GNATS Frequently Asked Questions

Hans-Albert Schneider

Table of Contents

GNATS Frequently Asked Questions

This document answers some frequently asked questions concerning GNATS, the GNU problem report management system, and related software. The most recent version of this FAQ can be found at `http://www.gnu.org/software/gnats/doc/faq/gnats-faq.html`.

The questions and answers have been compiled by Hans-Albert Schneider, mainly using the help-gnats mailing list and the GNATS documentation as input. Please report any errors and suggestions to him.

1 General Questions

1.1 What is GNATS?

GNATS is the GNU problem report management system. Problem report management systems are also known as "bug-tracking systems", though the entries need not be bugs (e.g., think of change requests). The acronym stands for "GNats: A Tracking System".

GNATS stores all information about the problem reports at a central site, and enables users to access this site by various means, including e-mail, WWW, and a network daemon. New problem reports can be created, and existing reports can be queried and updated, by most of these means.

GNATS is widely customizable: Of course you can define report categories (is the report about tool A or service B?), responsibles (who takes care of this report?), and submitters (is it from customer1 or from the sales department?). You can also define possible states of a report (open, analyzed, closed, etc.) and classes (software bug, documentation bug, change request, . . .).

Starting with GNATS 4.0, you can define your own custom fields, and customize many of the built-in fields; you can have fields automatically set to a certain value when another field changes its contents (e.g., set a "Closed-Date" field to the current date when the report goes to state "closed", and unset it when it goes back from "closed" to something else), or when the report is changed at all (e.g., to maintain a "Last-Modified" field).

1.1.1 How it Works—User's View

Users enter their problem reports (PRs) via some front-end. This front-end either directly contacts the GNATS server, or sends the report via e-mail. (The server is available since about version 3.90 of GNATS). For an incomplete list of front-ends see Chapter 4 [Clients], page 13.

Some front-ends (like TkGnats, see Section 4.2 [TkGnats], page 14) are capable of contacting several GNATS servers, or several problem report databases managed by the same GNATS server.

1.1.2 How it Works—Administrator's View

The server is started by a super-server like inetd or xinetd. It has a built-in access control mechanism based on IP addresses and username/password.

For users sending their PRs via e-mail, some mail addresses must be configured.

1.2 How is "GNATS" pronounced?

Q: So, How can I pronounce "GNATS?" Should I pronounce "G"?

It has been reported that the original developers pronounced the "G". (This is unlike the little insect called "gnat" where the "G" is not pronounced.)

1.3 Where do I get it?

See the GNATS home page at `http://www.gnu.org/software/gnats/`.

The current development version is available via CVS only, see the GNU savannah GNATS project page for the instructions how to get it.

1.4 What is the latest version?

The version numbers of GNATS 3.x is somewhat confusing, because 3.1xx is newer than 3.2.

Version 3.2 was released in 1993 (I never tried it, but it is said to be "really broken"). A lot of new features have been added since 3.2, including multiple database support and customized fields. Among the contributed software are comprehensive WWW and Tk based front-ends.

The latest 3.x release is 3.113.1. It is deprecated due to security concerns and lack of maintenance.

GNATS 4.0 was released in August 2003, and 4.1 was released in March 2005. All released versions are available for download at `http://ftp.gnu.org/pub/gnu/gnats/`.

1.5 Is there a Mailing List?

Currently, the following GNATS mailing lists exist:

- info-gnats@gnu.org This is a low volume moderated mailing lists for GNATS related announcements, like new releases or important information.
- help-gnats@gnu.org A low-to-middle volume mailing list where anything related to GNATS can be discussed (development, help, questions, suggestions, etc.).
- bug-gnats@gnu.org Bug reports can be sent there. Please note that it's preferred to submit bug reports via the bug tracking system (see Section 1.6 [Bug Reporting], page 3), which automatically mirrors them to this list.

Lists that are of interest mainly for the GNATS developers:

- gnats-prs@gnu.org Copies of the bug tracking system messages are sent there. Contrast to bug-gnats@gnu.org which only gets new bug reports, this list also gets the follow-ups (Whatever goes into GNATS' `>Audit-Trail:` field).
- gnats-commit@gnu.org CVS commit messages are sent there.
- gnats-diffs@gnu.org Diffs of committed files are sent automatically to this address.

1.6 How do I Report a Bug?

Before reporting a bug, make sure it really is a bug, not a simple misunderstanding or a misconfiguration. Please check the manual and this FAQ. You may also ask for help on the help-gnats@gnu.org mailing list (you may want to search the list archive first; it is available from `http://lists.gnu.org/mailman/listinfo/help-gnats`).

If it is a bug, please report it via the bug tracking system. It resides at `http://bugs.gnu.org/cgi-bin/gnatsweb.pl?database=gnats`. This is a common bug database for GNATS, Gnatsweb and TkGnats. (And, yes, of course: It uses GNATS and Gnatsweb.)

When you report problems concerning GNATS itself, please do not forget to provide especially the following information:

- The GNATS version you are using. If your problem involves Gnatsweb or TkGnats, please also report their versions.
- The *exact* way how to reproduce the bug.
- Your configuration.

- If you encounter a compilation or build problem, it is especially important to mention the operating system, compiler and possibly other build utilities you use.

Providing this information in the initial report avoids further unnecessary communication, saves the limited resources of the developers (keep in mind that they are working on GNATS and friends in their spare time) and helps to track down and fix the problem soon.

1.7 Upgrading Considerations

See Section "Upgrading from older versions" in *Keeping Track* for the upgrade process from 3.x to 4.x.

If you are running a 3.x version, *please consider upgrading to 4.0.* There are some security concerns about the 3.1xx code that have gone with 4.0. Furthermore, the GNATS 3 branch is not maintained anymore due to lack of capacity.

The default format of the reports has not changed during the transition from 3.113.1 to 4.0, though you may add more fields to 4.0 reports (and leave out others). The format of the index file is now binary.

However, you have to change the e-mail aliases because the meaning of the '`-d`' option to the client programs has changed: It took the *directory* of the GNATS database in 3.1xx, and now takes its *name* (plus, it is now installed into the `libexec` directory instead of the `lib` directory). I.e., you must change the aliases:

```
# GNATS 3 aliases:
ourdb-query: "|/usr/local/lib/gnats/mail-query -d /usr/local/gnats/db2"
ourdb-bugs:  "|/usr/local/lib/gnats/queue-pr -d /usr/local/gnats/db2 -q"
```

to

```
# GNATS 4 aliases:
ourdb-query: "|/usr/local/libexec/gnats/mail-query -d ourdb"
ourdb-bugs:  "|/usr/local/libexec/gnats/queue-pr -d ourdb -q"
```

Also note that the client/server protocol has changed from 3.1xx to 4.0; you cannot run clients from one version with the server from the other version.

2 Problems during Installation

As installation and configuration problems often overlap, please check also Chapter 3 [Configuration], page 6.

2.1 Gnatsweb and GNATS do not like each other.

Q: I have installed GNATS 3.999.x and Gnatsweb 2.9.x, and I cannot get them work together.

Symptoms are that "make test" fails; if Gnatsweb is installed nevertheless, it hangs when trying to login into the GNATS server.

A: The versions are incompatible. Gnatsweb 2.9.x is for use with GNATS 3.113.1. With GNATS 3.999.x, you must use Gnatsweb 3.99.x; it is in the contrib subdirectory of the GNATS distribution. (GNATS 3.999.x/Gnatsweb 3.99.x were pre-releases of the 4.0 releases. As 4.0 is out now, you may consider to upgrade to the non-pre versions.)

For GNATS 4.x, use Gnatsweb 4.0 or greater.

3 Configuration Issues

3.1 General Configuration Questions

3.1.1 How do I add a new database?

A: (GNATS 3.1xx) (Please consider upgrading to GNATS 4.0 or greater.)

1. Create the base directory for the new database, say `/usr/local/gnats/db2`, and make sure it is owned by the GNATS user.
2. In there, create subdirectories `gnats-adm`, `gnats-adm/locks`, and `gnats-queue`, and make sure they is owned by the GNATS user.
3. Now you need to populate the new `gnats-adm` directory. The easiest way is to copy all files from `gnats-db/gnats-adm` to `/usr/local/gnats/db2/gnats-adm`, and edit them to reflect the needs of your new database. Again, make sure they are owned by the gnats user.

   Note that there *must* be a category named `pending`. It is used when no category is given in a report, and when a report names an invalid category.

   Also note that each database needs its own mail address for submissions (see also step 8 below), and that you must enter it in the file `config`.
4. As the GNATS user, run '`mkcat --directory=/usr/local/gnats/db2`' to create the directories for the categories. Remove the files `current` and `index`, if they exist.
5. If the new database is split off of another one, move or copy the existing reports to their new location. This is easiest if the categories have the same names as in the original database. (See Section 3.1.2 [Renaming a Category], page 8 if some of them changed names.)

   Run the program `gen-index` to create the index file (see Section "Regenerating the index" in *Keeping Track*).

   Find the greatest report number and put it (or any larger number) into `/usr/local/gnats/db2/gnats-adm/current`.

   Caution: E-mail updates to the PRs you moved to the new database may still arrive at the old database. You may want to contact everybody who knows about these PRs, asking them to use the mail address of the new database when sending a follow-up.
6. Now add a line for the new database to `gnats-db.conf`, like this:

   ```
   /usr/local/gnats/db2:GreatNewDB
   ```

   Gnatsd reads it on startup (and as it is started by inetd, this means it becomes effective with the next connection to gnatsd). Gnatsweb (see Section 4.1 [Gnatsweb], page 13) learns the database list from gnatsd, so it will offer you the new database "GreatNewDB" when it is invoked next time.

   If you do not know where `gnats-db.conf` lives, run:

   '`strings /where/ever/gnatsd | grep gnats-db.conf`'
7. Don't forget to setup a cron job to run through the additional `gnats-queue`. Note that you need to specify the database directory to `queue-pr`, i.e.,

   '`/usr/local/lib/gnats/queue-pr -d /usr/local/gnats/db2 -r`' or, if you prefer the long options,

'`/usr/local/lib/gnats/queue-pr --directory=/usr/local/gnats/db2 --run`'

8. Don't forget to create mail aliases for your new database (see Section "Setting up mail aliases" in *Keeping Track*). Take care that the aliases have the right database, e.g.,

   ```
   GreatNewDB-bugs:  "|/usr/local/lib/gnats/queue-pr -d /usr/local/gnats/db2 -q"
   GreatNewDB-query: "|/usr/local/lib/gnats/mail-query -d /usr/local/gnats/db2"
   ```

   If you do not want to allow querying the database by mail, omit the '`GreatNewDB-query`' alias.

   You usually need the cooperation of a system administrator for this step (if you are not a system administrator yourself, of course).

   Make sure that `/usr/local/gnats/db2/gnats-adm/config` gives the correct mail addresses for *GNATS_ADDR* (this *must* be different for each database) and for *GNATS_ADMIN* (this is probably the same for all databases).

   ```
   GNATS_ADDR="GreatNewDB-bugs@bugs.example.com"
   GNATS_ADMIN="gnats-admin@bugs.example.com"
   ```

   If your GNATS sits behind a firewall and needs to exchange mails with the outside world, see also [Outgoing mail bounces], page 11.

A: (GNATS 4.x) With version 4, this has become much easier (see Section "Adding another database" in *Keeping Track*):

1. Add a line for the new database to `databases`, like this:

   `GreatNewDB:Our great tools:/usr/local/gnats/db2`

   Then, as the GNATS user, run '`mkdb GreatNewDB`' to create the database. Make sure that the directory (in our example, `/usr/local/gnats/db2`) can be created by the GNATS user.

   (Note that there *must* be a database named `default`. It is used as a fallback by some tools if no database is specified. You need not use it actively, but you should have run '`mkdb default`'.)

   Gnatsd reads the file `databases` on startup (and as it is started by inetd, this means it becomes effective with the next connection to gnatsd). Gnatsweb (see Section 4.1 [Gnatsweb], page 13) learns the database list from gnatsd, so it will offer you the new database "GreatNewDB" when it is invoked next time.

   If you do not know where `databases` lives, run:

   '`strings /where/ever/gnatsd | grep databases`'

2. Now, as the GNATS user, edit the files in directory `/usr/local/gnats/db2/gnats-adm` to reflect the needs of your new database.

3. Depending on the settings in the `dbconfig` file of the new database, you need not run `mkcat` anymore in order to create new category directories in your database—GNATS 4 creates them automatically when they are missing. See Section "The `dbconfig` file" in *Keeping Track*, for details.

4. Don't forget to setup a cron job to run through the additional `gnats-queue`. Note that you need to specify the database to `queue-pr`, i.e.,

   '`/usr/local/libexec/gnats/queue-pr -d GreatNewDB -r`'

   or, if you prefer the long options,

‘/usr/local/libexec/gnats/queue-pr --database=GreatNewDB --run’

5. Don’t forget to create mail aliases for your new database (see Section “Setting up mail aliases” in *Keeping Track*). Take care that the aliases have the right database, e.g.,

   ```
   GreatNewDB-bugs:  "|/usr/local/libexec/gnats/queue-pr -d GreatNewDB -q"
   GreatNewDB-query: "|/usr/local/libexec/gnats/mail-query -d GreatNewDB"
   ```

   If you are updating from GNATS 3.1xx, note that the ‘-d’ option has changed its meaning: it does not give the directory of the database, but its name. (In case you prefer the long form of the option, it is now ‘--database’ instead of ‘--directory’.)

   If your GNATS sits behind a firewall and needs to exchange mails with the outside world, see also [Outgoing mail bounces], page 11.

3.1.2 How do I rename a category?

Renaming a category requires to touch every PR in that category, because each report contains the name of its category.

To rename category A to B, proceed as follows:

1. Create a new category B.
2. Edit every report in category A, changing its category to B. This can be done with any GNATS client; check the archives of the HELP-GNATS mailing list for hints about automating this step.
3. Run gen-index (see Section “Regenerating the index” in *Keeping Track*) to refresh the index file.
4. If you are using GNATS 4 (or planning to upgrade soon), you should not remove the category A. When a follow-up to an existing PR arrives via e-mail, GNATS 4.x checks that both the category and the PR number indicated in the mail exist (this is a sanity check).

   To reduce the risk of new reports being filed to category A, change its description in the categories file to something like ‘obsolete, use category B instead’.

3.1.3 How do I add, remove, or rename a PR field?

A: (GNATS 3.1xx) The fields and their names are fixed in GNATS 3.1xx, so this is not possible.

A: (GNATS 4.x) Edit the file dbconfig to reflect your changes.

Note that the PR fields with the builtin-names severity, priority and state are required if you want automatic reminders (notify-about-expired-prs = true). In this case, the file submitters must also contain a response time. The severity field is checked for values critical and serious, and priority for value high. This is currently not configurable.

3.2 Gnatsd, the GNATS Daemon

3.2.1 Gnatsd Port Number

In 3.xxx versions, gnatsd uses port 1529 by default; as this port is officially assigned to another application (see the list of port numbers maintained by the Internet Assigned Numbers Authority) it will probably change in 4.x.

If you want/need to run gnatsd on another port, you can change the default port at compile time (call `configure --help` to learn about compile time configuration options). Most clients also accept an option or configuration variable to change the port. Don't forget to tell inetd (or xinetd, or whatever super-server you use) to start gnatsd on the other port.

3.2.2 How to Start Gnatsd

Gnatsd is intended to be started by some "super server", like `inetd` or `xinetd`. ("Super servers" are sometimes also called "super daemons".)

This is also described in the GNATS manual; see Section "Installing the daemon" in *Keeping Track*.

How to configure inetd to start gnatsd

If your gnatsd will be started by inetd (the "internet daemon"), add the following entry to your `/etc/inetd.conf`:

```
# port                       userid program
support stream tcp nowait gnats /usr/local/libexec/gnats/gnatsd gnatsd
```

and to `/etc/services`:

```
support 1529/tcp # GNATS
```

(You may need to use tabulator characters to separate the fields of `/etc/services`.) Then send inetd a hangup signal (`kill -HUP` *`pid-of-inetd`*).

You may want to use another port instead of 1529 (see Section 3.2.1 [Port Number], page 8).

How to configure xinetd to start gnatsd

If your gnatsd will be started by xinetd (the "extended internet daemon"), create a file `/etc/xinetd.d/support` with the following lines (see Section "Installing the daemon" in *Keeping Track*):

```
service support
{
    disable     = no
    socket_type = stream
    protocol    = tcp
    wait        = no
    user        = gnats
    server      = /usr/local/libexec/gnats/gnatsd
        server_args = gnatsd
}
```

or add theses lines to your `xinetd.conf` file, whatever is appropriate. The equal signs seem to need spaces around them with some versions of xinetd.

You need to add

```
support 1529/tcp # GNATS
```

to `/etc/services` (it may be necessary to use tabulator characters to separate the fields), and to tell xinetd to reread its configuration (`kill -HUP` *`pid-of-xinetd`*).

3.2.3 Gnatsd Messages

You are not on the host access list: stdin (stdin)

Q: When starting gnatsd manually (on the command line), I get '`520 You are not on the host access list: stdin (stdin)`'.

A: (GNATS 3.1xx) Gnatsd is not intended to be started manually, but via inetd, xinetd, or a similar "super server". When gnatsd is started this way, its stdin and stdout (standard input and standard output streams) are connected to a so called "TCP socket" (one end of the network connection), and from this socket gnatsd learns the IP address of the remote end (and from this it derives the remote host name). When you start gnatsd from the command line, its stdin and stdout are connected to the terminal, and thus gnatsd gets a nonsense value.

You usually need '`root`' privileges in order to change the configuration of your "super server". If you don't have them, contact your system administrator.

See Section 3.2.2 [Starting gnatsd], page 9 for inetd and xinetd configuration.

A: (GNATS 4.0) This error message should not occur with gnatsd 4.0 anymore, because gnatsd can be started from the command line. However, this probably only makes sense for debugging and diagnosing problems. Furthermore, you need to play the part of the client program yourself (i.e., you must "speak" the gnatsd client protocol).

You are not on the host access list

Q: Gnatsd rejects connections from a remote host with the error message '`520 You are not on the host access list`'.

A: Check the file `gnatsd.host_access`; if you are using the default locations, this is `/usr/local/etc/gnats/gnatsd.host_access`. (If you are still using GNATS 3.xxx, the file is named `gnatsd.conf`; its default place is `/usr/local/etc/gnatsd.conf`.)

Each line of the file names a host and its access level, separated by colons. Gnatsd tries the lines in turn to match the remote host, and the first line that matches wins.

The first field specifies the host(s); it may be a host name (like '`goedel.example.com`'), a partial domain (like '`*.example.com`'), an IP address (like '`192.168.1.5`'), or a partial IP address (like '`192.168.*`'). If it is only '`*`', it matches all hosts.

Depending on how IP addresses are mapped to hostnames on your gnatsd machine, you may be able to omit the domain (like in '`goedel`'). You may even *need* to omit the domain for some hosts.

The second field is the access level granted to the remote host. This is usually increased by the access level granted to the user as soon as (s)he logs in.

The third field is currently not used; just leave it empty (but supply the colon between the second and the third field).

Example:

```
#
# This is a comment
#
# Grant view access to all hosts with IP addresses
# ranging from 192.168.0.0 to 192.168.255.255:
192.168.*:view:
# Users on host goedel.example.com get (at least) edit access:
goedel.example.com:edit:
# Users from escher.example.com may view all reports, even
# confidential ones:
escher.example.com:viewconf:
# Users from bach.example.com may only view all non-confidential
# reports:
bach.example.com:view:
# Users from other example.com hosts
# only get the access specified for them in gnatsd.user_access:
*.example.com:none:
# Same for domain our-users.example:
*.our-users.example:none:
# All other hosts are rejected
# without even asking for username and password:
*:deny:
```

The format is described in detail in the comments at the beginning of the file.

3.3 E-Mail Issues

queue-pr not available for sendmail programs

Your sendmail installation uses '`smrsh`' to check programs that are invoked from a mail alias. '`smrsh`' only allows the execution of programs of which it is told that they are save. To tell it that `queue-pr` is save, create a symbolic link in the directory `/etc/smrsh` to `queue-pr`, like this:

```
mkdir /etc/smrsh # if it does not yet exist
cd /etc/smrsh
ln -s /usr/local/libexec/gnats/queue-pr queue-pr
```

Replace `/usr/local/libexec/gnats/queue-pr` by the real path to `queue-pr`.

Mail from GNATS bounces

You are getting bounce mails like this:

```
   ----- Transcript of session follows -----
... while talking to mail.example.com.:
>>> MAIL From:<gnats@mymachine.subdomain.example.com> SIZE=334
<<< 501 5.1.8 <gnats@mymachine.subdomain.example.com>...
      Domain of sender address gnats@mymachine.subdomain.example.com does not exist
```

Probably your GNATS sits behind a firewall, but it needs to exchange mails with the world outside the firewall. Make sure that *GNATS_ADDR* is valid outside.

This is what probably happens behind the scenes: On its way to the outside world, the sender address of the mail gets rewritten to something "official" (like `Hans-Albert.Schneider@example.com`, as opposed to `me@mymachine.subdomain.example.com`). This is especially necessary if the internal hosts are not visible outside. If your outgoing mail gateway still sees the internal address, it complains with a message like the above (it has probably checked `mymachine.subdomain.example.com` with your organization's external DNS server, which does not know `mymachine`).

To solve this problem, get an "official" address for your GNATS system, e.g., toolbugs@example.com, and use that one for *GNATS_ADDR*.

3.4 Miscellaneous Configuration Issues

New reports do not show up in the database

1. Make sure you have set up the mail aliases for your database(s). (See Section 3.1.1 [Add a database], page 6 and Section "Adding another database" in *Keeping Track*.)
2. Try to send a PR using '`send-pr`'. If you get error replies, investigate the causes. See Section 3.3 [E-Mail], page 11.
3. Does a new file show up in the `gnats-queue` directory of your database, with the report as its contents? If it does, make sure you have created the cron job for this database.
4. If the report does not show up in the `gnats-queue` of the desired database, does it show up in the `gnats-queue` directory of another database? In this case, make sure your mail aliases are set up correctly; especially, make sure that you have specified the correct '`--directory`' option ('`--database`' for GNATS 4.x) for each `queue-pr`.

Assigning responsible etc. when entering a report

Q: When entering a new PR, I want to assign a value to some field (e.g., `Responsible`) which is normally assigned by GNATS.

A: (GNATS 3.1xx) This is not possible in GNATS 3.1xx.

A: (GNATS 4.x) This is done by adding the field name (in this example, `Responsible`) to the `initial-entry` item at the very end of the `dbconfig` file. Gnatsweb will pick this up and add a `Responsible` field to the Create PR form.

4 Client Software

Several client applications can be found in the GNATS distribution. The most important are Gnatsweb (a WWW interface to GNATS), TkGnats (a Tcl/Tk based interface), a GNATS mode for Emacs and XEmacs, and send-pr (the traditional command line interface that sends a PR by e-mail).

4.1 Gnatsweb

See also Section 2.1 [Gnatsweb and GNATS], page 5 in the chapter on Chapter 2 [Installation], page 5.

4.1.1 Login Questions

What should be entered as login information?

A: If you get an HTML form asking you to enter username, password, and database, use what your gnats administrator has told you. In this case, your authentication data is checked with gnatsd's own user database.

If your browser gives you a window asking for username and password (or for "credentials"), the authentication is done by the WWW server. You should have got the necessary data either from your gnats administrator, or from the webmaster of the site offering Gnatsweb access.

Login data are not remembered

Q: After logging in via Gnatsweb, the main screen is displayed, but whatever action is selected (whatever button is clicked), the login screen gets displayed again.

A: This is usually a problem of cookies getting ignored. Gnatsweb uses cookies to store the login information (see "Gnatsweb and Cookies" below). Configure your web browser to allow the cookies Gnatsweb tries to set.

Q: There is not even the main screen, the user immediately gets the login screen when trying to log in. The login data entered is correct. It works fine on another machine and for other users.

A: This has actually been reported only once; but as the effect is similar to the previous one, it is included in the FAQ. It apparently was related to very strict security settings, but has not been investigated in detail.

Gnatsweb and Cookies

Q: Wait a moment! Cookies? Can I eat them?

A: No, not these. The cookies about which we are talking here are little data packages that a web-server sends to your browser (in our case, on behalf of Gnatsweb) and your browser sends them back to the server next time. Modern browsers give you the option to generally accept or reject cookies, or to be asked whenever a cookie arrives (some even can make this decision based on the web-server). To learn more about cookies, visit, e.g., the Cookie Central.

Q: So, which cookies does Gnatsweb send, and why?

A: Gnatsweb uses a cookie `gnatsweb-db-`*database_name* to store your login information (username, password), and a cookie `gnatsweb-global` to store the GNATS database you are working on, your e-mail address (to fill it in for you when you create a new report or reply to an existing report), a default list of fields to display in search results, and defaults for `Submitter-Id` and `Originator` fields.

Gnatsweb also sets a cookie for each "saved query" (named `gnatsweb-query-`*queryname*).

It is therefore essential that you allow Gnatsweb to set its cookies. For a typical effect of rejecting them, see [Login Not Remembered], page 13.

4.2 TkGnats

To be completed.

4.3 GNATS Mode for Emacs

The Emacs mode is in the file gnats.el

Does it also work for XEmacs?

A: (GNATS **3.1xx**) Yes.

A: (GNATS **4.x**) Still to be tested.

4.4 send-pr

A simple command line tool that sends a report via electronic mail. This is not contributed software, but part of GNATS itself.

The 3.xxx versions do not support multiple databases.

To be completed.

Appendix A Glossary

cookie A chunk of data that a WWW server stores in your browser. It is sent back to the server when you contact it again. Gnatsweb (see Section 4.1 [Gnatsweb], page 13) uses cookies to store your login information, preferences, and named queries.

gnats.el GNATS mode for (X)Emacs.

gnatsd The GNATS daemon (server program).

Gnatsweb A WWW interface to GNATS.

port A communication endpoint of various IP based protocols, notably TCP and UDP. A server program waits for connections ("listens") on a port, and a client program connects to this port. IANA (the Internet Assigned Numbers Authority) maintains a list of port numbers registered for certain purposes.

PR Short for “Problem Report”. An entry in a GNATS database.

send-pr A command line tool that sends a new report by e-mail.

super server

A super server (sometimes also called "super daemon") is a daemon (= UNIX server process) that waits for connections on the ports of various server programs and starts the corresponding server when a connection is made. This way, there is only one server process for a set of services (instead of one for each service), but each service in the set is available to its clients.

The most common super servers are `inetd` (the "InterNET Daemon" that comes with many UNIX installations) and xinetd (the "eXtended InterNET Daemon").

TkGnats A Tcl/Tk based interface to GNATS.

Index