NAME

git-send-email - Send a collection of patches as emails

SYNOPSIS

git send-email [<options>] (<file>|<directory>)...
git send-email [<options>] <format-patch-options>
git send-email --dump-aliases

DESCRIPTION

Takes the patches given on the command line and emails them out. Patches can be specified as files, directories (which will send all files in the directory), or directly as a revision list. In the last case, any format accepted by **git-format-patch**(1) can be passed to git send-email, as well as options understood by **git-format-patch**(1).

The header of the email is configurable via command-line options. If not specified on the command line, the user will be prompted with a ReadLine enabled interface to provide the necessary information.

There are two formats accepted for patch files:

1.
format files

This is what **git-format-patch**(1) generates. Most headers and MIME formatting are ignored.

2.
original format used by Greg Kroah-Hartman's *send_lots_of_email.pl* script

This format expects the first line of the file to contain the "Cc:" value and the "Subject:" of the message as the second line.

OPTIONS

Composing

--annotate

Review and edit each patch you're about to send. Default is the value of **sendemail.annotate**. See the CONFIGURATION section for **sendemail.multiEdit**.

--bcc=<address>,...

Specify a "Bcc:" value for each email. Default is the value of **sendemail.bcc**.

This option may be specified multiple times.

--cc=<address>,...

Specify a starting "Cc:" value for each email. Default is the value of **sendemail.cc**.

This option may be specified multiple times.

--compose

Invoke a text editor (see GIT_EDITOR in **git-var**(1)) to edit an introductory message for the patch series.

When **--compose** is used, git send-email will use the From, To, Cc, Bcc, Subject, Reply-To, and In-Reply-To headers specified in the message. If the body of the message (what you type after the headers and a blank line) only contains blank (or Git: prefixed) lines, the summary won't be sent, but the headers mentioned above will be used unless they are removed.

Missing From or In-Reply-To headers will be prompted for.

See the CONFIGURATION section for **sendemail.multiEdit**.

--from=<address>

Specify the sender of the emails. If not specified on the command line, the value of the **sendemail.from** configuration option is used. If neither the command-line option nor **sendemail.from** are set, then the user will be prompted for the value. The default for the prompt will be the value of GIT_AUTHOR_IDENT, or GIT_COMMITTER_IDENT if that is not set, as returned by "git var -l".

--reply-to=<address>

Specify the address where replies from recipients should go to. Use this if replies to messages should go to another address than what is specified with the --from parameter.

--in-reply-to=<identifier>

Make the first mail (or all the mails with **--no-thread**) appear as a reply to the given Message-ID, which avoids breaking threads to provide a new patch series. The second and subsequent emails will be sent as replies according to the **--[no-]chain-reply-to** setting.

So for example when **--thread** and **--no-chain-reply-to** are specified, the second and subsequent patches will be replies to the first one like in the illustration below where **[PATCH v2 0/3]** is in reply to **[PATCH 0/2]**:

```
[PATCH 0/2] Here is what I did...
  [PATCH 1/2] Clean up and tests
  [PATCH 2/2] Implementation
  [PATCH v2 0/3] Here is a reroll
    [PATCH v2 1/3] Clean up
    [PATCH v2 2/3] New tests
    [PATCH v2 3/3] Implementation
```

Only necessary if --compose is also set. If --compose is not set, this will be prompted for.

--subject=<string>

Specify the initial subject of the email thread. Only necessary if --compose is also set. If --compose is not set, this will be prompted for.

--to=<address>,...

Specify the primary recipient of the emails generated. Generally, this will be the upstream maintainer of the project involved. Default is the value of the **sendemail.to** configuration value; if that is unspecified, and --to-cmd is not specified, this will be prompted for.

This option may be specified multiple times.

--8bit-encoding=<encoding>

When encountering a non-ASCII message or subject that does not declare its encoding, add headers/quoting to indicate it is encoded in <encoding>. Default is the value of the *sendemail.assume8bitEncoding*; if that is unspecified, this will be prompted for if any non-ASCII files are encountered.

Note that no attempts whatsoever are made to validate the encoding.

--compose-encoding=<encoding>

Specify encoding of compose message. Default is the value of the *sendemail.composeEncoding*; if that is unspecified, UTF-8 is assumed.

--transfer-encoding=(7bit|8bit|quoted-printable|base64|auto)

Specify the transfer encoding to be used to send the message over SMTP. 7bit will fail upon encountering a non-ASCII message. quoted-printable can be useful when the repository contains files that contain carriage returns, but makes the raw patch email file (as saved from a MUA) much harder to inspect manually. base64 is even more fool proof, but also even more opaque. auto will use 8bit when possible, and quoted-printable otherwise.

Default is the value of the **sendemail.transferEncoding** configuration value; if that is unspecified, default to **auto**.

--xmailer, --no-xmailer

Add (or prevent adding) the "X-Mailer:" header. By default, the header is added, but it can be turned off by setting the **sendemail.xmailer** configuration variable to **false**.

Sending

--envelope-sender=<address>

Specify the envelope sender used to send the emails. This is useful if your default address is not the address that is subscribed to a list. In order to use the *From* address, set the value to "auto". If you use the sendmail binary, you must have suitable privileges for the -f parameter. Default is the value of the **sendemail.envelopeSender** configuration variable; if that is unspecified, choosing the envelope sender is left to your MTA.

--sendmail-cmd=<command>

Specify a command to run to send the email. The command should be sendmail-like; specifically, it must support the **-i** option. The command will be executed in the shell if necessary. Default is the value of **sendemail.sendmailCmd**. If unspecified, and if --smtp-server is also unspecified, git-send-email will search for **sendmail** in **/usr/sbin**, **/usr/lib** and $PATH.

--smtp-encryption=<encryption>

Specify in what way encrypting begins for the SMTP connection. Valid values are *ssl* and *tls*. Any other value reverts to plain (unencrypted) SMTP, which defaults to port 25. Despite the names, both values will use the same newer version of TLS, but for historic reasons have these names. *ssl* refers to "implicit" encryption (sometimes called SMTPS), that uses port 465 by default. *tls* refers to "explicit" encryption (often known as STARTTLS), that uses port 25 by default. Other ports might be used by the SMTP server, which are not the default. Commonly found alternative port for *tls* and unencrypted is 587. You need to check your provider’s documentation or your server configuration to make sure for your own case. Default is the value of **sendemail.smtpEncryption**.

--smtp-domain=<FQDN>

Specifies the Fully Qualified Domain Name (FQDN) used in the HELO/EHLO command to the SMTP server. Some servers require the FQDN to match your IP address. If not set, git send-email attempts to determine your FQDN automatically. Default is the value of **sendemail.smtpDomain**.

--smtp-auth=<mechanisms>

Whitespace-separated list of allowed SMTP-AUTH mechanisms. This setting forces using only the listed mechanisms. Example:

```
        $ git send-email --smtp-auth="PLAIN LOGIN GSSAPI" ...
```

If at least one of the specified mechanisms matches the ones advertised by the SMTP server and if it is supported by the utilized SASL library, the mechanism is used for authentication. If neither *sendemail.smtpAuth* nor **--smtp-auth** is specified, all mechanisms supported by the SASL library can be used. The special value *none* maybe specified to completely disable authentication independently of **--smtp-user**

--smtp-pass[=<password>]

Password for SMTP-AUTH. The argument is optional: If no argument is specified, then the empty string is used as the password. Default is the value of **sendemail.smtpPass**, however **--smtp-pass** always overrides this value.

Furthermore, passwords need not be specified in configuration files or on the command line. If a username has been specified (with **--smtp-user** or a **sendemail.smtpUser**), but no password has been specified (with **--smtp-pass** or **sendemail.smtpPass**), then a password is obtained using *git-credential*.

--no-smtp-auth

Disable SMTP authentication. Short hand for **--smtp-auth=none**

--smtp-server=<host>

If set, specifies the outgoing SMTP server to use (e.g. **smtp.example.com** or a raw IP address). If unspecified, and if **--sendmail-cmd** is also unspecified, the default is to search for **sendmail** in **/usr/sbin**, **/usr/lib** and $PATH if such a program is available, falling back to **localhost** otherwise.

For backward compatibility, this option can also specify a full pathname of a sendmail-like program instead; the program must support the **-i** option. This method does not support passing arguments or using plain command names. For those use cases, consider using **--sendmail-cmd** instead.

--smtp-server-port=<port>

Specifies a port different from the default port (SMTP servers typically listen to smtp port 25, but may also listen to submission port 587, or the common SSL smtp port 465); symbolic port names (e.g. "submission" instead of 587) are also accepted. The port can also be set with the **sendemail.smtpServerPort** configuration variable.

--smtp-server-option=<option>

If set, specifies the outgoing SMTP server option to use. Default value can be specified by the **sendemail.smtpServerOption** configuration option.

The --smtp-server-option option must be repeated for each option you want to pass to the server. Likewise, different lines in the configuration files must be used for each option.

--smtp-ssl

Legacy alias for *--smtp-encryption ssl*.

--smtp-ssl-cert-path

Path to a store of trusted CA certificates for SMTP SSL/TLS certificate validation (either a directory that has been processed by *c_rehash*, or a single file containing one or more PEM format certificates concatenated together: see verify(1) -CAfile and -CApath for more information on these). Set it to an empty string to disable certificate verification. Defaults to the value of the **sendemail.smtpSSLCertPath** configuration variable, if set, or the backing SSL library's compiled-in default otherwise (which should be the best choice on most platforms).

--smtp-user=<user>

Username for SMTP-AUTH. Default is the value of **sendemail.smtpUser**; if a username is not specified (with **--smtp-user** or **sendemail.smtpUser**), then authentication is not attempted.

--smtp-debug=(0|1)

Enable (1) or disable (0) debug output. If enabled, SMTP commands and replies will be printed. Useful to debug TLS connection and authentication problems.

--batch-size=<num>

Some email servers (e.g. smtp.163.com) limit the number emails to be sent per session (connection) and this will lead to a failure when sending many messages. With this option, send-email will disconnect after sending $<num> messages and wait for a few seconds (see --relogin-delay) and reconnect, to work around such a limit. You may want to use some form of credential helper to avoid having to retype your password every time this happens. Defaults to the **sendemail.smtpBatchSize** configuration variable.

--relogin-delay=<int>

Waiting $<int> seconds before reconnecting to SMTP server. Used together with --batch-size option. Defaults to the **sendemail.smtpReloginDelay** configuration variable.

Automating

--no-to, --no-cc, --no-bcc

Clears any list of "To:", "Cc:", "Bcc:" addresses previously set via config.

--no-identity

Clears the previously read value of **sendemail.identity** set via config, if any.

--to-cmd=<command>

Specify a command to execute once per patch file which should generate patch file specific "To:" entries. Output of this command must be single email address per line. Default is the value of *sendemail.toCmd* configuration value.

--cc-cmd=<command>

Specify a command to execute once per patch file which should generate patch file specific "Cc:" entries. Output of this command must be single email address per line. Default is the value of **sendemail.ccCmd** configuration value.

--header-cmd=<command>

Specify a command that is executed once per outgoing message and output RFC 2822 style header lines to be inserted into them. When the **sendemail.headerCmd** configuration variable is set, its value is always used. When --header-cmd is provided at the command line, its value takes precedence over the **sendemail.headerCmd** configuration variable.

--no-header-cmd

Disable any header command in use.

--[no-]chain-reply-to

If this is set, each email will be sent as a reply to the previous email sent. If disabled with "--no-chain-reply-to", all emails after the first will be sent as replies to the first email sent. When using this, it is recommended that the first file given be an overview of the entire patch series. Disabled by default, but the **sendemail.chainReplyTo** configuration variable can be used to enable it.

--identity=<identity>

A configuration identity. When given, causes values in the *sendemail.<identity>* subsection to take precedence over values in the *sendemail* section. The default identity is the value of **sendemail.identity**.

--[no-]signed-off-by-cc

If this is set, add emails found in the **Signed-off-by** trailer or Cc: lines to the cc list. Default is the value of **sendemail.signedOffByCc** configuration value; if that is unspecified, default to --signed-off-by-cc.

--[no-]cc-cover

If this is set, emails found in Cc: headers in the first patch of the series (typically the cover letter) are added to the cc list for each email set. Default is the value of *sendemail.ccCover* configuration value; if that is unspecified, default to --no-cc-cover.

--[no-]to-cover

If this is set, emails found in To: headers in the first patch of the series (typically the cover letter) are added to the to list for each email set. Default is the value of *sendemail.toCover* configuration value; if that is unspecified, default to --no-to-cover.

--suppress-cc=<category>

Specify an additional category of recipients to suppress the auto-cc of:

ϕ

will avoid including the patch author.

ϕ

will avoid including the sender.

ϕ

will avoid including anyone mentioned in Cc lines in the patch header except for self (use *self* for that).

ϕ

will avoid including anyone mentioned in Cc lines in the patch body (commit message) except for self (use *self* for that).

ϕ

will avoid including anyone mentioned in the Signed-off-by trailers except for self (use *self* for that).

ϕ

will avoid including anyone mentioned in Acked-by, Reviewed-by, Tested-by and other "-by" lines in the patch body, except Signed-off-by (use *sob* for that).

ϕ

will avoid running the --cc-cmd.

ϕ

is equivalent to *sob* + *bodycc* + *misc-by*.

ϕ

will suppress all auto cc values.

Default is the value of **sendemail.suppressCc** configuration value; if that is unspecified, default to *self* if --suppress-from is specified, as well as *body* if --no-signed-off-cc is specified.

--[no-]suppress-from

If this is set, do not add the From: address to the cc: list. Default is the value of **sendemail.suppressFrom** configuration value; if that is unspecified, default to --no-suppress-from.

--[no-]thread

If this is set, the In-Reply-To and References headers will be added to each email sent. Whether each mail refers to the previous email (**deep** threading per *git format-patch* wording) or to the first email (**shallow** threading) is governed by "--[no-]chain-reply-to".

If disabled with "--no-thread", those headers will not be added (unless specified with --in-reply-to). Default is the value of the **sendemail.thread** configuration value; if that is unspecified, default to --thread.

It is up to the user to ensure that no In-Reply-To header already exists when *git send-email* is asked to add it (especially note that *git format-patch* can be configured to do the threading itself). Failure to do so may not produce the expected result in the recipient's MUA.

Administering

--confirm=<mode>

Confirm just before sending:

ϕ

will always confirm before sending

ϕ

will never confirm before sending

ϕ

will confirm before sending when send-email has automatically added addresses from the patch to the Cc list

ϕ

will confirm before sending the first message when using --compose.

ϕ

is equivalent to *cc* + *compose*

Default is the value of **sendemail.confirm** configuration value; if that is unspecified, default to *auto* unless any of the suppress options have been specified, in which case default to *compose*.

--dry-run

Do everything except actually send the emails.

--[no-]format-patch

When an argument may be understood either as a reference or as a file name, choose to understand it as a format-patch argument (**--format-patch**) or as a file name (**--no-format-patch**). By default, when such a conflict occurs, git send-email will fail.

--quiet

Make git-send-email less verbose. One line per email should be all that is output.

--[no-]validate

Perform sanity checks on patches. Currently, validation means the following:

- the sendemail-validate hook if present (see **githooks**(5)).

- of patches that contain lines longer than 998 characters unless a suitable transfer encoding (*auto*, *base64*, or *quoted-printable*) is used; this is due to SMTP limits as described by **https://www.ietf.org/rfc/rfc5322.txt**.

Default is the value of **sendemail.validate**; if this is not set, default to **--validate**.

--force

Send emails even if safety checks would prevent it.

Information

--dump-aliases

Instead of the normal operation, dump the shorthand alias names from the configured alias file(s), one per line in alphabetical order. Note that this only includes the alias name and not its expanded email addresses. See *sendemail.aliasesFile* for more information about aliases.

CONFIGURATION

Everything below this line in this section is selectively included from the **git-config**(1) documentation. The content is the same as what’s found there:

sendemail.identity

A configuration identity. When given, causes values in the *sendemail.<identity>* subsection to take precedence over values in the *sendemail* section. The default identity is the value of

sendemail.identity.

sendemail.smtpEncryption

See **git-send-email**(1) for description. Note that this setting is not subject to the *identity* mechanism.

sendemail.smtpSSLCertPath

Path to ca-certificates (either a directory or a single file). Set it to an empty string to disable certificate verification.

sendemail.<identity>.*

Identity-specific versions of the *sendemail.** parameters found below, taking precedence over those when this identity is selected, through either the command-line or **sendemail.identity**.

sendemail.multiEdit

If true (default), a single editor instance will be spawned to edit files you have to edit (patches when **--annotate** is used, and the summary when **--compose** is used). If false, files will be edited one after the other, spawning a new editor each time.

sendemail.confirm

Sets the default for whether to confirm before sending. Must be one of *always*, *never*, *cc*, *compose*, or *auto*. See **--confirm** in the **git-send-email**(1) documentation for the meaning of these values.

sendemail.aliasesFile

To avoid typing long email addresses, point this to one or more email aliases files. You must also supply **sendemail.aliasFileType**.

sendemail.aliasFileType

Format of the file(s) specified in sendemail.aliasesFile. Must be one of *mutt*, *mailrc*, *pine*, *elm*, *gnus*, or *sendmail*.

What an alias file in each format looks like can be found in the documentation of the email program of the same name. The differences and limitations from the standard formats are described below:

sendmail

- aliases and quoted addresses are not supported: lines that contain a **"** symbol are ignored.

- to a file (**/path/name**) or pipe (**|command**) is not supported.

- inclusion (**:include: /path/name**) is not supported.

- are printed on the standard error output for any explicitly unsupported constructs, and any other lines that are not recognized by the parser.

sendemail.annotate, sendemail.bcc, sendemail.cc, sendemail.ccCmd, sendemail.chainReplyTo, sendemail.envelopeSender, sendemail.from, sendemail.headerCmd, sendemail.signedOffByCc, sendemail.smtpPass, sendemail.suppressCc, sendemail.suppressFrom, sendemail.to, sendemail.toCmd, sendemail.smtpDomain, sendemail.smtpServer, sendemail.smtpServerPort, sendemail.smtpServerOption, sendemail.smtpUser, sendemail.thread, sendemail.transferEncoding, sendemail.validate, sendemail.xmailer

These configuration variables all provide a default for **git-send-email**(1) command-line options. See its documentation for details.

sendemail.signedOffCc (deprecated)

Deprecated alias for **sendemail.signedOffByCc**.

sendemail.smtpBatchSize

Number of messages to be sent per connection, after that a relogin will happen. If the value is 0 or undefined, send all messages in one connection. See also the **--batch-size** option of **git-send-email**(1).

sendemail.smtpReloginDelay

Seconds to wait before reconnecting to the smtp server. See also the **--relogin-delay** option of **git-send-email**(1).

sendemail.forbidSendmailVariables

To avoid common misconfiguration mistakes, **git-send-email**(1) will abort with a warning if any configuration options for "sendmail" exist. Set this variable to bypass the check.

EXAMPLES

Use gmail as the smtp server

To use *git send-email* to send your patches through the GMail SMTP server, edit ~/.gitconfig to specify your account settings:

```
[sendemail]
        smtpEncryption = tls
        smtpServer = smtp.gmail.com
        smtpUser = yourname@gmail.com
        smtpServerPort = 587
```

If you have multi-factor authentication set up on your Gmail account, you can generate an app-specific password for use with *git send-email*. Visit **https://security.google.com/settings/security/apppasswords** to create it.

Once your commits are ready to be sent to the mailing list, run the following commands:

```
$ git format-patch --cover-letter -M origin/master -o outgoing/
$ edit outgoing/0000-*
$ git send-email outgoing/*
```

The first time you run it, you will be prompted for your credentials. Enter the app-specific or your regular password as appropriate. If you have credential helper configured (see **git-credential**(1)), the password will be saved in the credential store so you won’t have to type it the next time.

Note: the following core Perl modules that may be installed with your distribution of Perl are required: MIME::Base64, MIME::QuotedPrint, Net::Domain and Net::SMTP. These additional Perl modules are also required: Authen::SASL and Mail::Address.

SEE ALSO

git-format-patch(1), **git-imap-send**(1), mbox(5)

GIT

Part of the **git**(1) suite