

**NAME**

**bintrans**, **uuencode**, **uudecode**, **b64encode**, **b64decode** - encode/decode a binary file

**SYNOPSIS**

**bintrans** [algorithm] [...]  
**uuencode** [-m] [-r] [-o *output\_file*] [*file*] *name*  
**uudecode** [-cimprs] [*file* ...]  
**uudecode** [-i] -o *output\_file*  
**b64encode** [-r] [-w *column*] [-o *output\_file*] [*file*] *name*  
**b64decode** [-cimprs] [*file* ...]  
**b64decode** [-i] -o *output\_file* [*file*]  
**base64** [-d] [-w *column*] [*file*]

**DESCRIPTION**

The **uuencode** and **uudecode** utilities are used to transmit binary files over transmission mediums that do not support other than simple ASCII data. The **b64encode** utility is synonymous with **uuencode** with the **-m** flag specified. The **b64decode** utility is synonymous with **uudecode** with the **-m** flag specified.

The **base64** utility acts as a base64 decoder when passed the **--decode** (or **-d**) flag and as a base64 encoder otherwise. As a decoder it only accepts raw base64 input and as an encoder it does not produce the framing lines. **base64** reads standard input or *file* if it is provided and writes to standard output. Options **--wrap** (or **-w**) and **--ignore-garbage** (or **-i**) are accepted for compatibility with GNU base64, but the latter is unimplemented and silently ignored.

The **uuencode** utility reads *file* (or by default the standard input) and writes an encoded version to the standard output, or *output\_file* if one has been specified. The encoding uses only printing ASCII characters and includes the mode of the file and the operand *name* for use by **uudecode**.

The **uudecode** utility transforms *uuencoded* files (or by default, the standard input) into the original form. The resulting file is named either *name* or (depending on options passed to **uudecode**) *output\_file* and will have the mode of the original file except that setuid and execute bits are not retained. The **uudecode** utility ignores any leading and trailing lines.

The following options are available for **uuencode**:

- m**    Use the Base64 method of encoding, rather than the traditional **uuencode** algorithm.
- r**    Produce raw output by excluding the initial and final framing lines.
- o *output\_file***

Output to *output\_file* instead of standard output.

The following options are available for **uudecode**:

- c** Decode more than one uuencoded file from *file* if possible.
- i** Do not overwrite files.
- m** When used with the **-r** flag, decode Base64 input instead of traditional **uuencode** input. Without **-r** it has no effect.
- o output\_file**  
Output to *output\_file* instead of any pathname contained in the input data.
- p** Decode *file* and write output to standard output.
- r** Decode raw (or broken) input, which is missing the initial and possibly the final framing lines. The input is assumed to be in the traditional **uuencode** encoding, but if the **-m** flag is used, or if the utility is invoked as **b64decode**, then the input is assumed to be in Base64 format.
- s** Do not strip output pathname to base filename. By default **uudecode** deletes any prefix ending with the last slash '/' for security reasons.

Additionally, **b64encode** accepts the following option:

- w column**  
Wrap encoded output after *column*.

**bintrans** is a generic utility that can run any of the aforementioned encoders and decoders. It can also run algorithms that are not available through a dedicated program:

**qp** is a quoted-printable converter and accepts the following options:

- u** Decode.
- o output\_file**  
Output to *output\_file* instead of standard output.

## EXAMPLES

The following example packages up a source tree, compresses it, uuencodes it and mails it to a user on

another system. When **uudecode** is run on the target system, the file “src\_tree.tar.Z” will be created which may then be uncompressed and extracted into the original tree.

```
tar cf - src_tree | compress |  
uencode src_tree.tar.Z | mail user@example.com
```

The following example unpacks all uuencoded files from your mailbox into your current working directory.

```
uudecode -c < $MAIL
```

The following example extracts a compressed tar archive from your mailbox

```
uudecode -o /dev/stdout < $MAIL | zcat | tar xfv -
```

## SEE ALSO

basename(1), compress(1), mail(1), uucp(1) (*ports/net/freebsd-uucp*), uuencode(5)

## HISTORY

The **uudecode** and **uencode** utilities appeared in 4.0BSD.

## BUGS

Files encoded using the traditional algorithm are expanded by 35% (3 bytes become 4 plus control information).