROR — buffer will contain any messages, usually error and warning messages, resulting from running **lq--text** in a subprocess. If something goes wrong at this stage the usual symptom is the claim of **No gripes selected**. A more accurate report may be found in this buffer.

16 Files used by the gripe system

The *AIPS* gripe manipulation system is based on the **lq-text** system to select files and lisp programs running in **emacs** to manipulate the gripes. In addition to the gripes, a database of user address information is also maintained. There are several directories in this system:

`~gripe/bin` contains shell scripts to interact with **lq--text**.

`~gripe/lisp` contains the lisp source code.

`~gripe/gripe` contains the current gripe files, one gripe per file with names of the form **gripegripe_no.grp**.

`~gripe/gripedir` contains the files that **lq--text** needs for the gripes.

`~gripe/adr` contains the user information in files with names of the form `user_name.info`. Aliases are allowed inside the file and `lq--text` is used to actually find the correct file.

`~gripe/adrdir` contains the files that `lq--text` needs for the user information.

There are three kinds of program files needed for this system: `emacs` lisp programs (names ending in `.el`), shell scripts, and a few miscellaneous ones (named below with the full path). They are:

`gripeinit.el` is the file that is run when the `viewgripe` script is executed. It loads the necessary files and brings up the selection menu. Sets up to auto load `sendgripes` and `init-gripe-form`.

`gripeutil.el` contains a number of utility routines. The main command of interest to programmers and the gripes manager is `rebuild-lqtext` which re-builds a set of `lq--text` files.

`mfe.el` contains the form fill-out package.

`newgripes.el` contains programs to enter new gripes into the system. They are `init-gripe-form` to initialize a buffer for a new gripe; `init-user-info` to initialize a user info file; `convert-texgripes` to convert old `TEX` gripes; and `convert-AIPS-gripes` to convert *AIPS* binary gripe files.

`viewgripe.el` contains programs to access and manipulate the gripes, both for user read-only access and programmer/gripe manager access. Function `view-gripe` is the top level routine.

`sendgripes.el` contains programs to prepare files for the distribution of answered gripes and to update the gripe file to indicate this. Function `sendgripes` is the top level routine.

`aipsconsult.el` This program runs the "on-line" consultant routine for those programmers who need "psychiatric" help after using *AIPS* or the Gripes Database.

`viewgripe` script to set environment variables for the gripe and user information files and to start up `emacs` loading the appropriate files.

`lqmerge` script to select a list of gripe files which contain a list of given phrases. Output is to a file named in the call sequence.

`adrfind` script to find the user information file containing a specified string.

`upgripe` script to re-index a gripe file in the LQ text system.

`upadr` script to re-index a user information file in the LQ text system.

`rebuild` script to perform tasks of both the `upgripe` and `upadr` scripts in a batch-like mode. It can be set to run automatically at regular intervals to keep the gripe and user information files current.

`~gripe/gripe/nextnumber` contains the next gripe number to be assigned. It is maintained by function `gripe-number` (in `gripeutil.el`).

`~gripe/gripe/savegripelist` contains the last `gripe-by-user` list sent to `sendgripe`. This is kept for error recovery purposes.

`~gripe/Remote_user` is the shell remote users (user `gripe`) get when they login. It prompts for a terminal type (with a default) traps some errors and starts up `viewgripe`.

17 Database support/assistance

This concludes the documentation for programers' use of the *AIPS* Gripes Database. If there are any problems with the system or in your understanding of how to use it, please contact the database manager (Gustaaf van Moorsel) at the National Radio Astronomy Observatory at (505)835 7396 or send an e-mail message to gvanmoor@nrao.edu. We also welcome any suggestions for improving the performance or configuration of this database. If there are any features which you would like to see added to this system, please let us know.

18 Harvesting gripes²

18.1 Gather.GetMany

Gather.GetMany is a shell script to gather binary gripe *AIPS* GR files from a collection of machines available over the network. Because the shell script relies on the remote machine having the “remote shell” (**rsh**) command, it will only work on Unix machines.

A simple file, **Gather.Hosts**, controls where **Gather.GetMany** will look for gripes. Each line of the file contains three fields separated by spaces. First the name of the machine, next the name of an account with privilege to read/write the GR files, and third the path to the GR file. Additionally, any line beginning with a # sign in the first column is a comment and is otherwise ignored. A sample **Gather.Hosts** file is:

```
# Comments have to start in the first column with a # sign
# Format is:
#Site          via HOST          account    path to DA00
#
CVILLE-SUNS   baboon.cv.nrao.edu   aipgripe  /AIPS/DA00/BABOON
NRAO1         baboon.cv.nrao.edu   aipgripe  /nrao1/aips/code/DA00/NRAO1
AOC-SUNS      aguila.aoc.nrao.edu  aipsmgr   /AIPS/DA00/AGUILA
AOC-CONVEXES aguila.aoc.nrao.edu  aipsmgr   /yucca/AIPS/DA00/YUCCA
AOC-IBMS      zuni.aoc.nrao.edu    aipsmgr   /AIPS/DA00/ZUNI
```

Each machine listed in **Gather.Hosts** needs to permit the person running **Gather.GetMany** to use **rsh** as the appropriate user (see **man rhosts**). For instance, suppose **Gather.GetMany** is being run as user **aipgripe** on machine **baboon**. Then the file **.rhosts** in the home directories of **aipgripe** on **baboon**, and of **aipsmgr** on **aguila** should contain the lines

```
baboon.cv.nrao.edu aipgripe
baboon aipgripe
baboon.cv.nrao.edu aipsmgr
baboon aipsmgr
```

There must be a “dummy” GR file that contains no gripes. This is used to reset the gripes file to none by copying over the original GR file. If the GR file is named **GRC00000;1**, then the dummy GR file should be named **GRCdummy;1**. (If the format “version” of the files changes in a future release of *AIPS*, the C will become D, then E etc. **Gather.GetMany** is smart enough to do the right thing until we get past Z and this will probably never happen.)

The script **Gather.GetMany** uses the script **Gather.GetOne** to get gripes from one site. When run, **Gather.GetOne** will concatenate all the remote GR files it can get in the file **GRCremote** (or **GRDremote** etc). Because **Gather.GetOne** always appends, you must remove the **GRCremote** file once the gripes in it are entered into the gripes system. As it runs, **Gather.GetOne** will write progress messages on the terminal. It will also put the messages into a log file called **Gather.Log**.

Conditions that **Gather.GetMany** tries to notice and report include:

```
Machine or network down
No gripes entered on this host
* No GR files on host
* No ‘dummy’ file on host
Copy fails
* Concatenate fails (out of local disk?)
```

The items marked with * indicate something that needs to be looked into; everything else is normal if it doesn't persist.

²This section was written by Brian Glendenning 1990 Dec 12 and revised by Glen Langston on 1992 May 04.

18.2 Contents of a raw gripe

The **GRCremote** file contains many invalid characters as well as valid gripes. Each Gripe is composed of 10 pieces, and each piece is contained between a pair of curly brackets ("{" and "}"). It is critical for the Gripe harvesting that there are *exactly 10 pairs of curly brackets, and that the correct information is entered in its respective set of brackets*. What should appear in the ten (10) required sets of brackets is:

1. {date and time}
2. {computer}
3. {user# and version of AIPS}
4. {username}
5. {user address}
6. {user phone}
7. {gripe}
8. {one-line description of gripe}
9. {user e-mail address}
10. {I have NO IDEA what goes here}

18.3 Adding GRCremote to the database

After harvesting the gripes, several steps remain before the gripe is sitting happily in the gripes database.

1. Edit the **GRCremote** file to remove the extraneous characters (mostly `^Qs` and `^Ms`).
2. Then start up the gripes database as user **aipgripe** using **viewgripe** as described above.
3. Bring the new **GRCremote** file in by typing the control sequence `^X4^F` (which will ask for the name of a file to edit) and then the full name of the file. It will appear in one of the **emacs** windows.
4. Move into the file and type the control sequence `ESC-x convert-AIPS-gripes`. **emacs** will say something about editing files in the background.
5. Leave the **GRCremote** file by typing the control sequence `^X0`.
6. Look at the list of new gripes with the gripes data-base command **index**.

³the answer to the gripe, dummy