NAME

mktemp - make temporary file name (unique)

SYNOPSIS

mktemp [**-d**] [**-p** *tmpdir*] [**-q**] [**-t** *prefix*] [**-u**] *template ...*
mktemp [**-d**] [**-p** *tmpdir*] [**-q**] [**-u**] **-t** *prefix*

DESCRIPTION

The **mktemp** utility takes each of the given file name templates and overwrites a portion of it to create a file name. This file name is unique and suitable for use by the application. The template may be any file name with some number of 'Xs' appended to it, for example */tmp/temp.XXXX*. The trailing 'Xs' are replaced with the current process number and/or a unique letter combination. The number of unique file names **mktemp** can return depends on the number of 'Xs' provided; six 'Xs' will result in **mktemp** selecting 1 of 56800235584 (62 ** 6) possible file names.

If **mktemp** can successfully generate a unique file name, the file is created with mode 0600 (unless the **-u** flag is given) and the filename is printed to standard output.

If the **-t** *prefix* option is given, **mktemp** will generate a template string based on the *prefix* and the TMPDIR environment variable if set. If the **-p** option is set, then the given *tmpdir* will be used if the TMPDIR environment variable is not set. Finally, */tmp* will be used if neither TMPDIR or **-p** are set and used. Care should be taken to ensure that it is appropriate to use an environment variable potentially supplied by the user.

If no arguments are passed or if only the **-d** flag is passed **mktemp** behaves as if **-t** tmp was supplied.

Any number of temporary files may be created in a single invocation, including one based on the internal template resulting from the **-t** flag.

The **mktemp** utility is provided to allow shell scripts to safely use temporary files. Traditionally, many shell scripts take the name of the program with the pid as a suffix and use that as a temporary file name. This kind of naming scheme is predictable and the race condition it creates is easy for an attacker to win. A safer, though still inferior, approach is to make a temporary directory using the same naming scheme. While this does allow one to guarantee that a temporary file will not be subverted, it still allows a simple denial of service attack. For these reasons it is suggested that **mktemp** be used instead.

OPTIONS

The available options are as follows:

-d, **--directory**

Make a directory instead of a file.

-p *tmpdir*, **--tmpdir**[*=tmpdir*]

Use *tmpdir* for the **-t** flag if the TMPDIR environment variable is not set. Additionally, any provided *template* arguments will be interpreted relative to the path specified as *tmpdir*. If *tmpdir* is either empty or omitted, then the TMPDIR environment variable will be used.

-q, **--quiet**

Fail silently if an error occurs. This is useful if a script does not want error output to go to standard error.

-t *prefix*

Generate a template (using the supplied *prefix* and TMPDIR if set) to create a filename template.

-u, **--dry-run**

Operate in "unsafe" mode. The temp file will be unlinked before **mktemp** exits. This is slightly better than mktemp(3) but still introduces a race condition. Use of this option is not encouraged.

EXIT STATUS

The **mktemp** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

The following sh(1) fragment illustrates a simple use of **mktemp** where the script should quit if it cannot get a safe temporary file.

```
tempfoo=`basename $0`
TMPFILE=`mktemp /tmp/${tempfoo}.XXXXXX` || exit 1
echo "program output" >> $TMPFILE
```

To allow the use of $TMPDIR:

```
tempfoo=`basename $0`
TMPFILE=`mktemp -t ${tempfoo}` || exit 1
echo "program output" >> $TMPFILE
```

In this case, we want the script to catch the error itself.

```
tempfoo=`basename $0`
TMPFILE=`mktemp -q /tmp/${tempfoo}.XXXXXX`
if [ $? -ne 0 ]; then
```

```
            echo "$0: Can't create temp file, exiting..."
            exit 1
    fi
```

SEE ALSO

　　mkdtemp(3), mkstemp(3), mktemp(3), environ(7)

HISTORY

　　A **mktemp** utility appeared in OpenBSD 2.1. This implementation was written independently based on the OpenBSD man page, and first appeared in FreeBSD 2.2.7. This man page is taken from OpenBSD.