NAME

bts − developers' command line interface to the Debian Bug Tracking System

SYNOPSIS

bts [*options*] *command* [*args*] [**#***comment*] [**.**|**,** *command* [*args*] [**#***comment*]] ...

DESCRIPTION

This is a command line interface to the Debian Bug Tracking System (BTS), intended mainly for use by developers. It lets the BTS be manipulated using simple commands that can be run at the prompt or in a script, does various sanity checks on the input, and constructs and sends a mail to the BTS control address for you. A local cache of web pages and e−mails from the BTS may also be created and updated.

In general, the command line interface is the same as what you would write in a mail to control@bugs.debian.org, just prefixed with "bts". For example:

```
 % bts severity 69042 normal
 % bts merge 69042 43233
 % bts retitle 69042 blah blah
```

A few additional commands have been added for your convenience, and this program is less strict about what constitutes a valid bug number. For example, "severity Bug#85942 normal" is understood, as is "severity #85942 normal". (Of course, your shell may regard "#" as a comment character though, so you may need to quote it!)

Also, for your convenience, this program allows you to abbreviate commands to the shortest unique substring (similar to how cvs lets you abbreviate commands). So it understands things like "bts cl 85942".

It is also possible to include a comment in the mail sent to the BTS. If your shell does not strip out the comment in a command like "bts severity 30321 normal #inflated severity", then this program is smart enough to figure out where the comment is, and include it in the email. Note that most shells do strip out such comments before they get to the program, unless the comment is quoted. (Something like "bts severity #85942 normal" will not be treated as a comment!)

You can specify multiple commands by separating them with a single dot, rather like **update−rc.d**; a single comma may also be used; all the commands will then be sent in a single mail. It is important the dot/comma is surrounded by whitespace so it is not mistaken for part of a command. For example (quoting where necessary so that **bts** sees the comment):

```
 % bts severity 95672 normal , merge 95672 95673 \#they are the same!
```

The abbreviation "it" may be used to refer to the last mentioned bug number, so you could write:

```
 % bts severity 95672 wishlist , retitle it "bts: please add a --foo option"
```

Please use this program responsibly, and do take our users into consideration.

OPTIONS

bts examines the **devscripts** configuration files as described below. Command line options override the configuration file settings, though.

−o, **−−offline**
Make **bts** use cached bugs for the **show** and **bugs** commands, if a cache is available for the requested data. See the **cache** command, below for information on setting up a cache.

−−online, **−−no−offline**
Opposite of **−−offline**; overrides any configuration file directive to work offline.

−n, **−−no−action**
Do not send emails but print them to standard output.

−−cache, **−−no−cache**
Should we attempt to cache new versions of BTS pages when performing **show/bugs** commands? Default is to cache.

−−cache−mode={**min**|**mbox**|**full**}

When running a **bts cache** command, should we only mirror the basic bug (**min**), or should we also mirror the mbox version (**mbox**), or should we mirror the whole thing, including the mbox and the boring attachments to the BTS bug pages and the acknowledgement emails (**full**)? Default is **min**.

−−cache−delay=*seconds*

Time in seconds to delay between each download, to avoid hammering the BTS web server. Default is 5 seconds.

−−mbox

Open a mail reader to read the mbox corresponding to a given bug number for **show** and **bugs** commands.

−−mailreader=*READER*

Specify the command to read the mbox. Must contain a "`%s`" string (unquoted!), which will be replaced by the name of the mbox file. The command will be split on white space and will not be passed to a shell. Default is '**mutt −f** `%s`'. (Also, *%%* will be substituted by a single *%* if this is needed.)

−−cc−addr=*CC_EMAIL_ADDRESS*

Send carbon copies to a list of users. *CC_EMAIL_ADDRESS* should be a comma-separated list of email addresses.

−−use−default−cc

Add the addresses specified in the configuration file option **BTS_DEFAULT_CC** to the list specified using **−−cc−addr**. This is the default.

−−no−use−default−cc

Do not add addresses specified in **BTS_DEFAULT_CC** to the carbon copy list.

−−sendmail=*SENDMAILCMD*

Specify the **sendmail** command. The command will be split on white space and will not be passed to a shell. Default is */usr/sbin/sendmail*. The **−t** option will be automatically added if the command is */usr/sbin/sendmail* or */usr/sbin/exim**. For other mailers, if they require a **−t** option, this must be included in the *SENDMAILCMD*, for example: **−−sendmail="/usr/sbin/mymailer −t"**.

−−mutt

Use **mutt** for sending of mails. Default is not to use **mutt**, except for some commands.

Note that one of **`$DEBEMAIL`** or **`$EMAIL`** must be set in the environment in order to use **mutt** to send emails.

−−no−mutt

Don't use **mutt** for sending of mails.

−−soap−timeout=*SECONDS*

Specify a timeout for SOAP calls as used by the **select** and **status** commands.

−−smtp−host=*SMTPHOST*

Specify an SMTP host. If given, **bts** will send mail by talking directly to this SMTP host rather than by invoking a **sendmail** command.

The host name may be followed by a colon (":") and a port number in order to use a port other than the default. It may also begin with "ssmtp://" or "smtps://" to indicate that SMTPS should be used.

If SMTPS not specified, **bts** will still try to use STARTTLS if it's advertised by the SMTP host.

Note that one of **`$DEBEMAIL`** or **`$EMAIL`** must be set in the environment in order to use direct SMTP connections to send emails.

Note that when sending directly via an SMTP host, specifying addresses in **−−cc−addr** or **BTS_DEFAULT_CC** that the SMTP host will not relay will cause the SMTP host to reject the entire mail.

Note also that the use of the **reassign** command may, when either **−−interactive** or **−−force−interactive** mode is enabled, lead to the automatic addition of a Cc to *$newpackage*@packages.debian.org. In these cases, the note above regarding relaying applies. The submission interface (port 587) on reportbug.debian.org does not support relaying and, as such, should not be used as an SMTP server for **bts** under the circumstances described in this paragraph.

−−smtp−username=*USERNAME*, **−−smtp−password=***PASSWORD*

Specify the credentials to use when connecting to the SMTP server specified by **−−smtp−host**. If the server does not require authentication then these options should not be used.

If a username is specified but not a password, **bts** will prompt for the password before sending the mail.

−−smtp−helo=*HELO*

Specify the name to use in the *HELO* command when connecting to the SMTP server; defaults to the contents of the file */etc/mailname*, if it exists.

Note that some SMTP servers may reject the use of a *HELO* which either does not resolve or does not appear to belong to the host using it.

−−bts−server

Use a debbugs server other than https://bugs.debian.org.

−f, **−−force−refresh**

Download a bug report again, even if it does not appear to have changed since the last **cache** command. Useful if a **−−cache−mode=full** is requested for the first time (otherwise unchanged bug reports will not be downloaded again, even if the boring bits have not been downloaded).

−−no−force−refresh

Suppress any configuration file **−−force−refresh** option.

−−only−new

Download only new bugs when caching. Do not check for updates in bugs we already have.

−−include−resolved

When caching bug reports, include those that are marked as resolved. This is the default behaviour.

−−no−include−resolved

Reverse the behaviour of the previous option. That is, do not cache bugs that are marked as resolved.

−−no−ack

Suppress acknowledgment mails from the BTS. Note that this will only affect the copies of messages CCed to bugs, not those sent to the control bot.

−−ack

Do not suppress acknowledgement mails. This is the default behaviour.

−i, **−−interactive**

Before sending an e−mail to the control bot, display the content and allow it to be edited, or the sending cancelled.

−−force−interactive

Similar to **−−interactive**, with the exception that an editor is spawned before prompting for confirmation of the message to be sent.

−−no−interactive

Send control e−mails without confirmation. This is the default behaviour.

−q, **−−quiet**

When running **bts cache**, only display information about newly cached pages, not messages saying already cached. If this option is specified twice, only output error messages (to stderr).

−−no−conf, **−−noconf**
Do not read any configuration files. This can only be used as the first option given on the command-line.

COMMANDS

For full details about the commands, see the BTS documentation. <https://www.debian.org/Bugs/server−control>

show [*options*] [*bug number* | *package* | *maintainer* | **:**] [*opt*=*val* ...]
show [*options*] [**src:***package* | **from:***submitter*] [*opt*=*val* ...]
show [*options*] [**tag:***tag* | **usertag:***tag*] [*opt*=*val* ...]
show [**release-critical** | **release−critical/**... | **RC**]
This is a synonym for **bts bugs**.

bugs [*options*] [*bug_number* | *package* | *maintainer* | **:**] [*opt*=*val* ...]
bugs [*options*] [**src:***package* | **from:***submitter*] [*opt*=*val* ...]
bugs [*options*] [**tag:***tag* | **usertag:***tag*] [*opt*=*val* ...]
bugs [**release-critical** | **release−critical/**... | **RC**]
Display the page listing the requested bugs in a web browser using **sensible−browser** (1).

Options may be specified after the **bugs** command in addition to or instead of options at the start of the command line: recognised options at this point are: **−o/−−offline/−−online**, **−m/−−mbox**, **−−mailreader** and **−−**[**no−**]**cache**. These are described earlier in this manpage. If either the **−o** or **−−offline** option is used, or there is already an up-to-date copy in the local cache, the cached version will be used.

The meanings of the possible arguments are as follows:

(none) If nothing is specified, **bts bugs** will display your bugs, assuming that either **DEBEMAIL** or **EMAIL** (examined in that order) is set to the appropriate email address.

bug_number
Display bug number *bug_number*.

package Display the bugs for the package *package*.

src:*package*
Display the bugs for the source package *package*.

maintainer
Display the bugs for the maintainer email address *maintainer*.

from:*submitter*
Display the bugs for the submitter email address *submitter*.

tag:*tag* Display the bugs which are tagged with *tag*.

usertag:*tag*
Display the bugs which are tagged with usertag *tag*. See the BTS documentation for more information on usertags. This will require the use of a **users=***email* option.

: Details of the bug tracking system itself, along with a bug-request page with more options than this script, can be found on https://bugs.debian.org/. This page itself will be opened if the command ’bts bugs :’ is used.

release-critical, **RC**
Display the front page of the release-critical pages on the BTS. This is a synonym for https://bugs.debian.org/release−critical/index.html. It is also possible to say release−critical/debian/main.html and the like. RC is a synonym for release−critical/other/all.html.

After the argument specifying what to display, you can optionally specify options to use to format the page or change what it displayed. These are passed to the BTS in the URL downloaded. For example, pass dist=stable to see bugs affecting the stable version of a package, version=1.0 to see bugs affecting that version of a package, or reverse=yes to display newest messages first in a bug log.

If caching has been enabled (that is, **−−no−cache** has not been used, and **BTS_CACHE** has not been set to **no**), then any page requested by **bts show** will automatically be cached, and be available offline thereafter. Pages which are automatically cached in this way will be deleted on subsequent "**bts show|bugs|cache**" invocations if they have not been accessed in 30 days. Warning: on a filesystem mounted with the "noatime" option, running "**bts show|bugs**" does not update the cache files' access times; a cached bug will then be subject to auto-cleaning 30 days after its initial download, even if it has been accessed in the meantime.

Any other **bts** commands following this on the command line will be executed after the browser has been exited.

The desired browser can be specified and configured by setting the **BROWSER** environment variable. The conventions follow those defined by Eric Raymond at http://www.catb.org/~esr/BROWSER/; we here reproduce the relevant part.

The value of **BROWSER** may consist of a colon-separated series of browser command parts. These should be tried in order until one succeeds. Each command part may optionally contain the string **%s**; if it does, the URL to be viewed is substituted there. If a command part does not contain **%s**, the browser is to be launched as if the URL had been supplied as its first argument. The string *%%* must be substituted as a single %.

Rationale: We need to be able to specify multiple browser commands so programs obeying this convention can do the right thing in either X or console environments, trying X first. Specifying multiple commands may also be useful for people who share files like *.profile* across multiple systems. We need **%s** because some popular browsers have remote-invocation syntax that requires it. Unless *%%* reduces to %, it won't be possible to have a literal **%s** in the string.

For example, on most Linux systems a good thing to do would be:

BROWSER='mozilla −raise −remote "openURL(%s,new−window)":links'

select [*key***:***value ...*]

Uses the SOAP interface to output a list of bugs which match the given selection requirements.

The following keys are allowed, and may be given multiple times.

package Binary package name.

source Source package name.

maintainer
E−mail address of the maintainer.

submitter
E−mail address of the submitter.

severity Bug severity.

status Status of the bug. One of **open**, **done**, or **forwarded**.

tag Tags applied to the bug. If **users** is specified, may include usertags in addition to the standard tags.

owner Bug's owner.

correspondent
Address of someone who sent mail to the log.

affects Bugs which affect this package.

bugs List of bugs to search within.

users Users to use when looking up usertags.

archive Whether to search archived bugs or normal bugs; defaults to **0** (i.e. only search normal bugs). As a special case, if archive is **both**, both archived and unarchived bugs are returned.

For example, to select the set of bugs submitted by jrandomdeveloper@example.com and tagged **wontfix**, one would use

```
bts select submitter:jrandomdeveloper@example.com tag:wontfix
```

If a key is used multiple times then the set of bugs selected includes those matching any of the supplied values; for example

```
bts select package:foo severity:wishlist severity:minor
```

returns all bugs of package foo with either wishlist or minor severity.

status [*bug* | **file:***file* | **fields:***field*[**,***field* ...] | **verbose**] ...

Uses the SOAP interface to output status information for the given bugs (or as read from the listed files — use **–** to indicate STDIN).

By default, all populated fields for a bug are displayed.

If **verbose** is given, empty fields will also be displayed.

If **fields** is given, only those fields will be displayed. No validity checking is performed on any specified fields.

clone *bug new_ID* [*new_ID ...*]

The **clone** control command allows you to duplicate a *bug* report. It is useful in the case where a single report actually indicates that multiple distinct bugs have occurred. "New IDs" are negative numbers, separated by spaces, which may be used in subsequent control commands to refer to the newly duplicated bugs. A new report is generated for each new ID.

done *bug* [*version*]

Mark a *bug* as Done. This forces interactive mode since done messages should include an explanation why the bug is being closed. You should specify which *version* of the package closed the bug, if possible.

reopen *bug* [*submitter*]

Reopen a *bug*, with optional *submitter*.

archive *bug*

Archive a *bug* that has previously been archived but is currently not. The *bug* must fulfill all of the requirements for archiving with the exception of those that are time-based.

unarchive *bug*

Unarchive a *bug* that is currently archived.

retitle *bug title*

Change the *title* of the *bug*.

summary *bug* [*messagenum*]

Select a message number that should be used as the summary of a *bug*.

If no message number is given, the summary is cleared.

submitter *bug* [*bug* ...] *submitter-email*

Change the submitter address of a *bug* or a number of bugs, with **!** meaning 'use the address on the current email as the new submitter address'.

reassign *bug* [*bug* ...] *package* [*version*]

Reassign a *bug* or a number of bugs to a different *package*. The *version* field is optional; see the explanation at <https://www.debian.org/Bugs/server–control>.

found *bug* [*version*]

Indicate that a *bug* was found to exist in a particular package version. Without *version*, the list of fixed versions is cleared and the bug is reopened.

notfound *bug version*

Remove the record that *bug* was encountered in the given version of the package to which it is assigned.

fixed *bug version*

Indicate that a *bug* was fixed in a particular package version, without affecting the *bug*'s open/closed status.

notfixed *bug version*

Remove the record that a *bug* was fixed in the given version of the package to which it is assigned.

This is equivalent to the sequence of commands "**found** *bug version*“, ”**notfound** *bug version*".

block *bug* **by|with** *bug* [*bug* ...]

Note that a *bug* is blocked from being fixed by a set of other bugs.

unblock *bug* **by|with** *bug* [*bug* ...]

Note that a *bug* is no longer blocked from being fixed by a set of other bugs.

merge *bug bug* [*bug* ...]

Merge a set of bugs together.

forcemerge *bug bug* [*bug* ...]

Forcibly merge a set of bugs together. The first *bug* listed is the master bug, and its settings (those which must be equal in a normal **merge**) are assigned to the bugs listed next.

unmerge *bug*

Unmerge a *bug*.

tag *bug* [+|−|=] *tag* [*tag* ...]
tags *bug* [+|−|=] *tag* [*tag* ...]

Set or unset a *tag* on a *bug*. The tag may either be the exact tag name or it may be abbreviated to any unique tag substring. (So using **fixed** will set the tag **fixed**, not **fixed-upstream**, for example, but **fix** would not be acceptable.) Multiple tags may be specified as well. The two commands (tag and tags) are identical. At least one tag must be specified, unless the = flag is used, where the command

```
  bts tags <bug> =
```

will remove all tags from the specified *bug*.

Adding/removing the **security** tag will add ‘‘team\@security.debian.org’’ to the Cc list of the control email.

The list of valid tags and their significance is available at <https://www.debian.org/Bugs/Developer#tags>. The current valid tags are:

patch, wontfix, moreinfo, unreproducible, fixed, help, security, upstream, pending, d−i, confirmed, ipv6, lfs, fixed-upstream, l10n, newcomer, a11y, ftbfs

There is also a tag for each release of Debian since ‘‘potato’’. Note that this list may be out of date, see the website for the most up to date source.

affects *bug* [+|−|=] *package* [*package* ...]

Indicates that a *bug* affects a *package* other than that against which it is filed, causing the *bug* to be listed by default in the *package* list of the other *package*. This should generally be used where the *bug* is severe enough to cause multiple reports from users to be assigned to the wrong package. At least one *package* must be specified, unless the = flag is used, where the command

```
  bts affects <bug> =
```

will remove all indications that *bug* affects other packages.

user *email*

Specify a user *email* address before using the **usertags** command.

usertag *bug* [**+**|**−**|**=**] *tag* [*tag* ...]
usertags *bug* [**+**|**−**|**=**] *tag* [*tag* ...]

Set or unset a user tag on a *bug*. The *tag* must be the exact tag name wanted; there are no defaults or checking of tag names. Multiple tags may be specified as well. The two commands (**usertag** and **usertags**) are identical. At least one *tag* must be specified, unless the **=** flag is used, where the command

```
  bts usertags <bug> =
```

will remove all user tags from the specified *bug*.

claim *bug* [*claim*]

Record that you have claimed a *bug* (e.g. for a bug squashing party). *claim* should be a unique token allowing the bugs you have claimed to be identified; an e−mail address is often used.

If no *claim* is specified, the environment variable **DEBEMAIL** or **EMAIL** (checked in that order) is used.

unclaim *bug* [*claim*]

Remove the record that you have claimed a bug.

If no *claim* is specified, the environment variable **DEBEMAIL** or **EMAIL** (checked in that order) is used.

severity *bug severity*

Change the *severity* of a *bug*. Available severities are: **wishlist**, **minor**, **normal**, **important**, **serious**, **grave**, **critical**. The severity may be abbreviated to any unique substring.

forwarded *bug address*

Mark the *bug* as forwarded to the given *address* (usually an email address or a URL for an upstream bug tracker).

notforwarded *bug*

Mark a *bug* as not forwarded.

package [*package* ...]

The following commands will only apply to bugs against the listed *package*s; this acts as a safety mechanism for the BTS. If no packages are listed, this check is turned off again.

limit [*key*[**:***value*]] ...

The following commands will only apply to bugs which meet the specified criterion; this acts as a safety mechanism for the BTS. If no *value*s are listed, the limits for that *key* are turned off again. If no *key*s are specified, all limits are reset.

submitter
E−mail address of the submitter.

date Date the bug was submitted.

subject Subject of the bug.

msgid Message-id of the initial bug report.

package Binary package name.

source Source package name.

tag Tags applied to the bug.

severity Bug severity.

owner Bug's owner.

affects Bugs affecting this package.

archive Whether to search archived bugs or normal bugs; defaults to **0** (i.e. only search normal bugs). As a special case, if archive is **both**, both archived and unarchived bugs are returned.

For example, to limit the set of bugs affected by the subsequent control commands to those submitted by jrandomdeveloper@example.com and tagged **wontfix**, one would use

bts limit submitter:jrandomdeveloper@example.com tag:wontfix

If a key is used multiple times then the set of bugs selected includes those matching any of the supplied values; for example

bts limit package:foo severity:wishlist severity:minor

only applies the subsequent control commands to bugs of package foo with either **wishlist** or **minor** severity.

owner *bug owner-email*
Change the ‘‘owner’’ address of a *bug*, with **!** meaning ‘use the address on the current email as the new owner address’.

The owner of a bug accepts responsibility for dealing with it.

noowner *bug*
Mark a bug as having no ‘‘owner’’.

subscribe *bug* [*email*]
Subscribe the given *email* address to the specified *bug* report. If no email address is specified, the environment variable **DEBEMAIL** or **EMAIL** (in that order) is used. If those are not set, or **!** is given as email address, your default address will be used.

After executing this command, you will be sent a subscription confirmation to which you have to reply. When subscribed to a bug report, you receive all relevant emails and notifications. Use the unsubscribe command to unsubscribe.

unsubscribe *bug* [*email*]
Unsubscribe the given email address from the specified bug report. As with subscribe above, if no email address is specified, the environment variables **DEBEMAIL** or **EMAIL** (in that order) is used. If those are not set, or **!** is given as email address, your default address will be used.

After executing this command, you will be sent an unsubscription confirmation to which you have to reply. Use the **subscribe** command to, well, subscribe.

reportspam *bug ...*
The **reportspam** command allows you to report a *bug* report as containing spam. It saves one from having to go to the bug web page to do so.

spamreport *bug ...*
spamreport is a synonym for **reportspam**.

cache [*options*] [*maint_email* | *pkg* | **src:***pkg* | **from:***submitter*]
cache [*options*] [**release-critical** | **release−critical/**... | **RC**]
Generate or update a cache of bug reports for the given email address or package. By default it downloads all bugs belonging to the email address in the **DEBEMAIL** environment variable (or the **EMAIL** environment variable if **DEBEMAIL** is unset). This command may be repeated to cache bugs belonging to several people or packages. If multiple packages or addresses are supplied, bugs belonging to any of the arguments will be cached; those belonging to more than one of the arguments will only be downloaded once. The cached bugs are stored in *$XDG_CACHE_HOME/devscripts/bts/* or, if **XDG_CACHE_HOME** is not set, in *~/.cache/devscripts/bts/*.

You can use the cached bugs with the **−o** switch. For example:

```
bts −o bugs
bts −o show 12345
```

Also, **bts** will update the files in it in a piecemeal fashion as it downloads information from the BTS using the **show** command. You might thus set up the cache, and update the whole thing once a week, while letting the automatic cache updates update the bugs you frequently refer to during the week.

Some options affect the behaviour of the **cache** command. The first is the setting of **−−cache−mode**, which controls how much **bts** downloads of the referenced links from the bug page, including boring bits such as the acknowledgement emails, emails to the control bot, and the mbox version of the bug report. It can take three values: **min** (the minimum), **mbox** (download the minimum plus the mbox version of the bug report) or **full** (the whole works). The second is **−−force−refresh** or **−f**, which forces the download, even if the cached bug report is up-to-date. The **−−include−resolved** option indicates whether bug reports marked as resolved should be downloaded during caching.

Each of these is configurable from the configuration file, as described below. They may also be specified after the **cache** command as well as at the start of the command line.

Finally, **−q** or **−−quiet** will suppress messages about caches being up-to-date, and giving the option twice will suppress all cache messages (except for error messages).

Beware of caching RC, though: it will take a LONG time! (With 1000+ RC bugs and a delay of 5 seconds between bugs, you're looking at a minimum of 1.5 hours, and probably significantly more than that.)

cleancache *package* | **src:***package* | *maintainer*
cleancache from:*submitter* | **tag:***tag* | **usertag:***tag* | *number* | **ALL**

Clean the cache for the specified *package*, *maintainer*, etc., as described above for the **bugs** command, or clean the entire cache if **ALL** is specified. This is useful if you are going to have permanent network access or if the database has become corrupted for some reason. Note that for safety, this command does not default to the value of **DEBEMAIL** or **EMAIL**.

listcachedbugs [*number*]

List cached bug ids (intended to support bash completion). The optional number argument restricts the list to those bug ids that start with that number.

version

Display version and copyright information.

help

Display a short summary of commands, suspiciously similar to parts of this man page.

ENVIRONMENT VARIABLES

DEBEMAIL

If this is set, the From: line in the email will be set to use this email address instead of your normal email address (as would be determined by **mail**).

DEBFULLNAME

If **DEBEMAIL** is set, **DEBFULLNAME** is examined to determine the full name to use; if this is not set, **bts** attempts to determine a name from your *passwd* entry.

BROWSER

If set, it specifies the browser to use for the **show** and **bugs** options. See the description above.

CONFIGURATION VARIABLES

The two configuration files */etc/devscripts.conf* and *~/.devscripts* are sourced by a shell in that order to set configuration variables. Command line options can be used to override configuration file settings. Environment variable settings are ignored for this purpose. The currently recognised variables are:

BTS_OFFLINE

If this is set to **yes**, then it is the same as the **−−offline** command line parameter being used. Only has an effect on the **show** and **bugs** commands. The default is **no**. See the description of the **show** command above for more information.

BTS_CACHE

If this is set to **no**, then it is the same as the **−−no−cache** command line parameter being used. Only has an effect on the **show** and **bug** commands. The default is **yes**. Again, see the **show** command above for more information.

BTS_CACHE_MODE={min,mbox,full}
How much of the BTS should we mirror when we are asked to cache something? Just the minimum, or also the mbox or the whole thing? The default is **min**, and it has the same meaning as the **−−cache−mode** command line parameter. Only has an effect on the cache. See the **cache** command for more information.

BTS_FORCE_REFRESH
If this is set to **yes**, then it is the same as the **−−force−refresh** command line parameter being used. Only has an effect on the **cache** command. The default is **no**. See the **cache** command for more information.

BTS_MAIL_READER
If this is set, specifies a mail reader to use instead of **mutt**. Same as the **−−mailreader** command line option.

BTS_SENDMAIL_COMMAND
If this is set, specifies a **sendmail** command to use instead of */usr/sbin/sendmail*. Same as the **−−sendmail** command line option.

BTS_ONLY_NEW
Download only new bugs when caching. Do not check for updates in bugs we already have. The default is **no**. Same as the **−−only−new** command line option.

BTS_SMTP_HOST
If this is set, specifies an SMTP host to use for sending mail rather than using the **sendmail** command. Same as the **−−smtp−host** command line option.

Note that this option takes priority over **BTS_SENDMAIL_COMMAND** if both are set, unless the **−−sendmail** option is used.

BTS_SMTP_AUTH_USERNAME, **BTS_SMTP_AUTH_PASSWORD**
If these options are set, then it is the same as the **−−smtp−username** and **−−smtp−password** options being used.

BTS_SMTP_HELO
Same as the **−−smtp−helo** command line option.

BTS_INCLUDE_RESOLVED
If this is set to **no**, then it is the same as the **−−no−include−resolved** command line parameter being used. Only has an effect on the **cache** command. The default is **yes**. See the **cache** command for more information.

BTS_SUPPRESS_ACKS
If this is set to **yes**, then it is the same as the **−−no−ack** command line parameter being used. The default is **no**.

BTS_INTERACTIVE
If this is set to **yes** or **force**, then it is the same as the **−−interactive** or **−−force−interactive** command line parameter being used. The default is **no**.

BTS_DEFAULT_CC
Specify a list of e−mail addresses to which a carbon copy of the generated e−mail to the control bot should automatically be sent.

BTS_SERVER
Specify the name of a debbugs server which should be used instead of https://bugs.debian.org.

SEE ALSO

Please see <https://www.debian.org/Bugs/server−control> for more details on how to control the BTS using emails and <https://www.debian.org/Bugs/> for more information about the BTS.

querybts (1), **reportbug** (1), **pts−subscribe** (1), **devscripts.conf** (5)

COPYRIGHT