



Focus on Linux

Linux / Unix Command: *unzip*

[Command Library](#)

NAME

unzip - list, test and extract compressed files in a ZIP archive

SYNOPSIS

```
unzip [-Z] [-cflptuvz[abjnoqsCLMVX$/:]] file[.zip] [file(s) ...] [-x xfile(s) ...] [-d exdir]
```

DESCRIPTION

unzip will list, test, or extract files from a ZIP archive, commonly found on MS-DOS systems. The default behavior (with no options) is to extract into the current directory (and subdirectories below it) all files from the specified ZIP archive. A companion program, *zip*(1L), creates ZIP archives; both programs are compatible with archives created by PKWARE's *PKZIP* and *PKUNZIP* for MS-DOS, but in many cases the program options or default behaviors differ.

ARGUMENTS

file[.zip]

Path of the ZIP archive(s). If the file specification is a wildcard, each matching file is processed in an order determined by the operating system (or file system). Only the filename can be a wildcard; the path itself cannot. Wildcard expressions are similar to those supported in commonly used Unix shells (*sh*, *ksh*, *bash*) and may contain:

*

matches a sequence of 0 or more characters

?

matches exactly 1 character

[...]

matches any single character found inside the brackets; ranges are specified by a beginning character, a hyphen, and an ending character. If an exclamation point or a caret (! or ^) follows the left bracket, then the range of characters within the brackets is complemented (that is, anything *except* the characters inside the brackets is considered a match).

(Be sure to quote any character that might otherwise be interpreted or modified by the operating system, particularly under Unix and VMS.) If no matches are found, the specification is assumed to be a literal filename; and if that also fails, the suffix .zip is appended. Note that self-extracting ZIP files are supported, as with any other ZIP archive; just specify the .exe suffix (if any) explicitly.

[file(s)]

An optional list of archive members to be processed, separated by spaces. (VMS versions compiled with VMSCLI defined must delimit files with commas instead. See **-v** in **OPTIONS** below.) Regular expressions (wildcards) may be used to match multiple members; see above. Again, be sure to quote expressions that would otherwise be expanded or modified by the operating system.

[-x xfile(s)]

An optional list of archive members to be excluded from processing. Since wildcard characters match directory separators (/), this option may be used to exclude any files that are in subdirectories. For example, ``unzip foo *.ch] -x */*" would extract all C source files in the main directory, but none in any subdirectories. Without the **-x** option, all C source files in all directories within the zipfile would be extracted.

[-d exdir]

An optional directory to which to extract files. By default, all files and subdirectories are recreated in the current directory; the **-d** option allows extraction in an arbitrary directory (always assuming one has permission to write to the directory). This option need not appear at the end of the command line; it is also accepted before the zipfile specification (with the normal options), immediately after the zipfile specification, or between the *file(s)* and the **-x** option. The option and directory may be concatenated without any white space between them, but note that this may cause normal shell behavior to be suppressed. In particular, ``-d ~" (tilde) is expanded by Unix C shells into the name of the user's home directory, but ``-d~" is treated as a literal subdirectory ``~" of the current directory.

OPTIONS

Note that, in order to support obsolescent hardware, *unzip*'s usage screen is limited to 22 or 23 lines and should therefore be considered only a reminder of the basic *unzip* syntax rather than an exhaustive list of all possible flags.

Most Popular

- [Linux Commands - Command Library - Man Pages](#)
- [find - Linux Command - Unix Command](#)
- [Linux Commands and Shell Commands Libraries](#)
- [Shell Commands Libraries - Linux Unix](#)
- [Linux/Unix/Computing Glossary](#)

Related Topics

- [PC Hardware / Reviews](#)
- [Focus on Windows](#)
- [Focus on Macs](#)
- [Focus on Unix](#)
- [Computer Certification](#)

The exhaustive list follows:

-Z

[zipinfo\(1L\)](#) mode. If the first option on the command line is **-Z**, the remaining options are taken to be [zipinfo\(1L\)](#) options. See the appropriate manual page for a description of these options.

-A

[OS/2, Unix DLL] print extended help for the DLL's programming interface (API).

-c

extract files to stdout/screen ("CRT"). This option is similar to the **-p** option except that the name of each file is printed as it is extracted, the **-a** option is allowed, and ASCII-EBCDIC conversion is automatically performed if appropriate. This option is not listed in the *unzip* usage screen.

-f

freshen existing files, i.e., extract only those files that already exist on disk and that are newer than the disk copies. By default *unzip* queries before overwriting, but the **-o** option may be used to suppress the queries. Note that under many operating systems, the TZ (timezone) environment variable must be set correctly in order for **-f** and **-u** to work properly (under Unix the variable is usually set automatically). The reasons for this are somewhat subtle but have to do with the differences between DOS-format file times (always local time) and Unix-format times (always in GMT/UTC) and the necessity to compare the two. A typical TZ value is "PST8PDT" (US Pacific time with automatic adjustment for Daylight Savings Time or "summer time").

-l

list archive files (short format). The names, uncompressed file sizes and modification dates and times of the specified files are printed, along with totals for all files specified. If UnZip was compiled with OS2_EAS defined, the **-l** option also lists columns for the sizes of stored OS/2 extended attributes (EAs) and OS/2 access control lists (ACLs). In addition, the zipfile comment and individual file comments (if any) are displayed. If a file was archived from a single-case file system (for example, the old MS-DOS FAT file system) and the **-L** option was given, the filename is converted to lowercase and is prefixed with a caret (^).

-p

extract files to pipe (stdout). Nothing but the file data is sent to stdout, and the files are always extracted in binary format, just as they are stored (no conversions).

-t

test archive files. This option extracts each specified file in memory and compares the CRC (cyclic redundancy check, an enhanced checksum) of the expanded file with the original file's stored CRC value.

-T

[most OSes] set the timestamp on the archive(s) to that of the newest file in each one. This corresponds to *zip*'s **-go** option except that it can be used on wildcard zipfiles (e.g., "unzip -T *.zip") and is much faster.

-u

update existing files and create new ones if needed. This option performs the same function as the **-f** option, extracting (with query) files that are newer than those with the same name on disk, and in addition it extracts those files that do not already exist on disk. See **-f** above for information on setting the timezone properly.

-v

be verbose or print diagnostic version info. This option has evolved and now behaves as both an option and a modifier. As an option it has two purposes: when a zipfile is specified with no other options, **-v** lists archive files verbosely, adding to the basic **-l** info the compression method, compressed size, compression ratio and 32-bit CRC. When no zipfile is specified (that is, the complete command is simply "unzip -v"), a diagnostic screen is printed. In addition to the normal header with release date and version, *unzip* lists the home Info-ZIP ftp site and where to find a list of other ftp and non-ftp sites; the target operating system for which it was compiled, as well as (possibly) the hardware on which it was compiled, the compiler and version used, and the compilation date; any special compilation options that might affect the program's operation (see also **DECRYPTION** below); and any options stored in environment variables that might do the same (see **ENVIRONMENT OPTIONS** below). As a modifier it works in conjunction with other options (e.g., **-t**) to produce more verbose or debugging output; this is not yet fully implemented but will be in future releases.

-z

display only the archive comment.

MODIFIERS

-a

convert text files. Ordinarily all files are extracted exactly as they are stored (as "binary" files). The **-a** option causes files identified by *zip* as text files (those with the 't' label in *zipinfo* listings, rather than 'b') to be automatically extracted as such, converting line endings, end-of-file characters and the character set itself as necessary. (For example, Unix files use line feeds (LFs) for end-of-line (EOL) and have no end-of-file (EOF) marker; Macintoshes use carriage returns (CRs) for EOLs; and most PC operating systems use CR+LF for EOLs and control-Z for EOF. In addition, IBM mainframes and the Michigan Terminal System use EBCDIC rather than the more common ASCII character set, and NT supports Unicode.) Note that *zip*'s identification of

text files is by no means perfect; some ``text" files may actually be binary and vice versa. *unzip* therefore prints ``[text]" or ``[binary]" as a visual check for each file it extracts when using the **-a** option. The **-aa** option forces all files to be extracted as text, regardless of the supposed file type.

-b

[general] treat all files as binary (no text conversions). This is a shortcut for **---a**.

-b

[Tandem] force the creation files with filecode type 180 ('C') when extracting Zip entries marked as "text". (On Tandem, **-a** is enabled by default, see above).

-b

[VMS] auto-convert binary files (see **-a** above) to fixed-length, 512-byte record format. Doubling the option (**-bb**) forces all files to be extracted in this format. When extracting to standard output (**-c** or **-p** option in effect), the default conversion of text record delimiters is disabled for binary (**-b**) resp. all (**-bb**) files.

-B

[Unix only, and only if compiled with UNIXBACKUP defined] save a backup copy of each overwritten file with a tilde appended (e.g., the old copy of ``foo" is renamed to ``foo~"). This is similar to the default behavior of [emacs\(1\)](#) in many locations.

-C

match filenames case-insensitively. *unzip*'s philosophy is ``you get what you ask for" (this is also responsible for the **-L/-U** change; see the relevant options below). Because some file systems are fully case-sensitive (notably those under the Unix operating system) and because both ZIP archives and *unzip* itself are portable across platforms, *unzip*'s default behavior is to match both wildcard and literal filenames case-sensitively. That is, specifying ``makefile" on the command line will *only* match ``makefile" in the archive, not ``Makefile" or ``MAKEFILE" (and similarly for wildcard specifications). Since this does not correspond to the behavior of many other operating/file systems (for example, OS/2 HPFS, which preserves mixed case but is not sensitive to it), the **-C** option may be used to force all filename matches to be case-insensitive. In the example above, all three files would then match ``makefile" (or ``make*", or similar). The **-C** option affects files in both the normal file list and the excluded-file list (xlist).

-E

[MacOS only] display contents of MacOS extra field during restore operation.

-F

[Acorn only] suppress removal of NFS filetype extension from stored filenames.

-F

[non-Acorn systems supporting long filenames with embedded commas, and only if compiled with ACORN_FTYPE_NFS defined] translate filetype information from ACORN RISC OS extra field blocks into a NFS filetype extension and append it to the names of the extracted files. (When the stored filename appears to already have an appended NFS filetype extension, it is replaced by the info from the extra field.)

-i

[MacOS only] ignore filenames stored in MacOS extra fields. Instead, the most compatible filename stored in the generic part of the entry's header is used.

-j

junk paths. The archive's directory structure is not recreated; all files are deposited in the extraction directory (by default, the current one).

-J

[BeOS only] junk file attributes. The file's BeOS file attributes are not restored, just the file's data.

-J

[MacOS only] ignore MacOS extra fields. All Macintosh specific info is skipped. Data-fork and resource-fork are restored as separate files.

-L

convert to lowercase any filename originating on an uppercase-only operating system or file system. (This was *unzip*'s default behavior in releases prior to 5.11; the new default behavior is identical to the old behavior with the **-U** option, which is now obsolete and will be removed in a future release.) Depending on the archiver, files archived under single-case file systems (VMS, old MS-DOS FAT, etc.) may be stored as all-uppercase names; this can be ugly or inconvenient when extracting to a case-preserving file system such as OS/2 HPFS or a case-sensitive one such as under Unix. By default *unzip* lists and extracts such filenames exactly as they're stored (excepting truncation, conversion of unsupported characters, etc.); this option causes the names of all files from certain systems to be converted to lowercase. The **-LL** option forces conversion of every filename to lowercase, regardless of the originating file system.

-M

pipe all output through an internal pager similar to the Unix [more\(1\)](#) command. At the end of a screenful of output, *unzip* pauses with a ``--More--" prompt; the next screenful may be viewed by pressing the Enter (Return) key or the space bar. *unzip* can be terminated by pressing the ``q" key and, on some systems, the Enter/Return key. Unlike Unix [more\(1\)](#), there is no forward-searching or editing capability. Also, *unzip* doesn't notice if long lines wrap at the edge of the screen, effectively resulting in the printing of two or more lines and the likelihood that some text will scroll off the top of the screen before being viewed. On some systems the number of available lines on the screen is not detected, in which case *unzip* assumes the height is 24 lines.

-n

never overwrite existing files. If a file already exists, skip the extraction of that file without prompting. By default *unzip* queries before extracting any file that already exists; the user may choose to overwrite only the current file, overwrite all files, skip extraction of the current file, skip extraction of all existing files, or rename the current file.

-N

[Amiga] extract file comments as Amiga filenotes. File comments are created with the **-c** option of *zip*(1L), or with the **-N** option of the Amiga port of *zip*(1L), which stores filenotes as comments.

-o

overwrite existing files without prompting. This is a dangerous option, so use it with care. (It is often used with **-f**, however, and is the only way to overwrite directory EAs under OS/2.)

-P *password*

use *password* to decrypt encrypted zipfile entries (if any). **THIS IS INSECURE!** Many multi-user operating systems provide ways for any user to see the current command line of any other user; even on stand-alone systems there is always the threat of over-the-shoulder peeking. Storing the plaintext password as part of a command line in an automated script is even worse. Whenever possible, use the non-echoing, interactive prompt to enter passwords. (And where security is truly important, use strong encryption such as Pretty Good Privacy instead of the relatively weak encryption provided by standard zipfile utilities.)

-q

perform operations quietly (**-qq** = even quieter). Ordinarily *unzip* prints the names of the files it's extracting or testing, the extraction methods, any file or zipfile comments that may be stored in the archive, and possibly a summary when finished with each archive. The **-q[q]** options suppress the printing of some or all of these messages.

-s

[OS/2, NT, MS-DOS] convert spaces in filenames to underscores. Since all PC operating systems allow spaces in filenames, *unzip* by default extracts filenames with spaces intact (e.g., ``EA DATA. SF``). This can be awkward, however, since MS-DOS in particular does not gracefully support spaces in filenames. Conversion of spaces to underscores can eliminate the awkwardness in some cases.

-U

(obsolete; to be removed in a future release) leave filenames uppercase if created under MS-DOS, VMS, etc. See **-L** above.

-v

retain (VMS) file version numbers. VMS files can be stored with a version number, in the format file.ext;##. By default the ``;##`` version numbers are stripped, but this option allows them to be retained. (On file systems that limit filenames to particularly short lengths, the version numbers may be truncated or stripped regardless of this option.)

-X

[VMS, Unix, OS/2, NT] restore owner/protection info (UICs) under VMS, or user and group info (UID/GID) under Unix, or access control lists (ACLs) under certain network-enabled versions of OS/2 (Warp Server with IBM LAN Server/Requester 3.0 to 5.0; Warp Connect with IBM Peer 1.0), or security ACLs under Windows NT. In most cases this will require special system privileges, and doubling the option (**-XX**) under NT instructs *unzip* to use privileges for extraction; but under Unix, for example, a user who belongs to several groups can restore files owned by any of those groups, as long as the user IDs match his or her own. Note that ordinary file attributes are always restored--this option applies only to optional, extra ownership info available on some operating systems. [NT's access control lists do not appear to be especially compatible with OS/2's, so no attempt is made at cross-platform portability of access privileges. It is not clear under what conditions this would ever be useful anyway.]

-\$

[MS-DOS, OS/2, NT] restore the volume label if the extraction medium is removable (e.g., a diskette). Doubling the option (**-\$-\$**) allows fixed media (hard disks) to be labelled as well. By default, volume labels are ignored.

-I *extensions*

[Acorn only] overrides the extension list supplied by Unzip\$Ext environment variable. During extraction, filename extensions that match one of the items in this extension list are swapped in front of the base name of the extracted file.

-:

[all but Acorn, VM/CMS, MVS, Tandem] allows to extract archive members into locations outside of the current `` extraction root folder``. For security reasons, *unzip* normally removes ``parent dir`` path components (``../``) from the names of extracted file. This safety feature (new for version 5.50) prevents *unzip* from accidentally writing files to ``sensitive`` areas outside the active extraction folder tree head. The **-:** option lets *unzip* switch back to its previous, more liberal behaviour, to allow exact extraction of (older) archives that used ``../`` components to create multiple directory trees at the level of the current extraction folder.

ENVIRONMENT OPTIONS

unzip's default behavior may be modified via options placed in an environment variable. This can be done with any

option, but it is probably most useful with the **-a**, **-L**, **-C**, **-q**, **-o**, or **-n** modifiers: make *unzip* auto-convert text files by default, make it convert filenames from uppercase systems to lowercase, make it match names case-insensitively, make it quieter, or make it always overwrite or never overwrite files as it extracts them. For example, to make *unzip* act as quietly as possible, only reporting errors, one would use one of the following commands:

Unix Bourne shell:

```
UNZIP=-qq; export UNZIP
```

Unix C shell:

```
setenv UNZIP -qq
```

OS/2 or MS-DOS:

```
set UNZIP=-qq
```

VMS (quotes for *lowercase*):

```
define UNZIP_OPTS ""-qq"
```

Environment options are, in effect, considered to be just like any other command-line options, except that they are effectively the first options on the command line. To override an environment option, one may use the "minus operator" to remove it. For instance, to override one of the quiet-flags in the example above, use the command

```
unzip --q[other options] zipfile
```

The first hyphen is the normal switch character, and the second is a minus sign, acting on the *q* option. Thus the effect here is to cancel one quantum of quietness. To cancel both quiet flags, two (or more) minuses may be used:

```
unzip -t--q zipfile
```

```
unzip ---qt zipfile
```

(the two are equivalent). This may seem awkward or confusing, but it is reasonably intuitive: just ignore the first hyphen and go from there. It is also consistent with the behavior of Unix [nice\(1\)](#).

As suggested by the examples above, the default variable names are UNZIP_OPTS for VMS (where the symbol used to install *unzip* as a foreign command would otherwise be confused with the environment variable), and UNZIP for all other operating systems. For compatibility with [zip\(1L\)](#), UNZIPOPT is also accepted (don't ask). If both UNZIP and UNZIPOPT are defined, however, UNZIP takes precedence. *unzip*'s diagnostic option (**-v** with no zipfile name) can be used to check the values of all four possible *unzip* and *zipinfo* environment variables.

The timezone variable (TZ) should be set according to the local timezone in order for the **-f** and **-u** to operate correctly. See the description of **-f** above for details. This variable may also be necessary in order for timestamps on extracted files to be set correctly. Under Windows 95/NT *unzip* should know the correct timezone even if TZ is unset, assuming the timezone is correctly set in the Control Panel.

DECRYPTION

Encrypted archives are fully supported by Info-ZIP software, but due to United States export restrictions, de-/encryption support might be disabled in your compiled binary. However, since spring 2000, US export restrictions have been liberated, and our source archives do now include full crypt code. In case you need binary distributions with crypt support enabled, see the file "WHERE" in any Info-ZIP source or binary distribution for locations both inside and outside the US.

Some compiled versions of *unzip* may not support decryption. To check a version for crypt support, either attempt to test or extract an encrypted archive, or else check *unzip*'s diagnostic screen (see the **-v** option above) for "[decryption]" as one of the special compilation options.

As noted above, the **-P** option may be used to supply a password on the command line, but at a cost in security. The preferred decryption method is simply to extract normally; if a zipfile member is encrypted, *unzip* will prompt for the password without echoing what is typed. *unzip* continues to use the same password as long as it appears to be valid, by testing a 12-byte header on each file. The correct password will always check out against the header, but there is a 1-in-256 chance that an incorrect password will as well. (This is a security feature of the PKWARE zipfile format; it

helps prevent brute-force attacks that might otherwise gain a large speed advantage by testing only the header.) In the case that an incorrect password is given but it passes the header test anyway, either an incorrect CRC will be generated for the extracted data or else *unzip* will fail during the extraction because the ``decrypted" bytes do not constitute a valid compressed data stream.

If the first password fails the header check on some file, *unzip* will prompt for another password, and so on until all files are extracted. If a password is not known, entering a null password (that is, just a carriage return or ``Enter") is taken as a signal to skip all further prompting. Only unencrypted files in the archive(s) will thereafter be extracted. (In fact, that's not quite true; older versions of *zip*(1L) and *zipcloak*(1L) allowed null passwords, so *unzip* checks each encrypted file to see if the null password works. This may result in ``false positives" and extraction errors, as noted above.)

Archives encrypted with 8-bit passwords (for example, passwords with accented European characters) may not be portable across systems and/or other archivers. This problem stems from the use of multiple encoding methods for such characters, including Latin-1 (ISO 8859-1) and OEM code page 850. DOS *PKZIP* 2.04g uses the OEM code page; Windows *PKZIP* 2.50 uses Latin-1 (and is therefore incompatible with DOS *PKZIP*); Info-ZIP uses the OEM code page on DOS, OS/2 and Win3.x ports but Latin-1 everywhere else; and Nico Mak's *WinZip* 6.x does not allow 8-bit passwords at all. *UnZip* 5.3 (or newer) attempts to use the default character set first (e.g., Latin-1), followed by the alternate one (e.g., OEM code page) to test passwords. On EBCDIC systems, if both of these fail, EBCDIC encoding will be tested as a last resort. (EBCDIC is not tested on non-EBCDIC systems, because there are no known archivers that encrypt using EBCDIC encoding.) ISO character encodings other than Latin-1 are not supported.

EXAMPLES

To use *unzip* to extract all members of the archive *letters.zip* into the current directory and subdirectories below it, creating any subdirectories as necessary:

```
unzip letters
```

To extract all members of *letters.zip* into the current directory only:

```
unzip -j letters
```

To test *letters.zip*, printing only a summary message indicating whether the archive is OK or not:

```
unzip -tq letters
```

To test *all* zipfiles in the current directory, printing only the summaries:

```
unzip -tq \*.zip
```

(The backslash before the asterisk is only required if the shell expands wildcards, as in Unix; double quotes could have been used instead, as in the source examples below.) To extract to standard output all members of *letters.zip* whose names end in *.tex*, auto-converting to the local end-of-line convention and piping the output into [more](#)(1):

```
unzip -ca letters \*.tex | more
```

To extract the binary file *paper1.dvi* to standard output and pipe it to a printing program:

```
unzip -p articles paper1.dvi | dvips
```

To extract all FORTRAN and C source files--*.f, *.c, *.h, and Makefile--into the /tmp directory:

```
unzip source.zip ".*[fch]" Makefile -d /tmp
```

(the double quotes are necessary only in Unix and only if globbing is turned on). To extract all FORTRAN and C source files, regardless of case (e.g., both *.c and *.C, and any makefile, Makefile, MAKEFILE or similar):

```
unzip -C source.zip "*.fch" makefile -d /tmp
```

To extract any such files but convert any uppercase MS-DOS or VMS names to lowercase and convert the line-endings of all of the files to the local standard (without respect to any files that might be marked ``binary``):

```
unzip -aaCL source.zip "*.fch" makefile -d /tmp
```

To extract only newer versions of the files already in the current directory, without querying (NOTE: be careful of unzipping in one timezone a zipfile created in another--ZIP archives other than those created by Zip 2.1 or later contain no timezone information, and a ``newer`` file from an eastern timezone may, in fact, be older):

```
unzip -fo sources
```

To extract newer versions of the files already in the current directory and to create any files not already there (same caveat as previous example):

```
unzip -uo sources
```

To display a diagnostic screen showing which *unzip* and *zipinfo* options are stored in environment variables, whether decryption support was compiled in, the compiler with which *unzip* was compiled, etc.:

```
unzip -v
```

In the last five examples, assume that UNZIP or UNZIP_OPTS is set to -q. To do a singly quiet listing:

```
unzip -l file.zip
```

To do a doubly quiet listing:

```
unzip -ql file.zip
```

(Note that the ``.zip`` is generally not necessary.) To do a standard listing:

```
unzip --ql file.zip
```

or

```
unzip -l-q file.zip
```

or

```
unzip -l--q file.zip
```

(Extra minuses in options don't hurt.)

TIPS

The current maintainer, being a lazy sort, finds it very useful to define a pair of aliases: `tt` for ```unzip -tq``` and `ii` for ```unzip -Z``` (or ```zipinfo```). One may then simply type ```tt zipfile``` to test an archive, something that is worth making a habit of doing. With luck `unzip` will report ```No errors detected in compressed data of zipfile.zip,``` after which one may breathe a sigh of relief.

The maintainer also finds it useful to set the UNZIP environment variable to ```-aL``` and is tempted to add ```-C``` as well. His ZIPINFO variable is set to ```-z```.

SEE ALSO

[funzip\(1L\)](#), [zip\(1L\)](#), [zipcloak\(1L\)](#), [zipgrep\(1L\)](#), [zipinfo\(1L\)](#), [zipnote\(1L\)](#), [zipsplit\(1L\)](#)

URL

The Info-ZIP home page is currently at

<http://www.info-zip.org/pub/infozip/>

or

<ftp://ftp.info-zip.org/pub/infozip/> .

Important: Use the `man` command (`% man`) to see how a command is used on your particular computer.

[>> Linux/Unix Command Library](#)

[>> Shell Command Library](#)



From [Juergen Haas](#),
Your Guide to [Focus on Linux](#).
FREE Newsletter. [Sign Up Now!](#)

[Topic Index](#) | [Email to a Friend](#) | [Bookmark this Site](#) | [Make this Site Your Homepage!](#) | [Print this Page](#)

[Our Story](#) | [Be a Guide](#) | [Advertising Info](#) | [Work at About](#) | [Site Map](#) | [Icons](#) | [Help](#)
[User Agreement](#) | [Ethics Policy](#) | [Patent Info.](#) | [Privacy Policy](#) | [Kids' Privacy Policy](#)

©2006 About, Inc., A part of [The New York Times Company](#). All rights reserved.