

NAME

mailatt – send e-mail with attachments

SYNOPSIS

```
mailatt [ -r recipients ] [ -c cc-addresses ] [ -b bcc-addresses ] [ -f from ]
[ -R reply-to-addresses ] [ -s subject ] [ -H headerline ] [ -8Iadimqu ] [ -C charset ]
[ file1 .. fileN ]

mailatt { -h | -v }
```

DESCRIPTION

mailatt is used to send multipart MIME e-mail with file attachments of any format.

- Files of type *text/** in US-ASCII will be attached as-is (7bit-encoded).
- Plaintext files are supposed to be clean 7-bit ASCII unless another character set has been specified on the commandline. See the **-C** option below.
- Files of type *text/** in a character set other than US-ASCII will be attached using quoted-printable encoding.
- Binary files will be attached base64-encoded by default (but see the **-q** and **-u** options below).
- If explicitly requested, yEncoded files (**.ync*) will be attached as-is (8bit-encoded). Note that not all Mail Transfer Agents can handle this type of transfer cleanly. See the **-8** option below.

mailatt can send messages in two possible formats. *multipart/mixed* is used to send a collection of files which are not necessarily related. *multipart/alternative* is used to provide alternative representations of the same message (*e.g.* plain text and HTML), in which case the mail client can select one to present. See also the **-a** option below.

This manual pertains to **mailatt** version 1.13.1.

- 8 Include certain MIME types (currently only yEncoded files) using 8bit encoding. The default is to include all binary files using base64 encoding. The reason for allowing this is that yEncoded files are supposed to use less bandwidth. This method is, however, not supported by all Mail Transfer Agents.

-C charset

This option may be used multiple times, and in any position on the commandline. It specifies the character set to be used for all subsequent attachments and header lines. If unspecified, US-ASCII is used. See the EXAMPLES below.

-H headerline

Specifies a line to add to the mail header. The option can be used multiple times. Do not end the line with a newline character.

- I This option is nearly identical to **-i**, but uses the old behavior, which is to include a filename for all MIME parts, even those that are attached using `Content-Disposition: inline`.

Some e-mail clients may not display MIME parts correctly when this option is used. It is provided only for backward-compatibility.

Do not use. Use **-i** instead.

-R addr1[,addr2..]

Specify Reply-To: addresses for the e-mail. See **-r** for allowed formats.

- a Use *multipart/alternative* for the mail message instead of the default *multipart/mixed*. All message parts will automatically be included with `Content-Disposition: inline`, regardless of any **-i** option. Note that the simplest representation of the mail (probably *text/plain*) should be attached first.

-b addr1[,addr2..]

Specify Bcc: addresses for the e-mail. See **-r** for allowed formats.

-c *addr1[,addr2..]*

Specify Cc: addresses for the e-mail. See **-r** for allowed formats.

-d For debugging purposes only. The composed mail will be printed on *stdout* instead of actually being sent. This option takes precedence over the MAILATT_DEBUG variable (see below).

-f *addr1*

Specify From: address for the e-mail.

-h Print help (usage information) and exit.

-i Specifies that the first file on the commandline must be forcibly included with a Content-Disposition: inline header, and without a filename. If this option is not specified, all parts of the message will be included with Content-Disposition: attachment and a specified filename.

The option **-a** takes precedence over **-i**.

See also the **-I** option.

-m This option may be used multiple times, and in any position on the commandline. It indicates that base64 encoding should be used for all subsequent binary attachments and header lines. This is the default if unspecified.

-q This option may be used multiple times, and in any position on the commandline. It indicates that quoted-printable encoding should be used for all binary attachments and header lines. Although quoted-printable is normally only used for ASCII attachments, **mailatt**'s implementation will make binaries come through intact.

-r *addr1[,addr2..]*

Specify recipients (To: addresses) for the e-mail.

Each one of the addresses may be specified in one of the following formats:

```

user
<user>
<user@domain>
name <user>
name <user@domain>

```

The first two forms will be expanded (using the system *passwd* file or (if configured) the NIS *passwd* map) to the form "name <user>", which the mailer system may further expand to "name <user@domain>".

-s *subject*

Specify subject for the e-mail.

-u This option may be used multiple times, and in any position on the commandline. It indicates that uuencode should be used for all binary attachments. Warning: uuencode is reportedly not 100 percent portable.

-v Display version information and exit.

file1 [*file2..*]

Indicate which files should be attached to the mail. Depending on the extension (*i.e.* the filename part following a . in the filename), a corresponding MIME type will be selected from a built-in list.

If the file type is not known or the extension is empty, the file will be sent as type *application/octet-stream*, unless the **-a** or **-i** option requires it to be included inline. In that case, the MIME type will forcibly be changed to *text/plain*.

At most one of the filenames may be specified as a single - (minus), which will cause **mailatt** to read a document from stdin. If entered interactively, such a document should be ended with the eof-character (usually CTRL-D). **mailatt** will always assume that this document is of type *text/plain*.

EXAMPLES

Send all *.eps files to two recipients:

```
mailatt -s 'Filesystem usage' -r alexander,maxima fsgraph-*.eps
```

Send a file fragment, carbon copy to the sysadmin (possibly forwarded by a *forward* file):

```
mailatt -s 'Fragment 2/10' -r beatrix -c root dumps.tar.gz.xab
```

Send a formatted file, preceded by an introductory message, typed interactively:

```
mailatt -s 'Sendmail manual' -r sysadmin@domain.nl -i - sendmail.pdf
Hello Bernhard,
Here is the sendmail manual I promised.
^D
```

Send a formatted mail in HTML format, specifying a From: address:

```
mailatt -f 'Pieter <pvvollenhoven@hotmail.com>' -r margriet
-i birthdaycard.html
```

Send a mail in both plain text and HTML format:

```
mailatt -a -r constantijn,friso -s greeting.txt greeting.html
```

Attach a file of unknown MIME type using quoted-printable encoding (*e.g.* to increase readability). This overrides the default encoding (base64) that is used for attachments of unknown MIME type coming *before* the **-q** option.

```
mailatt -r christina -s "config files" -i body.txt
/etc/sendmail.cf -q /etc/hosts
```

Send a mail using the Cyrillic character set to multiple addresses:

```
mailatt -r 'Mr. Putin <putin@kremlin.ru>,root'
-s 'Important letter' -i -C ISO-8859-5 letter.txt
```

Add custom lines to the mail header:

```
mailatt -r 'Benedictus_XVI@vatican.va' -s 'Oldest translation'
-i -H "X-Year-Finished: 1637" message.txt SV.pdf
```

Send a mail with mixed character sets:

```
mailatt -C ISO-8859-15 -r 'François <francois@nimporte.ou>'
-s 'Allô' -C UTF-8 -i message.html -C windows-1253 m.txt
```

MAILATT_DEBUG

For debugging purposes only. If set and not **0**, causes the composed mail to be printed on *stdout* instead of actually being sent.

BUGS and WARNINGS

An HTML-file cannot be included using input redirection, as its type will be reported in the mail as *text/plain*. Attach it by filename instead.

VERSION

This manual pertains to **mailatt** version 1.13.1.

SEE ALSO

mail (1), *mailx* (1), *mmencode* (1), *sendmail* (8), *uuencode* (1).

RFC 822: Format of Internet Text Messages

RFC 1049: The Content-Type Header Field for Internet Messages

RFC 2045: MIME: Format of Internet Message Bodies

RFC 2046: MIME: Media Types

RFC 2047: MIME: Message Header Extensions for Non-ASCII Text

RFC 2183: The Content-Disposition Header Field

RFC 2387: The multipart/related Content-Type

RFC 2392: Content-ID and Message-ID URLs

RFC 2822: Internet Message Format

AUTHOR

Copyright © 2019, René Uittenbogaard (ruittenb@users.sourceforge.net)