The BATTS Trouble Ticketing System

Jon Lasser

The BATTS Trouble Ticketing System
by Jon Lasser

$Header: /var/cvs/batts/doc/batts.sgml,v 1.5 2001/11/30 17:15:32 lasser Exp $ Edition

Table of Contents

Chapter 1. About BATTS

You've downloaded BATTS, or you're reading this on the web. You probably have a couple of questions. This is the introductory bit of the manual where we attempt to answer the obvious questions for you.

What Is BATTS?

BATTS is the Barnhard Associates Trouble Ticket System. BATTS is a command-line oriented trouble-ticketing system for Unix systems with the following features:

- User-configurable priorities, categories, and billing codes
- Log entries with start times, end times, billing codes, and text associated with tickets
- A full-featured e-mail interface for customers
- Much, much more ...

BATTS is written in Perl, and uses MySQL for its back end. Use of the Perl DBI theoretically provides for portability to other databases, though no effort has yet been taken to test this theory. BATTS is Open-Source software, available under the terms of the GNU General Public License (GPL), Version 2 or later. See Appendix A for the text of the license.

Where do I go (to get) BATTS?

The official BATTS home page is http://www.xisp.net/batts/[1]. New releases and news should be available there, along with a copy of the documentation.

Who Is BATTS?

BATTS is written and maintained by Jon Lasser <`jon@cluestickconsulting.com`> of Cluestick Consulting[2], on contract for Barnhard Associates[3], and has been made available as a public service. Not to mention the free advertising. Both Cluestick Consulting and Barnhard Associates may be hired to install, maintain, or extend a BATTS installation or the BATTS code.

How Can I Help?

The best way to help BATTS is to send me money. If you don't want to send me money for no benefit to yourself, hire me or Barnhard Associates to work on BATTS. Or make some useful changes to the code: The BATTS distribution includes a file named `TODO` listing high-priority items. Code that solves these problems would be wonderful, as would additional code that meets your needs. The documentation needs help too.

If you don't need to hire consultants, and you aren't a programmer or a writer, you can submit bug reports. If something doesn't work, send mail. If something acts strangely but correctly, that may be worth sending mail, too.

Putting a big heaping pile of code on the Internet for public consumption is a frightening endeavor, and is a big time-sink. Words of kindness and support are always appreciated.

Notes

1. http://www.xisp.net/batts/
2. http://www.cluestickconsulting.com/
3. http://www.barnhard.com/

Chapter 2. Installing BATTS

Installing BATTS consists of five steps: installing BATTS' prerequisites; modifying and installing the configuration files; creating the database; installing the scripts; and installing the mail gateway.

If you are familiar with Unix or Linux systems, BATTS should be relatively easy to install. These instructions were written based on installation of BATTS on a Red Hat 7.1 system, though other Unix systems should be rather similar. BATTS has never been installed on a Windows system. I doubt that it would run out-of-the-box on any Windows system, but if you manage to get it working, please let us know.

Installing Prerequisites

In order to function, BATTS needs several other pieces of software installed. These include Perl, MySQL, Sendmail or equivalent, and several Perl Modules. Installation of Perl and MySQL are not trivial; both ship with many modern Linux distributions and, in these cases, the default installation should suffice.

If you are running a Unix system without Perl installed (egads!), Perl may be obtained from www.perl.com[1]. BATTS was developed on a system running Perl 5.6.0; Any version of Perl greater than 5 should probably work, but nothing else has been tested. If you are successfully running BATTS on a system with a different version of Perl, or if you have found a version of Perl incompatible with BATTS, please send me mail at <`jon@cluestickconsulting.com`> so that future releases of this manual can be kept up-to-date.

MySQL is available from www.mysql.com[2]. Red Hat 7.1 includes version 3.23.36; this is the version of MySQL that BATTS has been developed and tested with. Any version greater than 3.23.0 should work; if you are successfully running BATTS with a different version of MySQL, or you have discovered that BATTS is incompatible with a particular version of MySQL, please send e-mail and let me know.

Virtually every Unix and Linux distribution comes with either genuine Sendmail or a substitute that contains a binary named (or symbolically linked to) `sendmail`. As long as it accepts standard Sendmail options, we don't really care if it's the real thing. [3] If you are using something other than stock Sendmail and needed to make changes in order to make it work, please send me mail. (Are you noticing a pattern here?)

Installing Perl Modules

In order to work, BATTS needs the Perl DBI (Database Interface) module and the DBD (Database Driver) module for MySQL installed. In addition, the Mail::Header and Mail::Internet modules from the MailTools package are required.

The DBI home page is located at http://dbi.symbolstone.org/[4]. Complete instructions for installing DBI and DBD may be found there. Red Hat Linux and some other Linux distributions include packages of the DBI and DBD modules. Red Hat's packages are named `perl-DBI` and `perl-DBD-MySQL`; they are available as part of the base Red Hat distribution for versions 7.0 and later.

If these modules are not already installed on your system, they can be installed via CPAN, the Comprehensive Perl Archive Network. You can enter the CPAN shell by running the command **`perl -MCPAN -e shell \;`** as root. You will be asked a number of questions regarding the installation of CPAN. Answer these as best you can.

Eventually you will arrive at the CPAN shell prompt, which is simply a line that begins with the word `cpan>` followed by the cursor. DBI and DBD are available as *bundles*, which are essentially a group of separate modules. To install the DBD bundle, type **`install Bundle::DBD`** at the CPAN prompt and hit **Enter**. The DBI may be installed in the same fashion; not surprisingly, it is named `Bundle::DBI`.

After installing these modules, you can install the `Mail::Header` and `Mail::Internet` modules in the same fashion. There does not appear to be a MailTools bundle, though it should not be difficult to install the entire MailTools package via CPAN.

If all of these commands worked, you have installed all of the prerequisites for BATTS. Congratulations!

The BATTS Security Model

BATTS doesn't have any security. Well, almost.

BATTS will create for itself a MySQL user named `batts` who has full control over the ticketing system. Anyone who can run the BATTS binary and who can read the `batts.myrc` file discussed below will be able to do anything to the BATTS ticketing system, including the ability to delete records or even the entire database.

BATTS assumes that you will use the Unix permissions to control access to the `batts.myrc` file, which contains the database connection information. Our advice is to create a group named `batts` in your `/etc/group` file and to make all authorized users of the ticketing system members of that group. Other people will be able to create tickets and even add information to the ticket log via the e-mail interface.

Once you have done this, all you need to do is to change the group of the `batts.myrc` to `batts` and change permissions on the file to `0640`. After this, only members of the group `batts` and the owner of the file, who is probably root, will have the password to the database.

One last note: Unix group information is obtained at login; once you have created the group, log out and log back in to become a member of that group. Until you log out and back in, you will not be a member of the `batts` group.

Modifying and Installing Configuration Files

BATTS has three configuration files that need to be installed in `/etc`: `batts.myrc`, `battsrc`, and `batts-reply.txt`. Sample files are included in the `config_files` subdirectory of the BATTS distribution; each of these files must be modified before you can use BATTS.

The `batts.myrc` file contains the database connection information. It includes the host name, database user, and password for the BATTS database. You will need to change the `host=` and `pass=` lines of the configuration file, at a minimum. We recommend sticking with `batts` as the user name, but you can change this as well if you desire.

The `battsrc` file contains configuration information required by the BATTS program for sending mail. This file is read directly by Perl, and is in fact live Perl code. If you don't have any complex configuration, this file is quite simple. These are variable assignments, which are of the general form

```
$variable = 'value';
```

All Perl variable names (the ones we care about, anyway) begin with a dollar sign. The only two configuration variables supported at present are `$return_address` and

$sendmail. You should not need to alter the assignment statements except for the values indicated between single quotes.

The $return_address variable should be set to the return address for the ticket-by-mail system. If you choose not to use the ticket-by-mail system, then set this to something that will be seen by a live person, as it will also be used as the return address when mailing customers the resolutions to their problems.

The $sendmail provides the full path to the **sendmail** executable and its required command-line options. At a minimum, the -oi and -t options are required; the -odq option queues the message in the local sendmail queue. If the host on which you run BATTS does not automatically send queued mail, remove this option. Many systems do not have the sendmail binary in /usr/sbin/sendmail as is standard on most Linux systems; other Unix flavors tend to use /usr/lib/sendmail instead. If the path is wrong, then sendmail will not be run and mail can not be sent.

The batts-reply.txt file includes the text of the message sent to users when a new ticket is generated. At the very least, change the e-mail address included in the message to point to your system.

Installing the files are easy. Simply copy them to your /etc directory, change the group to batts, and change the file mode to 0640. This can be done with the following sequence of commands executed from the config_files subdirectory of the BATTS distribution:

```
cp batts.myrc battsrc batts-reply.txt /etc
pushd /etc
chgrp batts batts.myrc battsrc batts-reply.txt
chmod 0640 batts.myrc battsrc batts-reply.txt
popd
```

Creating the Database

In the sql_scripts, there are several files that you will need to modify. First, edit the create-tables.sql script to point to your account and password. The fourth line of the file, which begins grant ALL on batts.* is the line that needs to be modified. If you have changed the database account name in the batts.myrc file, then change batts@localhost to point to the account name. It may be necessary to change localhsot to the actual host name of your system as well. The password, which is set by default to gimmeanewpassword should also be changed to match the setting from your batts.myrc file.

Next, modify people.txt, and change the e-mail address to match that of your trouble ticketing system. The file is tab-delimeted, and the database is very sensitive to the format. The name unassigned must not be changed. The name of the trouble ticket system, which is set by default to BATTS Trouble Ticket System may be changed, but caution is warranted.

In the customers.txt file, change internal to a short username to identify the organization running the ticketing system, and (following a **tab**), change Your Company Name Here to a reasonable long name for your company or organization.

Finally, you can run the **initialize-database** script to create the database. If you have set a password on MySQL's root account, run with the -p option.

Installing the Scripts

There are two scripts that are part of BATTS: `batts` and `ticket-via-email`. The first line of each of the scripts must point to your Perl installation: if your preferred version of Perl is in `/usr/local/bin`, change the part of the first line that reads `/usr/bin/perl` to read `/usr/local/bin/perl` instead.

The scripts merely need to be copied to a binary directory on your system; we recommend that they be installed in `/usr/local/bin`. They may be installed via the following sequence of commands executed from the `bin` subdirectory of the BATTS distribution:

```
cp batts ticket-via-email /usr/local/bin
pushd /usr/local/bin
chgrp batts batts ticket-via-email
chmod 0750 batts ticket-via-email
```

At this point, you may wish to install symbolic links for the various BATTS commands. If you are using a Bourne shell derivative such as the Korn Shell or BASH, you may execute the following command from the `/usr/local/bin` directory to install symbolic links for most of the BATTS commands:

```
for foo in `batts show-commands
| egrep -v '(help|mysql|touch)'`; do ln -s batts $foo; done
```

If you have a command on your system named `log`, you should add log to the **egrep** command, which eliminates commands from the list of installed symlinks. It is presumed that you do not wish to override **mysql** or **touch** commands, and the **help** would be confusing for most users, who will not intend to be using BATTS.

Installing the Mail Gateway

Installing the Mail gateway is not difficult, but requires some modest system administration experience. Because of the group-membership security requirement for BATTS, it's simplest to create a user who is a member of the `batts` group. On my system, I created a user named `trouble`; if this user will not by default receive mail sent to the return address you configured in the above configuration files, you should create an alias that points at this account.

Most Sendmail installations have **smrsh** (Sendmail Restricted Shell) configured. This requires that programs which may be executed by users be stored (or symbolically linked to) in the `smrsh` directory. On my Linux system, this is `/etc/smrsh`, but this may differ on other Unix flavors. From that directory, create a symbolic link to the **ticket-via-email** script. On my system, this may be done as follows:

```
pushd /etc/smrsh
ln -s /usr/local/bin/ticket-via-email ticket-via-email
popd
```

The next step is making the script execute when mail is sent to the appropriate e-mail address. Assuming that you created a user account for BATTS, create a `.forward` in that user's home directory which points to the script. The script name should be in quotes, with a pipe (|) prior to the filename within the quotes. On my system, trouble's `.forward` file reads as follows:

```
"|/etc/smrsh/ticket-via-email",root
```

The `,root` at the end of the line is optional. It mails a copy of the incoming ticket to everyone who is listed in the `root` alias. If instead you wanted the ticket to be processed by the script, mailed to everyone on the <`developers@example.net`> mailing list run on another machine, and to a user named `joe` on the local machine, this line could read

```
"|/etc/smrsh/ticket-via-email",developers@example.net,joe
```

The file should be writable only by the user; mode 0644 is fine. If other users can write to the `.forward` file, then Sendmail will refuse to execute it.

While it may be possible to use an alias in `/etc/aliases` and have the **ticket-via-email** script setgid to the `BATTS` group, my gut feeling is that this would be less secure than creating another user. I have no real evidence for that assertion, however.

At this point, BATTS is installed on your system. Send a test ticket in via the e-mail gateway and see that everything performs as expected.

Notes

1. http://www.perl.com/
2. http://www.mysql.com/
3. It is, only inside out!
4. http://dbi.symbolstone.org/

Chapter 3. A Brief Introduction To BATTS

Using BATTS is not difficult, if you are familiar with the Unix command line. If you are not, I recommend buying (and reading!) my book to learn all about Unix. [1] BATTS was designed to work simply and interact well with other Unix programs. This tutorial, however, concentrates on basic BATTS features.

People and Customers

The point of a trouble ticket system is to better server your customers. Most customers are people who work for organizations, which may or may not be the real customers but who really pay the bills. BATTS embodies this division by having *people* and *customers*.

People have an ID number, with which you need not be concerned, a user name, a long name, an e-mail address, and the customer for whom they work. By default, there is a customer 'unknown' (which you can think of as unaffiliated) and one (by default 'internal', though you should have changed this) for employees of your organization. Let's say that your real name is "John Smith," and your e-mail address is <`me@example.org`>. You may want your username to be the same as the user portion of your e-mail address; this is the default when tickets are submitted via the e-mail gateway. The command **`batts person-new me "John Smith" "me@example.org" internal`** will create a user with the short name of `me`, whose customer is `internal`. If you've changed the name of the first customer following unknown when installing BATTS, you will get an error if you try to use `internal`.

Note that **person-new** is a sub-command of the **batts**. In the rest of the examples, we will assume that the symbolic links have been installed, and that only the subcommand names need be typed. If you have not installed the symlinks on your system, imagine that each of the BATTS commands is prefaced by the name of the BATTS executable. Note also that the full name and e-mail address are surrounded by quotes. This is so that the system knows to treat each of these as a single item on the command line. This is *shell quoting*, and it is explained in almost any in-depth Unix tutorial.

As of the current release, BATTS does not distinguish between people who have access to modify tickets directly and people who can only submit tickets. If people should not have the power to modify tickets, they should not be given permission to run the BATTS executable, and should restrict themselves to the e-mail gateway.

A customer is similar to a person, in that there is a short name (the *customer username* (which is `internal` in the above example) and a long name, which should be the full name of the company. To create a new customer, the syntax is similar to the **person-new**, but slightly simpler: **`customer-new bigspender "Big Spender Enterprises"`**. In this case, `bigspender` is the customer short name, and `Big Spender Enterprises` (again, quoted in the shell) is the customer's full name.

If you wish to change information about a person or a customer after having created them, several commands are available to do this. Check Reference I, *BATTS Manual Pages* for further information.

Creating a Ticket

Creating a ticket is simple, but will feel strange if you are relatively unfamiliar with Unix. The command requires the e-mail address of the submitter and a short description: **`ticket-new me@example.net "I can't find my shoes"`** would create a ticket with John Smith as the customer and a short description of `I can't` `find my shoes`. However, you will not immediately be returned to the command prompt. Instead, you will have no prompt. Enter your long description, taking as many lines as necessary. End the long description entry process by typing **Control-D** on a blank line; you will be told the ticket number of the new ticket then be returned to the command prompt.

Why this odd behavior? BATTS wants to cooperate with other Unix programs, which can feed input into other programs. For example, let's say that John Smith had written up a long description of his problem in the file `my-problem.txt`. To enter this problem into BATTS, he need only run the command **`ticket-new me@example.net "I can't find my shoes" < my-problem.txt`**. Or perhaps his complaint is that the user `joe` is running too many processes. He can create his complaint by running the command **`ps -fu joe | ticket-new me@example.net "joe runs too much stuff"`**. **ps -fu joe** is a standard Unix command that will display all of joe's processes. This list will become the long description for the ticket.

Note that if you provide an e-mail address that does not match any known user, a temporary user will be created so that the ticket can be entered into the system. It is highly recommended that you modify this customer immediately after ticket creation.

Looking at Tickets

Creating tickets is good, but what do you do with them once they're in the system? The first thing you probably wish to do is to see all of the new tickets in the system. This can be done with the **show-new** command. It will print out a list of new tickets, their creators, their status, and their short descriptions. To look at a particular ticket, for example ticket number 123, run **`show-ticket 123`**. A full display of the ticket information, including the long description will follow.

You may also view all tickets that have not been resolved with the **show-open** command. To view only tickets that he has created or been assigned, John Smith could run the command **`show-my me`**. All of these commands, along with commands to modify existing tickets, are documented in Reference I, *BATTS Manual Pages*.

Adding Events to a Ticket

The **log** can add events to a ticket. Events may be additional information gleaned via discussions with the subject, or may include work performed on the ticket. The arguments required by **log** are, in order, *`ticket-id`*, *`username`*, *`billing-code`*, and then either a `Y` or `N` indicating if the customer should be permitted to see the log entry, or if it is only for internal use. (This last piece is not enforced at present, though it should be in later versions of BATTS.)

The only piece of information we have not discussed previously is the billing code. To see a list of available billing codes, use the **show-billing** command. The code `informational` is available for informational messages. To create a note on ticket 123, John Smith would run the command **`log 123 me informational Y`**, after which he could enter text, as with writing the long description for a ticket. Again, type **Control-D** to end text entry. A log ID number will be generated and returned to you.

To see a particular log entry, simply run **show-log *`log-id`***; to see all log entries associated with a particular ticket, run **show-ticket-log *`ticket-id`***.

If you wish to create a log event with a duration (that is, you started work at one point and finished later), simply create the log entry when you begin work. When you are done with the work, run **log-end *`log-id`***.

Resolving a Ticket

To resolve a ticket, run **resolve-ticket** ***`ticket-number`***. You can then enter text, typing **Control-D** to end text entry. The text of this resolution will be mailed to the person who entered the ticket into the system. (If you changed the ticket requestor via the **requestor-ticket** command, that person will be mailed instead of the ticket creator.) To suppress mailing the response text, run **resolve-ticket –nomail** ***`ticket-number`*** instead.

There are some additional commands, to mark tickets as being changed when another ticket is altered. **pending-on** will set a ticket as pending until the given ticket number is resolved, at which point it will revert to assigned (or new) status; **resolve-when** will resolve the ticket when the master ticket is resolved. Check Reference I, *BATTS Manual Pages* for more information.

The BATTS E-Mail Gateway

The BATTS e-mail gateway can really simplify things, from a user's perspective. To submit a ticket, mail the system's e-mail address. An auto-reply will be sent that indicates the ticket number in brackets on the subject line. Any future e-mails sent to the trouble ticketing system which include the ticket ID in square brackets will be added as informational entries in the ticket's event log.

If a customer responds to the ticket resolution message, or in general sends in a message regarding a resolved ticket, that ticket will be re-opened and assigned as it had previously been assigned. This is because usually customers reply to complain that the problem is not actually fixed. If the customer simply responds to say 'thanks,' you may resolve the ticket without modifying the ticket status or mailing the customer using the **status-ticket** command.

Notes

1. *Think Unix*, by Jon Lasser, Que. 2000. See

   2. http://www.cluestickconsulting.com/think-unix/
   for more information.

2. http://www.cluestickconsulting.com/think-unix/

I. BATTS Manual Pages

assign-ticket

Name

`assign-ticket` — Assign a ticket to a user

Synopsis

assign-ticket *`ticket-number`* *`username`*

Description

Assign the ticket identified by *`ticket-number`* to the user identified by *`username`*. Leading zeros in the ticket number are ignored; if the user name contains several words, as is likely for users whose names have been derived from e-mail, *`username`* must be surrounded by quotes.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

billing-new

Name

`billing-new` — Create a new category for billing items

Synopsis

billing-new *`new-billing-name`*

Description

Create a new category of billing items named *`new-billing-name`*

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

category-new

Name

`category-new` — Create a new category for tickets

Synopsis

category-new *`new-category-name`*

Description

Create a new category of tickets named *`new-category-name`*

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1),

show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

category-ticket

Name

`category-ticket` — Change the category of a ticket

Synopsis

category-ticket *`ticket-number category-name`*

Description

Ticket *`ticket-number`* is given a category of *`category-name`*.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

close-ticket

Name

`close-ticket` — Close a resolved ticket

Synopsis

close-ticket *ticket-number*

Description

Set the status of the resolved ticket identified by *ticket-number* to closed. If the ticket's status is not resolved, **close-ticket** will issue an error and exit.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

customer-new

Name

`customer-new` — Add a new customer to BATTS

Synopsis

customer-new *customer-username customer-fullname*

Description

Add a new customer to BATTS. A customer is really just a wrapper for a group of people, and (so far) does not have much purpose. Eventually, BATTS will use customers as a notion for billing and as a notion for permission to see or operate on a group of tickets.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

customer-ticket

Name

`customer-ticket` — Assign a ticket to a customer

Synopsis

customer-ticket *`ticket-number customer-username`*

Description

Set the customer for a ticket *`ticket-number`* to the customer named *`customer-name`*. (This is an organizational customer, *NOT* a person.)

A customer is really just a wrapper for a group of people, and (so far) does not have much purpose. Eventually, BATTS will use customers as a notion for billing and as a notion for permission to see or operate on a group of tickets.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1),

show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

duplicate-ticket

Name

`duplicate-ticket` — Mark a ticket as a duplicate.

Synopsis

duplicate-ticket *`duplicate-ticket-number`* *`master-ticket-number`*

Description

Ticket number *`duplicate-ticket-number`* is marked as a duplicate of ticket number *`master-ticket-number`*. When the latter ticket is resolved, the former is resolved as well.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

help

Name

`help` — Produce a listing of available commands and their syntax

Synopsis

help *`username`*

Description

Produce a list of commands supported by the BATTS ticketing system, including the arguments and options taken by each individual command.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

log

Name

`log` — Add an event log message to a ticket

Synopsis

log *`ticket-number username billing-name public?`*

Description

Add an event to the log for ticket *`ticket-number`*. The log is created as the user *`username`* with an associated billing code of *`billing-name`*. If *`public?`* is set to `Y`, then the log message can be read by all people authorized to see the ticket, including the customer. Tickets with *`public?`* set to `N` can be read only by authorized ticket-workers.

Text for the log entry is taken from `STDIN`. The time on the log entry is set to the present moment and is considered to be instantaneous.

Bugs

Security is enforced only by standard Unix permission bits. The `public?` option is not enforced; it is waiting on a meaningful Web interface first. There is no way to create an event with arbitrary start and end times.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

log-end

Name

`log-end` — Set the end-time on a log entry

Synopsis

log-end *`log-id-number`*

Description

Set the end time on event log id *`log-id-number`* to the present moment. This creates a ticket log entry that stretches from the ticket's start log time to the present.

Bugs

Security is enforced only by standard Unix permission bits. There is no way to create an event with arbitrary start and end times.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1),

show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

longdesc-ticket

Name

`longdesc-ticket` — Rewrite the long description on a ticket

Synopsis

longdesc-ticket *`ticket-number`*

Description

Rewrite the new description for ticket *`ticket-number`*, taking the text of that description from `STDIN`.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

mysql

Name

`mysql` —

Synopsis

batts mysql

Description

Enter the MySQL command interpreter as the BATTS user. This is a shortcut to avoid typing long command lines to run MySQL with proper permissions.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

pending-on

Name

`pending-on` — Mark a ticket as waiting for another ticket's completion.

Synopsis

pending-on *`pending-ticket-number`* *`master-ticket-number`*

Description

Ticket number *`pending-ticket-number`* is marked as a pending on ticket number *`master-ticket-number`*. When the latter ticket is resolved, the former is marked as either new (if it had not previously been assigned to a user) or assigned (if it had already been assigned).

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

person-fullname

Name

`person-fullname` — Set a person's full name in the people database

Synopsis

person-fullname *`username`* *`full-name`*

Description

For the person whose username is *`username`*, set their full name to *`full-name`*.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

person-new

Name

`person-new` — Create a new person in the database

Synopsis

person-new *`username full-name e-mail customer-username`*

Description

Add a new person to the database, with parameters as specified. Note that if there are spaces in the user name and customer name, these need to be properly quoted in the shell.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

person-username

Name

`person-username` —

Synopsis

person-username *`e-mail username`*

Description

Set the username of the person whose e-mail address is `e-mail` to `username`.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

priority-ticket

Name

`priority-ticket` — Change the priority on a ticket

Synopsis

priority-ticket *`ticket-number priority-name`*

Description

Change the priority on ticket `ticket-number` to `priority-name`. For a complete list of priorities in descending order, see show-priorities(1).

Bugs

Security is enforced only by standard Unix permission bits. It should be possible to change the priority to a given priority *number* too.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

requestor-ticket

Name

`requestor-ticket` — Change the requestor of a ticket

Synopsis

requestor-ticket *`ticket-number username`*

Description

Change the requestor of ticket *`ticket-number`* to the person *`username`*. The requestor is similar to the creator of the ticket, but the creator of the ticket is never changed and is superseded by the requestor. (That is to say, the requestor is the person who will receive mail when the ticket is resolved.)

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

resolve-ticket

Name

`resolve-ticket` — Set the status of a given ticket to resolved.

Synopsis

resolve-ticket [-- [no]mail] `ticket-number`

Description

Set the status of the ticket identified by `ticket-number` to resolved. Take text from `STDIN` and store that text in the resolution field of the ticket.

When a ticket is resolved, other tickets are scanned. Tickets that are `resolved on` or are a `duplicate of` the current ticket are also resolved; tickets that are `pending on` the current ticket will have their status reset to `assigned` or `new`, depending on whether the ticket had previously been assigned.

By default, mail is sent to the ticket's requestor; if the `--nomail` option is provided, sending mail will be suppressed for the ticket and all subsidiary tickets.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

resolve-when

Name

`resolve-when` — Mark a ticket as resolved when another ticket is resolved

Synopsis

resolve-when `subsidiary-ticket-number master-ticket-number`

Description

Ticket number *subsidiary-ticket-number* is marked as waiting for *master-ticket-number* to be resolved. When this master ticket is resolved, the subsidiary ticket is resolved as well.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

shortdesc-ticket

Name

`shortdesc-ticket` — Reset the short description of a ticket

Synopsis

shortdesc-ticket *ticket-number short-description*

Description

Set the short description of ticket *ticket-number* to the text of *short-description*. Spaces in the short description must be protected from the shell via quoting.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-billing

Name

`show-billing` — Display the list of available billing codes

Synopsis

show-billing

Description

Display a list of billing codes available for the classification of ticket log events, one per line.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-cat-ticket

Name

`show-cat-ticket` — Show short list of all tickets in a category

Synopsis

show-cat-ticket [--html] *`category-name`*

Description

Display a list of all open tickets belonging to the category given on the command line. If the `--html` is given, display the ticket list as an HTML table.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-categories

Name

`show-categories` — Show the list of BATTS categories

Synopsis

show-categories

Description

Display a list of categories available for the classification of tickets, one per line.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-commands

Name

`show-commands` —

Synopsis

show-commands

Description

Display a brief list of available BATTS commands, one per line.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-cust-ticket

Name

`show-cust-ticket` — Display all tickets belonging to a customer

Synopsis

show-cust-ticket [--html] *`customer-username`*

Description

Display all tickets belonging to customer *`customer-username`*. If the `html` option is given, list the tickets in an HTML table.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-customers

Name

`show-customers` — Display a list of customers

Synopsis

show-customers

Description

Display a list of customers, one per line. The customer's short name is followed by a space, a colon, and the customer's long name.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-log

Name

`show-log` — Display a single event log entry

Synopsis

show-log [--html] *`log-entry-number`*

Description

Display the log event identified by *`log-entry-number`*. If the `--html` is given, display the log entry in an HTML table.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-my

Name

`show-my` — Display open tickets belonging to a user

Synopsis

show-my [--html] *`username`*

Description

Display all current tickets belonging to user *`username`*. Current tickets are those that are not closed, resolved, or marked as a duplicate of another ticket. Tickets displayed are sorted first by priority and then by date. If the option `--html` is given, display the output as an HTML table.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-new

Name

`show-new` — Show new tickets

Synopsis

show-new [--html]

Description

Display all tickets whose status is new, in ticket-number order. If the `--html` option is provided, display the tickets in an HTML table.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-open

Name

`show-open` — List all open tickets

Synopsis

show-open [--html]

Description

List all open tickets, in ticket number order. If the `--html` option is given, display the tickets in an HTML table.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-people

Name

`show-people` — Display a list of people known by the system

Synopsis

show-people

Description

Display a list of people known by the BATTS ticketing system. The format of the list is the username, e-mail address, full name, and company affiliation; the field delimiter is a space, a colon, and another space.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-priorities

Name

`show-priorities` — List all priorities available within the system

Synopsis

show-priorities

Description

List all priorities available within the system, one per line, from highest priority to lowest.

The priority list is as follows:

1. critical
2. extremely high
3. very high
4. high
5. medium-high
6. medium
7. medium-low
8. low
9. very low
10. as time permits

Bugs

Security is enforced only by standard Unix permission bits. Perhaps this should be documentation and not a separate command. The list of priorities should get its own entry in a different section of the manual.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-status

Name

`show-status` — Display a list of available status codes.

Synopsis

show-status

Description

Display status codes provided by the system, one per line.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-ticket

Name

`show-ticket` — Display the entire contents of a ticket

Synopsis

show-ticket *`ticket-number`*

Description

Display the full contents of ticket `ticket-number`. This includes the ticket number, its creator, their company, the creation time, to whom the ticket is assigned, the time that the ticket was last modified, its status, its category, its priority, the short description, and the long description.

Bugs

Security is enforced only by standard Unix permission bits.

Display of dates is in native MySQL timestamp format, which is unreadable to normal people. There should be a way to display log entries for the ticket, not that the log functions are working yet.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

show-ticket-log

Name

`show-ticket-log` — Display a single event log entry

Synopsis

show-ticket-log [--html] `ticket-number`

Description

Display all log events for the ticket `ticket-number`. If the `--html` is given, display the log entries in an HTML table.

Bugs

Security is enforced only by standard Unix permission bits.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

status-ticket

Name

`status-ticket` — Change the status code for a ticket

Synopsis

status-ticket *`ticket-number`* *`status-name`*

Description

Change the status code for ticket *`ticket-number`* to *`status-name`*.

Bugs

Security is enforced only by standard Unix permission bits. This command can be used to evade the mailing of ticket resolutions to customers, or evade the requirement that a ticket be resolved before it can be closed. Note that if a ticket is resolved or closed via **status-ticket**, subsidiary tickets are not automatically resolved. This behavior is strongly discouraged, of course.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

ticket-new

Name

`ticket-new` — Create a new ticket

Synopsis

ticket-new *`customer-email`* *`short-description`*

Description

Create a new ticket requested by person *`customer-email`* with a short description of *`short-description`*. Priority is set to medium, ticket is set to new and assigned to 'new ticket.' The long description text is read from `STDIN`.

Bugs

Security is enforced only by standard Unix permission bits. More options should be available for creating new tickets.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

ticket-via-email

Name

`ticket-via-email` — Create a new ticket based on an e-mail message

Synopsis

ticket-via-email

Description

Create a new ticket based on an e-mail message read via `STDIN`. The creator of the ticket is set to the person listed in the `From` field; if no such e-mail address is listed in the person database, their account is created. The short description is set to the text of the subject line of the message, and the long description is set to the body of the message.

Bugs

This code relies on the ticket-new command and is bound by all of its restrictions.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

touch

Name

`touch` — Update the last-modified time for a ticket.

Synopsis

touch *`ticket-number`*

Description

Update the last-modified time for ticket *`ticket-number`*.

Bugs

Security is enforced only by standard Unix permission bits. This command is likely to be confused with the standard Unix **touch** command.

See Also

assign-ticket(1), billing-new(1), category-new(1), category-ticket(1), close-ticket(1), customer-new(1), customer-ticket(1), duplicate-ticket(1), help(1), log(1), log-end(1), longdesc-ticket(1), mysql(1), pending-on(1), person-fullname(1), person-new(1), person-username(1), priority-ticket(1), requestor-ticket(1), resolve-ticket(1), resolve-when(1), shortdesc-ticket(1), show-billing(1), show-cat-ticket(1), show-categories(1), show-commands(1), show-cust-ticket(1), show-customers(1), show-log(1), show-my(1), show-new(1), show-open(1), show-people(1), show-priorities(1), show-status(1), show-ticket(1), show-ticket-log(1), status-ticket(1), ticket-new(1), ticket-via-email(1), touch(1)

Appendix A. GNU General Public License

Preamble

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

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

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

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

We protect your rights with two steps:

1. copyright the software, and
2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

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

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

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

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

Section 0

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

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

Section 1

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

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

Section 2

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

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

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

c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

Exception:: If the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

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

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

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

Section 3

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

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

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

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

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

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

Section 4

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

Section 5

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

Section 6

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

Section 7

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

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

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

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

Section 8

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

Section 9

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

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

Section 10

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

NO WARRANTY

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

Section 12

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED