



WinTAR™

TAR for Windows NT, Windows 95, and Win32s

Brings the most popular Unix backup utility to users of Microsoft Windows

Copyright © 1994-96 SpiralCom Communications Inc.

Contents

WinTar

[What's New](#)

[Introduction](#)

[Product Highlights](#)

[System Requirements](#)

Using WinTar

[Installation](#)

[Menu Commands](#)

[Tool Bars](#)

[Remote Device Access](#)

[Frequently Asked Questions](#)

[Trouble-shooting](#)

[Several Unix Command References](#)

[UnInstall WinTAR](#)

SpiralCom

[About SpiralCom Communications Inc.](#)

[Acknowledgment](#)

[Licence Agreement](#)

[Benefits of Register WinTAR](#)

[Order Information](#)

[Technical Support](#)

What's New

Ø Full 32-bit implementation

WinTAR uses Microsoft's Win32 APIs. Careful selection of the Win32 APIs allows WinTAR to run in Win32s V1.30A (under Windows 3.1x and Windows for Workgroup), Windows 95, and Windows NT 3.5x.

Ø Supports long file names (up to 260 characters including the drive letter)

Except within Win32s environment, WinTAR now supports long filenames that are part of Windows 95's VFAT and Windows NT's NTFS. The traditional TAR format only supports filename sizes of up to 100 characters. WinTAR uses the extensions as used in GNU TAR, available to virtually all Unix platforms, to add this feature.

Ø Supports extended DOS attributes

WinTAR now includes HIDDEN, SYSTEM, and ARCHIVE attributes as found in DOS and Microsoft Windows products. Although additional information are included in the TAR format, they should still be compatible to the existing TAR implementations found on Unix machines.

Ø Supports extended file time (Only available in Registered version)

With VFAT and NTFS, each file contains not only the modification time but also the creation and accessing times. WinTAR backs up these file times also if the user instructs it to do so. Please be careful when using this feature since it has been shown that some Unix TAR implementations (such as the ones found on Sun Solaris 2.3 or above and DEC Alpha OSF/1) are not compatible with this format.

Ø Supports registry (only under Windows 95 and Windows NT)

WinTAR now uses registry to store its internal data and user setups. The location for the registry follows the guidelines as suggested by Microsoft.

Ø Supports larger blocking factor (only under Windows 95 and Windows NT)

WinTAR now allows the user to specify a blocking factor up to 256 blocks or 128 kilobytes.

Introduction

TAR is a very popular archiving program in the Unix world. In a networked heterogeneous computing environment consisting of workstations and PCs which use different file formats, it is desirable to have a single file backup application that can access one type of backup media such as a tape drive.

We all know that backing up files regularly is a good computing practice. Use of tape drives to backup data is very common nowadays. In fact, in a networked computing environment where all workstations are connected together, there is usually at least one tape backup unit. When a PC is connected to such a network, is there any way to backup data from the local hard disk without purchasing a different type of tape drive solely for data backup purpose?

TAR for Windows (WinTAR) can solve this problem for you. WinTAR uses Windows Sockets to communicate with the rest of network. The communication protocol follows that of TCP/IP (Transmission Control Protocol/Internet Protocol) which is another popular network protocol. Explanation of the operations is beyond the scope of this text, but a few points are summarized here:

- Ø WinTAR uses port 512 (Remote exec) or port 514 (Remote shell) for data transfer.
- Ø It relies on the availability of workstations providing Unix commands or equivalents such as dd, mt, tar, gzip, compress, and zcat. (The last three commands are used if compression is needed.)

Product Highlights

- Ø User interface to the popular "tar" program in Windows 3.1x and Windows for Workgroup with Win32s 1.30A, Windows 95, and Windows NT 3.5x.
- Ø Can create/open archive files on a local hard disk.
- Ø Can create/open archive directly on 3.5" 1.44MB floppy diskettes. The disks can directly be read/written on a Sun workstation equipped with 3.5" floppy drives.
- Ø Can create/open compressed or normal archive files on a remote hard disk.
- Ø Can create/open archive on any storage medium supported by your workstation.
- Ø Can manipulate your remote tape drive (forward, rewind, and "end of media") through icons similar to a cassette tape deck.
- Ø Supports templates for archiving directory list and for remote connection parameters.
- Ø Supports GNU compatible volume header and long file name extensions.

System Requirements

The following requirements are needed for WinTAR to run properly:

- ∅ A 386 or above IBM compatible PC with at least 8 MB RAM and 2MB free hard disk space.
- ∅ Windows 3.1 or Windows for Workgroup with Win32s 1.30A installed, Windows 95, or Windows NT 3.5x
- ∅ Windows Sockets DLL (winsock.dll) that is fully Version 1.1 compliant (such as Trumpet Winsock and Microsoft TCP/IP)

The Winsock DLL can interface to an ethernet card or a modem (using SLIP, PPP, etc.) or whatever hardware interface that it supports.
- ∅ (Optional) Unix workstations equipped with storage devices such as tape drives and availability of services "remote exec" and/or "remote shell".

Installation

Please make sure that your distribution contains the following files:

- ct13d32.dll
- licence.txt
- readme.txt
- wintar.exe
- wintar.hlp
- wintar.inf

If you miss any files, please contact William Cheung via email at wcheung@ee.ubc.ca and indicate where you have downloaded the distribution.

If you are using Windows 95, click [here](#).

If you are using Windows NT, click [here](#).

If you are using Windows 3.1x or Windows for Workgroup 3.11, click [here](#).

Installation - Windows 95

Follow the instructions below for software installation.

- 1) After unzipping wtar20u.zip to a temporary directory, use the explorer to open up that directory.
- 2) Right-clicked on the WinTAR icon with the extension .inf (the icon has a yellow donut at the lower right hand corner.)
- 3) Select "Install".
- 4) Follow the instructions as displayed to locate your temporary unzipped folder. Windows 95 may ask you if you want to restart the computer. Answer "Yes".
- 5) WinTAR will be installed under c:\Program Files\Spiralcm\WinTAR and a program group is created with the name "SpiralCom WinTAR".

Installation - Windows NT

- 1) Create a destination directory for WinTAR, e.g. c:\WinTAR, and unzip all files into that directory.
- 2) Move ctl3d32.dll to your Windows installation directory under system\, e.g. c:\windows\system (Remember to remove ctl3d32.dll from your WinTAR installation directory.)
- 3) Add wintar.exe to an existing program group or you can create a new program group for WinTAR. Please make sure that the working directory is the same as the directory where WinTAR.exe is located or the program cannot start properly.

Installation - Win32s for Windows 3.1

- 1) If your Win32s is not at least Version 1.30A, you should obtain the latest from one of Microsoft's distribution channel. The upgrade is free of charge. Please see the [FAQ](#) section in the WinTAR help on how to obtain your copy.
- 2) Create a destination directory for WinTAR, e.g. c:\WinTAR, and unzip all files into that directory.
- 3) Move ctl3d32.dll to your Windows installation directory under system\, e.g. c:\windows\system (Remember to remove ctl3d32.dll from your WinTAR installation directory.)
- 4) Add wintar.exe to an existing program group or you can create a new program group for WinTAR. Please make sure that the working directory is the same is the directory where WinTAR.exe is located or the program cannot start properly.
- 5) If you are using Windows 3.1x and not Windows for Workgroup, you must add the following line your config.sys if it is not already there. Then, reboot your computer.

```
install=c:\dos\share.exe
```

Uninstall WinTAR

For Windows 95 users only

If you have followed the installation steps as outlined in "[Installation - Windows 95](#)", follow the steps below to uninstall WinTAR. Otherwise, read the **Manual Uninstall** section below.

- Ø Open up your Control Panel and double-clicked on "Add/Remove Programs" icon.
- Ø In the software list, find and double-clicked on "SpiralCom WinTAR".
- Ø Change directory to "C:\Program Files" and remove the directory "SPIRALCM".

Manual Uninstall WinTAR

This section tells you how to properly uninstall WinTAR manually if you decide to remove WinTAR from your system:

- Ø Remove the WinTAR icon from your program group.
- Ø Remove the directory where your WinTAR is installed. Please note that under Windows 95, a hidden file, `wintar.gid`, may be created in the directory where WinTAR is installed. You need to run "`attrib -h -r wintar.gid`" on that file before you can remove it.
- Ø In your Windows directory, remove "wintar.log"
- Ø If you are using Windows 3.1 or Windows for Workgroup, remove "wintar.ini" from your Windows directory.
- Ø If you are using Windows 95 or Windows NT, open up your Registry editor program (*regedit* under Windows 95 and *regedt32* under Windows NT.) Remove the following keys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\SpiralCom\WinTAR  
HKEY_CURRENT_USER\Software\SpiralCom\WinTAR
```

Menu Commands

File Menu

[New Archive](#)
[Open Archive](#)
[Close Archive](#)

Options Menu

[Save Settings on Exit](#)
[Prompt user on all connections](#)
[Log all error messages](#)
[View Log File](#)
[Select Log File](#)
[Preferences](#)

Utilities Menu

Rewind Tape
Forward Tape
End of Media
Eject/Offline Media
Media Information

Help Menu

Contents
About

Menu Commands

File Menu

New Archive

This item creates an archive. Its options are:

- Local** creates a local archive file on the local hard disk
- Local Floppy** creates a local archive directly on a 3.5" floppy disk. Only 1.44MB HD floppy diskettes are supported.
- Remote** creates a remote archive file on a workstation or writes directly to a storage media such as a tape drive on a remote workstation, depending on the remote command used.

Upon selecting any of the above options, the user will be prompted to select directories, files, and/or drives to backup from. The drive letter is always eliminated in the archive. The user has four options to store the file names:

- Ø *Relative to Drive Letter*: The drive letter together with the first "/" is dropped, e.g. "c:\windows\system\commdlg.dll" is converted to "windows/system/commdlg.dll".
- Ø *Relative to Sub-directory*: The whole directory tree is eliminated, e.g. "c:\windows\system\commdlg.dll" is converted to "commdlg.dll". If the backup entry is a directory name, the displayed directory name will be eliminated but the rest of the files/directories are kept in their original location relative to the directory name as shown in the backup list. This has the same effect as changing to the displayed directory and execute "tar cvf - ." in that directory on a Unix machine. *Please note that you should only include one directory for archiving if you select this option or the archived directory tree may be lumped together.* The following example should clarify this option:

Let's say we want to back up the Windows directory as shown below:

```
c:\windows\notepad.exe
: : : :
c:\windows\system\commdlg.dll
: : : :
c:\windows\system\win32s\comdlg32.dll
: : : :
```

Then, if "c:\windows" is displayed in the backup file/directory list, the following will be shown in the archive:

```
./
notepad.exe
: : : :
system/commdlg.dll
: : : :
system/win32s/comdlg32.dll
: : : :
```

- Ø *Absolute Path*: Only the drive letter is eliminated, e.g. "c:\windows\system\commdlg.dll" is converted to "/windows/system/commdlg.dll".
- Ø *Full Path*: The complete path (including drive letter) is included, e.g. "c:\windows\system\commdlg.dll" is converted to "c:/windows/system/commdlg.dll". This option is useful to

backup a multi-partition hard disk to a single archive in one step.

The volume name/template is used to save the current archive selection into a template so that the same set of files/directories can be used in the future. The template name will then be used as the volume name for the archive. This volume header information is compatible with GNU's implementation of TAR. If this entry is empty or if the "No volume name in Archive" option is enabled, no volume information will be added to the archive. This allows backward compatibility with Sun's TAR utility. The default selection is always "default", and this setting cannot be removed or edited. To add a new directory template:

1. Put the files/directories you want to be archived in the "Include in Archive" list if the list is not already there.
2. Click in the directory template and enter a name for your template.
3. Click the "Save Config" button to save the template.

To select a predefined template, click on the "down arrow" button and select the one that you want. The list is then shown in "Include in Archive".

When the user has finished the selections, click the OK button, and the user will be prompted for the destination of the archive:

Local	open a common file saving dialog box to input an archive file name that must end with ".tar"
Local Floppy	prompts to select a floppy drive to backup to.
Remote	prompts for the remote host name, user name, and password to access the remote machine. Also shown is the remote command that will be executed after a connection has established. The user can change the command by selecting <u>Options</u> button.

During backup, the user can move the WinTAR window around or minimize the window. In addition, if, for any reasons, the user wants to stop the current process, he/she can press the **Stop** button.

After backing up, the user can optionally re-read the archive to verify the archive. Please note that if you are backing up to a remote tape drive, it takes time for the tape drive to rewind (if you are using a rewinding device). Hence, if you re-read the archive immediately, you may get a device busy error from the remote program.

Menu Commands

File Menu

Open Archive

This item opens an existing archive. Its options are:

- Local** opens a local archive file on the local hard disk
- Local Floppy** opens a local archive directly on a 3.5" floppy disk. Only 1.44MB HD floppy diskettes are supported.
- Remote** opens a remote archive file on a workstation or writes directly to a storage media such as a tape drive on a remote workstation, depending on the remote command used.

Upon selecting any of the above options, the user will be prompted for the source of the archive:

- Local** open a common file opening dialog box to input an archive file name that must end with ".tar"
- Local Floppy** prompts to select a floppy drive to restore from.
- Remote** prompts for the remote host name, user name, and password to access the remote machine. Also shown is the remote command that will be executed after a connection has established. The user can change the command by selecting **Options** button.

Opening an archive only displays a list of files and directories that are available in the archive. This is equivalent to the "-t -v" options in "tar". If the user wants to stop listing the files, he/she can press the **Stop** button. This is especially needed if you want to restore a whole hard disk from a remote tape drive. If the user wants the whole list again, he/she can press the "Re-read archive" button. The user can extract all or a selected list of files/directories from the archive. For the first option, the user simply clicks on the Extract button. For the second option, the user selects one or more files from the file list using the standard file selection convention in Windows applications. (Use Control-Click or Shift-Click to select multiple files.) Then, the user will be prompted for the destination directory.

In the extract archive dialog box, the user is asked to input a destination directory. This directory can be any directory listed in the list box or a user created one. (To create a new destination directory, simply type in a valid directory name in the text input box.) Now the user has the option to restore the archive in various ways:

- Ø *Strip drive letter*: The drive letter is dropped, e.g. "c:/windows/system/commdlg.dll" is converted to "/windows/system/commdlg.dll". This option is useful to restore various logical drives into one drive. Please note that the destination for "/windows/system/commdlg.dll" is dependent only on the drive specified in the "destination" text input box. For example, if the text in the input box is "d:\" or "d:\restored", the final destination will be "d:/windows/system/commdlg.dll".
- Ø *Strip first "/"*: Any text before the first "/" is dropped, e.g. "c:\windows\system\commdlg.dll" is converted to "windows/system/commdlg.dll". Please note that selecting this option will automatically select "Strip drive letter".
- Ø *Strip all directories*: Any text before the last "/" is dropped. This option has the effect of flattening a directory structure into a "no-directory" structure. Please note selecting this option will automatically select the above two options, e.g. "c:\windows\system\commdlg.dll" is converted to "commdlg.dll".

If none of the above options are selected, the archive will be restored to whatever is shown in the file list if all files in the file list were backed up using the "Full Path" option, or to a path relative to the destination

path you have specified if all files in the file list were backed up using any “Relative Path” options. If the drive shown in the archive does not exist (may be due to repartition a hard disk or moving the archive to a different hard disk,) the user will be prompted for a new drive letter.

Menu Commands

File Menu

Close Archive

This item closes the current archive in active.

Menu Commands

Options Menu

Save Settings on Exit

Checking this item saves the options in the Preference dialog box and other settings.

Menu Commands

Options Menu

Prompt User on All Connections

Checking this item prompts the user for the host name, user name, and password whenever any commands that require remote connection. When unchecked, the user will be prompted initially for the host name, user name, and password, and any subsequent remote connections will use the same connection information. The remote command will be as shown in the **Advanced** options in the **Preference** section.

Menu Commands

Options Menu

Log All Error Messages

Checking this item logs all messages displayed in the status bar to the log file "wintar.log" located in the WinTAR installation directory. All messages are appended to this file.

Menu Commands

Options Menu *Preferences*

This item prompts the user for various options that are available for WinTAR:

Option	Meaning
Re-block during reading	is needed on all remote archive connections (-B)
Ignore blocks of zeros	continue even when archive checksum error occurs (-i)
Keep existing files	do not overwrite existing files (-k)
Don't extract file modification time	use the current time as the file modification time (-m)
Extract all protection information	restore the read/write permissions as in the archive (-p)
Use old V7 format	when creating an archive, write an old format archive that does not include information about directories, pipes, or device files. This option is only provided for compatibility for older archives (-o)
Include Access/Create Time	include the access time and create time under Windows 95 and Windows NT. <i>Please note that the resulting TAR format may not be compatible with your generic Unix TAR program.</i> WinTAR is compatible with GNU TAR.
Blocking Factor	specify the block size for transferring data to the archive. This is a default value for all local archives. Typically the larger the number, the faster is the transfer time. However, there are a few restrictions depending on the options used and the archiving medium: <i>Floppy drives(R/W)</i> the factor is automatically switched to 18 for the most optimized performance. <i>Local archive file(R/W)</i> can be any number but the suggested value is 20. <i>Remote archive file/tape (R/W)</i> The value is determined from the remote device database. R = reading (extracting), W = writing

Remote commands can be changed via the **Advanced** button.

Menu Commands

Options Menu

Advanced Button

Within the **Advanced** button, the following options are available for operations on the remote host:

Option	Meaning
Read command	read (extract) archive
Write command	write archive
Compress command	compress archive on the fly
Decompress command	decompress archive on the fly
Rewind command	rewind tape
End of device command	forward to the end of media
Forward Tape command	forward to next archive command (Use %d to allow this value to be changed)
Tape Status command	retrieve the tape/media information such as the tape drive type
Response Delay	The time between a command being sent to the remote host and a response being received by WinTAR. Sixty seconds is a good value.

For all of the above commands, use "%s" in the command for device name substitution and "%d" in the command for blocking factor substitution. (In the forward tape command, "%d" means the number of archives to advance.)

In addition, a remote device database is also set up in this dialog box. There are two pieces of information shown in the list. The number in square bracket shows the blocking factor while the name on the right shows the device name.

The **Add** button adds new devices/files to the list. When this button is clicked, the user is prompted for a new device name and a blocking factor for that device. The default value is 20.

The **Edit** button changes the value of the currently selected device. This is the same as double clicking any selected device. The user is prompted to change the existing values for the device.

The **Remove** button removes the selected device on the device list.

Menu Commands

Options Menu

View Log File

This option is only enabled when a log file exists. This option enables the user to view the logged error messages. The log file is located in your Windows installation directory, by default, with the name “wintar.log” unless the user has selected a different log file.

Menu Commands

Options Menu

Select Log File

This option is used to change the log file to a different name and/or location. Some users prefer the log file to be located somewhere instead of the Windows directory.

This item rewinds the tape to the beginning

This item prompts the user for how many archives to be skipped. The value asked will be substituted into "%d" in the ***Forward Tape command***.

This item forwards the tape to the end so that new archives can be added.

This item ejects a tape on a remote host.

This item gets the media type and other information.

This item displays this help text.

This item displays information about WinTAR.

Tool Bars

Most of the frequently used menu items are replicated in the toolbar. When the cursor is above any of the active toolbar buttons, a brief description of the corresponding function is displayed in the status bar at the bottom of the window.

The group of buttons on the left side of WinTAR windows deals with creating, opening, and extracting archives in various locations:



Create archive on local hard disk



Open archive from local hard disk



Create archive on a remote storage device



Open archive on a remote storage device



Create archive on local floppy drive directly



Open archive from local floppy drive directly



Re-read archive list from the opened archive



Extract selected files/directories from the archive. Refer to [Open Archive](#) for information about the procedure in extracting files from an archive.



Close archive from an already open archive.

The group of buttons on the right side of WinTAR windows deals with manipulating the tape drive, and they are self explanatory:



Obtain tape information



Eject/Make off-line a remote media



Rewind tape to the beginning



Stop current processing



Forward tape to the next archive



Forward tape to the end of the last archive

Also, the set of buttons < (back one page), > (forward one page), **H** (first page), and **E** (last page) allows you to control the “page” of directory you want to go to.

Remote Device Access

Whenever the user opens or creates an archive on a remote workstation, he/she is accessing a remote device. After accessing the device, the user is prompted for several pieces of remote access information:

- Template:** Specifies a remote access device template. To add a template, fill in all the information below, click in the template box, enter a template name, and click on "Save Config" button. To select a template, click on the "down arrow" button, and select the one that you want. Please note that for security reasons, the password is not saved as part of the template. It is however kept in memory for the duration of running WinTAR.
- Host name:** The workstation name or IP address to be connected to. Normally a host list is displayed if the environment variable RSHHOST_DIR is set to the directory where the local host data base file "hosts" exists. Some sites rely on Domain Name Servers (DNS) entirely and do not have a "hosts" file. In that case, the list would be empty. However, whenever you enter a host name that is not on the host list, you will be asked whether to add the host name to the host list. If you answer "Yes", that host name will appear in the host list everytime you are requested to log onto a remote host.
- User name:** A valid user name on the network.
- Password:** Password for the user. This is used only when the "Remote EXEC" service is used.
- Device/File:** A remote device or file to be used. To add a device, you must click the Options button. To use this feature, you must use "%s" instead of a physical device name when setting up the remote commands.
- Protocol:** The service protocol to be used. The supported ones are "RExec" for Remote EXEC (Port 512) and "RSH" for Remote Shell (Port 514).
- Compress Archive:** Compress the archive on the remote device. This option should not be used for a tape device.
- Do not Read Content:** When opening an archive, do not read in the content. Checking this option simply changes the current operation state to "Open Remote Archive". It does not make any connection to the destination machine.

Trouble-shooting

General

1) Upon program startup, WinTAR complains that "CTL3D32.DLL" is not installed properly.

- Ø Is CTL3D32.DLL in your Windows system directory?
- Ø Have you remove CTL3D32.DLL from your WinTAR installation directory?

2) Upon program startup, WinTAR complains that a license file is missing or corrupted.

- Ø Make sure that "licence.txt" is in the same directory as "wintar.exe", and your working directory is point to your WinTAR installation directory.

Network

1) During network connection, WinTAR reports:

a) **Connection is aborted due to time-out**

- Ø Is the response delay value large enough? The response delay should be at least 30 seconds.
- Ø Try to use a network monitoring program such as "ping" to check the connection.

b) **Error from network: (Some messages)**

- Ø Please check your Unix manual pages corresponding to the service protocol for the meaning of the error messages. (The manual pages are [rexec\(8c\)](#) and [rshd\(8c\)](#).)

c) **Other time-out messages (From send or receive)**

- Ø Try to use a network monitoring program such as "ping" to check the connection.

2) During any tape operations, WinTAR reports:

a) **Cannot receive data from the network (No response). Abort connection ?**

- Ø Is the response delay value large enough? The response delay should be at least 60 seconds.
- Ø Try to use a network monitoring program such as "ping" to check the connection.
- Ø With operations like rewind or forward, the response is usually quite slow. You can abort the connection (since the command is still executing on your workstation) You can then check whether the operation is done using the Tape Information command. If the command is done, Tape Information should return the current status of the tape; otherwise, "Device busy" error message should be received.

3) After opening the archive, the user tries to extract some files from archive but nothing is extracted.

- Ø What type of tape device are you using? (Rewinding or Non-rewinding) A rewinding device rewinds the tape after each device access. A non-rewinding device only rewinds the tape when it is requested.
- Ø Have you used the forward tape command? If so, you have to, first of all, rewind the tape to the beginning and forward it to the desire archive location again using the forward tape command. Then, you should issue your extract command.

4) During remote backup process, the user receives "Send failed. The virtual circuit has been reset by the remote side".

- Ø Please check whether you have a "core" dump file on your Unix workstation. There have been reports that some rexecd implementations cannot handle a large amount of data.

WinTAR Setup/Compatibility

- 1) **After backing up to a remote device, I use "tar tv" or similar commands to read the contents of the archive on my Unix workstation but two very huge numbers precedes my file names. If I try to use "tar xv" to extract the archive on my Unix workstation, those numbers are treated as directory names.**

- Ø The registered version of WinTAR uses GNU's extension to store a file's access and creation dates. Some Unix TAR programs use those regions to store the directory name of the archive file if the file path has a length greater than 100 characters. You can solve this program by using GNU TAR or uncheck the "Include Access/Create Time" option in the *Preference* Menu. (The default is to include the Access and Create times.)

- 2) **After backing up to a remote device, I use "tar tv" or similar commands to read the contents of the archive on my Unix workstation but some file and directory names are replaced by "@LongLink" or something similar.**

- Ø The original TAR format can only support a maximum path size of 100 characters. GNU TAR extends this format by storing file names longer than 100 characters in the next block and replaces the current block with a file name "@LongLink". WinTAR follows the same approach. It is recommended that if you have path names in Windows 95 and Windows NT that are longer than 100 characters, you should use GNU TAR on the Unix side to extract the archive.

Frequently Asked Questions (FAQs)

This page tries to answer some of the commonly asked questions about WinTAR. They are by no means complete.

Ø Why can't I find any options in accessing my tape drive situated on my PC?

WinTAR does not support local tape drives, either QIC-type or SCSI, currently. This option may be supported in a future release.

Ø Can I use WinTAR to duplicate a directory tree from a PC to a hard disk on a Unix workstation like what "rcp" does?

Yes. The trick is to change the remote write command in the Advanced options dialog box to "tar xvBbf %d -" where %d is the blocking factor.

Ø Why does WinTAR limit the blocking factor to a maximum of 63 under Win32s?

This is mainly due to the 16-bit interface of Winsock under Win32s and 16-bit Windows. The following FAQ has a workaround solution.

Ø Is there any way to tell WinTAR under Win32s to use a blocking factor bigger than 63 blocks because my remote tape drive is optimized for 126 blocks?

To workaround the limitations under Win32s, you can specify the blocking factor to be still 63, but explicitly put your desired blocking factor to 126 in the remote commands. Please make sure that the two blocking factors should be a multiple of each other to avoid some misalignment problems.

For example:

Write: dd of=%s ibs=%db obs=126b

Read: dd if=%s ibs=126b obs=%db

Ø Why is the total archive size shown in WinTAR much less than the size shown using "chkdsk" or "Properties" under Windows 95?

This is related to the way the file system under MSDOS (using FAT) and Windows 95 (using VFAT) partitions your hard disk. Under FAT or VFAT, each file size is a multiple of a certain minimum block size depending on the size of the hard disk partition. For example, a 1 GB hard disk partition has a block size of 32KB. Hence, a file which is only one byte still takes up 32KB hard disk space. WinTAR counts the archive size using the actual size of the file and NOT the size as allocated by FAT. Hence, the total archive size can be considerably smaller.

Ø How can I speed up my backup time?

You can decrease your backup time by increasing the blocking factor. The default value is 20 blocks, and a typical throughput using this value is 3 to 4 MB per minute. Change this value to 63 blocks increases the throughput to 8 to 12 MB per minute. Note that this corresponds to approximately 1.3 Mbps which is a typical effective throughput value under a 10 Mbps ethernet. Hence, there is no point in increasing this blocking factor further unless your remote backup device requires a larger blocking factor for optimum performance.

Ø Where can I find the latest version of WinTAR?

You can find the most up-to-date version of WinTAR by ftping to com8.ee.ubc.ca as an anonymous user, and change directory to /pub/pc/wintar. Read the README file in that directory to see which one is the latest version.

Ø Where can I find Win32s 1.30A required by WinTAR?

You can find Win32s 1.30A for Windows 3.1, Windows 3.11, and Windows for Workgroup 3.11 via the following channels:

CompuServe

GO MSL

Search for PW1118.EXE

Display Results and download

Internet (anonymous FTP)

<ftp://ftp.microsoft.com/SOFTLIB/MSLFILES/PW1118.EXE>

Several Unix Command References

Please select one of the Unix command manual pages from below:

- [compress](#)
- [dd](#)
- [gunzip](#)
- [gzip](#)
- [mt](#)
- [rexecd](#)
- [rshd](#)
- [tar](#)
- [uncompress](#)
- [zcat - Sun's version](#)
- [zcat - GNU's version](#)

About SpiralCom Communications Inc.

SpiralCom Communications Inc. is a privately owned incorporated company situated in the beautiful city of Vancouver, British Columbia, Canada. SpiralCom was founded in September 1995 and is jointly owned by William Cheung and Danny Lam.

After one year of marketing WinTAR under Cheung-Smith Ltd., William Cheung feels that there is a market for WinTAR in many large corporations, government agencies, universities, and software companies. Therefore, William has decided to form SpiralCom Communications Inc. with Danny Lam in order to expand the functionality of WinTAR. Soon after it is formed, SpiralCom bought the ownership of WinTAR, including all source codes, executables, and documentation, from Cheung-Smith Ltd. With the new structure in this new company, we hope that we can make WinTAR a success and a useful backup utility for anyone using Microsoft Windows Operating Systems in a heterogenous network environment.

Acknowledgment

Part of the tar engine code is based on the Public Domain TAR by John Gilmore (1985).

I would like to thank the following people who have beta tested WinTAR 1.x:

V. Carpenter	(CARPENTERV@VMS.CSD.MU.EDU)
Charlie Chang	(chaolin.chang@fcit.monash.edu.au)
Mike Fessler	(MIKEF@BROWNVN.brown.edu)
David W. Forslund	(dwf@acl.lanl.gov)
Reynaldo Flores	(reynaldo.flores@dseg.ti.com)
Blake Freeburg	(blake@piglet.portable.com)
Michael W. Grenier	(mike@planet8.sp.paramax.com)
Ulrich R. Herken	(urh@specs.de)
Martin Husemann	(martin@euterpe.owl.de)
Stefan Kjellberg	(stefank@esi.com.au)
Greg Kulosa	(gkulosa@oberon.com)
Michael C. Loewen	(mloewen@cpumagic.scol.pa.us)
Karl Lovink	(karl@class3.apd.dec.com)
Krishna Nayak	(nayak@ibm1.scri.fsu.edu)
Kevin Rosenberg	(kevin@nlm.nih.gov)

In particular, I would like to give special thanks to Stefan and Ulrich who have given me lots of feedback and useful suggestions.

Benefits of Registering WinTAR

- ∅ The licence agreement dialog box disappears during WinTAR startup.
- ∅ The complete archive listing is shown in the list box. This allows the user to restore any files in the archive.
- ∅ The user is allowed to backup the full path name including the drive letter. This eases the user to backup a multi-partition hard disk.
- ∅ The user has the option to backup the accessed and created times of a file in addition to the modified time as supported by both Windows 95 and Windows NT.
- ∅ You are a proud user of the registered version of WinTAR!
- ∅ Special discounted price on future upgrades to major releases of WinTAR.

Licence Agreement

There are two sets of license agreement for WinTAR:

- ∅ If you have the shareware version, you should read [this](#) one.
- ∅ If you have the registered version, you should read [this](#) one.

Licence Agreement

Shareware Version

WinTAR (TAR for Windows)

Copyright (c) 1994-96 SpiralCom Communications Inc. All Rights Reserved

IMPORTANT-READ CAREFULLY: This SpiralCom End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and SpiralCom Communications Inc. for using the registered version of WinTAR (the "SHAREWARE PRODUCT" or SHAREWARE"). By installing, copying, or otherwise using the SHAREWARE, you agree to be bound by the terms of this EULA.

SHAREWARE PRODUCT LICENSE

The SHAREWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SHAREWARE PRODUCT is licensed, not sold. This agreement shall be governed by the laws of the Province of British Columbia, Canada.

You are hereby licensed to:

- 1) use the SHAREWARE for a 21 day evaluation period,
- 2) make as many copies of the SHAREWARE and documentation as you wish,
- 3) give exact copies of the original SHAREWARE to anyone,
- 4) distribute the SHAREWARE and documentation in its unmodified form via electronic means.

There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the SHAREWARE and/or documentation with other products commercial or otherwise) without prior written permission.

Using the SHAREWARE after the 21-day evaluation period is in violation of federal copyright laws.

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. Because of the variety of hardware and software environments into which the SHAREWARE may be put, **NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.**

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program. **ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE. NEITHER THE AUTHOR NOR SPIRALCOM COMMUNICATIONS INC. SHALL BE HELD LIABLE FOR ANY KIND OF DAMAGES OR CLAIMS THAT DIRECTLY OR INDIRECTLY RESULT FROM USING THIS SHAREWARE.**

Licence Agreement

Registered Version

WinTAR (TAR for Windows)

Copyright (c) 1994-96 SpiralCom Communications Inc. All Rights Reserved

IMPORTANT-READ CAREFULLY: This SpiralCom End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and SpiralCom Communications Inc. for using the registered version of WinTAR (the "SOFTWARE PRODUCT" or "SOFTWARE"). By installing, copying, or otherwise using the SOFTWARE, you agree to be bound by the terms of this EULA.

SOFTWARE PRODUCT LICENSE

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold. This agreement shall be governed by the laws of the Province of British Columbia, Canada.

UPGRADES VERSION

If the SOFTWARE is an upgrade, whether from SpiralCom or another supplier, you may use or transfer the SOFTWARE only in conjunction with the upgraded product.

REGISTERED VERSION

One copy of the SOFTWARE may either be used by a single person who uses the software personally on one or more computers, or installed on a single workstation used non-simultaneously by multiple people, but not both. The SOFTWARE may not be duplicated other than for backup purposes. The SOFTWARE must be treated like a book, in that the same copy of the SOFTWARE may not be used in more than one computer at the same time, just as a book may not be read by more than one person at the same time (without making an illegal copy).

You may access the SOFTWARE through a network, provided that you have obtained a site license for the SOFTWARE covering all workstations and/or PCs that will access the SOFTWARE through the network.

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. Because of the variety of hardware and software environments into which the SOFTWARE may be put, **NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.**

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program. **ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE. NEITHER THE AUTHOR NOR SPIRALCOM COMMUNICATIONS INC. SHALL BE HELD LIABLE FOR ANY KIND OF DAMAGES OR CLAIMS THAT DIRECTLY OR INDIRECTLY RESULT FROM USING THIS SOFTWARE.**

Order Information

WinTAR 2.0 Order Form

Ordering by cheque: To order by cheque, send this order form and a cheque to:

SpiralCom Communications Inc.
211 W63rd Avenue,
Vancouver, B.C.
Canada V5X 2H9

In Canada or USA: Payments can be in US or Canadian dollars drawn on a Canadian or US bank.
Other countries: Payments must be in US or Canadian dollars drawn on a Canadian or US bank (i.e. international bank draft), or you can send international postal money orders in either US or Canadian dollars.

*Please make cheque payable to **SpiralCom Communications Inc.*** All orders are F.O.B. from the above address. Please allow 4 to 6 weeks for delivery.

Site licences: A site licence for WinTAR entitles an organization to receive one copy of the distribution package and duplicate the distribution disk for the specified number of copies. Please note that **prices are calculated in an incremental fashion.** See below for a sample calculation.

Regular Prices (For unregistered new users)

WinTAR Single Copy: _____ copies at US\$40 (CDN\$52), each = _____

WinTAR Site Licence

2 to 9 computers: _____ computers at US\$35 (CDN\$45), each = _____
10 to 49 computers: _____ computers at US\$27 (CDN\$35), each = _____
50+ computers: _____ computers at US\$18 (CDN\$24), each = _____

Upgrade Prices (ONLY for users who have registered WinTAR 1.4x before Nov. 20, 1995)*

Registration Number: _____ **Date of Purchase:** _____

Name: _____ **Company Name:** _____

WinTAR Single Copy: _____ copies at US\$10 (CDN\$13), each = _____

WinTAR Site Licence

2 to 9 computers: _____ computers at US\$9 (CDN\$12), each = _____
10 to 49 computers: _____ computers at US\$6 (CDN\$8), each = _____
50+ computers: _____ computers at US\$4 (CDN\$5), each = _____

Shipping & handling** US\$5 (CDN\$7) = _____
Canadian residents add 7% GST (GST#897477550RT) + _____
B.C. residents add 7% PST + _____

Total payment
=====

* Using post-stamp to determine your order date.

** Shipping and handling fee is required only if the distribution is via regular mail. See below.

Technical Support

Technical support is available only through electronic mail. You should, first of all, check whether the [FAQ](#) or [Trouble-shooting](#) sections in this manual can solve your problems. If not, you can reach William Cheung at wcheung@ee.ubc.ca.

You can always find the latest WinTAR release from <ftp://com8.ee.ubc.ca/pub/pc/wintar/>.

Bug Reports

When you submit a bug report, please have the following information ready. This can considerably speed up the process.

- Ø WinTAR Version number:
- Ø Machine configuration: (CPU, memory)
- Ø Windows Socket Vendor and Version number (if applicable): (e.g. PCNFS Version 5.1a)
- Ø Window System: (Win32s 1.30A, Windows 95, Windows NT)
- Ø Other Winsock applications you are running concurrently if any.
- Ø Settings/Parameters in WinTAR you have used (including the remote command you have used): (Blocking factor, etc.)
- Ø Detailed description of the bug(s) including the steps in causing the problem.
- Ø Is/Are the bug(s) repeatable?

NAME

compress, uncompress, zcat – compress or expand files, display expanded contents

SYNOPSIS

compress [-cfv] [-b *bits*] [*filename...*]

uncompress [-cv] [*filename...*]

zcat [*filename...*]

DESCRIPTION

compress reduces the size of the named files using adaptive Lempel-Ziv coding. Whenever possible, each file is replaced by one with the extension **.Z**, while keeping the same ownership modes, as well as access and modification times. If no files are specified, the standard input is compressed to the standard output.

The amount of compression obtained depends on the size of the input, the number of *bits* per code, and the distribution of common substrings. Typically, text such as source code or English is reduced by 50–60%. Compression is generally much better than that achieved by Huffman coding (as used in **sys-unconfig(8)**), or adaptive Huffman coding (**old-compact(1)**), and takes less time to compute. The *bits* parameter specified during compression is encoded within the compressed file, along with a magic number to ensure that neither decompression of random data nor recompression of compressed data is subsequently allowed.

Compressed files can be restored to their original form using **uncompress**.

zcat produces uncompressed output on the standard output, but leaves the compressed **.Z** file intact.

OPTIONS

- c** Write to the standard output; no files are changed. The nondestructive behavior of **zcat** is identical to that of '**uncompress-**'.
- f** Force compression, even if the file does not actually shrink, or the corresponding **.Z** file already exists. Except when running in the background (under **sh(1)**), if **-f** is not given, prompt to verify whether an existing **.Z** file should be overwritten.
- v** Verbose. Display the percentage reduction for each file compressed.
- bbits** Set the upper limit (in *bits*) for common substring codes. *bits* must be between 9 and 16 (16 is the default).

SEE ALSO

ln(1V), **old-compact(1)**, **sh(1)**, **sys-unconfig(8)**

A Technique for High Performance Data Compression, Terry A. Welch, *computer*, vol. 17, no. 6 (June 1984), pp. 8-19.

DIAGNOSTICS

Exit status is normally 0. If the last file was not compressed because it became larger, the status is 2. If an error occurs, exit status is 1.

Usage: compress [-fvc] [-bmaxbits] [*filename...*]

Invalid options were specified on the command line.

Missing maxbits

Maxbits must follow **-b**.

filename: **not in compressed format**

The file specified to uncompress has not been compressed.

filename: **compressed with xx bits, can only handle yy bits**

filename was compressed by a program that could deal with more bits than the compress code on this machine. Recompress the file with smaller bits.

filename: already has .Z suffix--nochange

The file is assumed to be already compressed. Rename the file and try again.

filename: already exists; do you wish to overwrite (y or n)?

Respond **y** if you want the output file to be replaced; **n** if not.

uncompress: corrupt input

A SIGSEGV violation was detected, which usually means that the input file is corrupted.

Compression: xx.xx%

Percentage of the input saved by compression. (Relevant only for **-v**.)

— not a regular file: unchanged

When the input file is not a regular file, (such as a directory), it is left unaltered.

— has xx other links: unchanged

The input file has links; it is left unchanged. See **ln(1V)** for more information.

— file unchanged

No savings are achieved by compression. The input remains uncompressed.

BUGS

Although compressed files are compatible between machines with large memory, **-b12** should be used for file transfer to architectures with a small process data space (64KB or less).

compress should be more flexible about the existence of the **.Z** suffix.

NAME

`dd` – convert and copy files with various data formats

SYNOPSIS

`dd` [option=*value*] ...

DESCRIPTION

`dd` copies a specified input file to a specified output with possible conversions. The standard input and output are used by default. The input and output block size may be specified to take advantage of raw physical I/O.

OPTIONS

if=*name* Input file is taken from *name*; standard input is default.

of=*name* Output file is taken from *name*; standard output is default. Note: `dd` creates an explicit output file; therefore the **seek** option is usually useless with explicit output except in special cases such as using magnetic tape or raw disk files.

ibs=*n* Input block size *n* bytes (default 512).

obs=*n* Output block size *n* bytes (default 512).

bs=*n* Set both input and output block size, superseding **ibs** and **obs**; also, if no conversion is specified, it is particularly efficient since no copy need be done. Block sizes for the Sun386i are 9k for 3.5-inch floppy disks, and 126b (blocks) for quarter-inch tapes.

cbs=*n* Conversion buffer size.

skip=*n* Skip *n* input records before starting copy

files=*n* Copy *n* input files before terminating (makes sense only when input is a magtape or similar device).

seek=*n* Seek *n* records from beginning of output file before copying. This option generally only works with magnetic tapes and raw disk files and is otherwise usually useless if the explicit output file was named with the **of** option.

count=*n* Copy only *n* input records.

conv=ascii Convert EBCDIC to ASCII.

ebcdic Convert ASCII to EBCDIC.

ibm Slightly different map of ASCII to EBCDIC.

block Convert variable length records to fixed length.

unblock Convert fixed length records to variable length.

lcase Map alphabetic to lower case.

ucase Map alphabetic to upper case.

swab Swap every pair of bytes.

noerror Do not stop processing on an error.

sync Pad every input record to **ibs**.

arg,arg [...]
Several comma-separated conversions, for a combination of effects. For instance, **conv=sync,block** is useful for reading variable-length output from a pipe.

Where sizes are specified, a number of bytes is expected. A number may end with **k** (kilobytes) to specify multiplication by 1024, **b** (blocks of 512 bytes) to specify multiplication by 512, or **w** (words) to specify multiplication by 4; a pair of numbers may be separated by **x** to indicate a product.

cbs is used only if **ascii**, **unblock**, **ebcdic**, **ibm**, or **block** conversion is specified. In the first two cases, **cbs** characters are placed into the conversion buffer, any specified character mapping is done, trailing blanks trimmed and **NEWLINE** added before sending the line to the output. In the latter three cases, characters are read into the conversion buffer, and blanks added to make up

an output record of size **cbs**.

After completion, **dd** reports the number of whole and partial input and output blocks.

EXAMPLES

To read an EBCDIC tape blocked ten 80-byte EBCDIC card images per record into the ASCII file *x*:

```
example% dd if=/dev/rmt0 of=x ibs=800 cbs=80 conv=ascii,lcase
```

Note: the use of raw magtape: **dd** is especially suited to I/O on the raw physical devices because it allows reading and writing in arbitrary record sizes.

Sun386i EXAMPLES

The following write the file *filename* to a 3.5-inch floppy and read from the floppy into a file *filename*, respectively:

```
example% dd if=filename of=/dev/rfd0c bs=9k
```

```
example% dd if=/dev/rfd0c of=filename bs=9k
```

Sun386i files names are shown in **fdformat(1)**.

SEE ALSO

cp(1), **fdformat(1)**, **tr(1V)**

DIAGNOSTICS

***f+p* records in(out):**

Numbers of full and partial records read(written).

BUGS

The ASCII/EBCDIC conversion tables are taken from the 256 character standard in the CACM Nov, 1968. The **ibm** conversion, while less blessed as a standard, corresponds better to certain IBM print train conventions. There is no universal solution.

The **block** and **unblock** options cannot be combined with the **ascii**, **ebcdic** or **ibm**. Invalid combinations silently ignore all but the last mutually-exclusive keyword.

NAME

gzip, gunzip, zcat – compress or expand files

SYNOPSIS

```
gzip [-acdfhLnNrtvV19] [-S suffix] [ name ... ]
gunzip [-acfhLnNrtvV] [-S suffix] [ name ... ]
zcat [-fhLV] [ name ... ]
```

DESCRIPTION

Gzip reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension **.gz**, while keeping the same ownership modes, access and modification times. (The default extension is **-gz** for VMS, **z** for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. *Gzip* will only attempt to compress regular files. In particular, it will ignore symbolic links.

If the compressed file name is too long for its file system, *gzip* truncates it. *Gzip* attempts to truncate only the parts of the file name longer than 3 characters. (A part is delimited by dots.) If the name consists of small parts only, the longest parts are truncated. For example, if file names are limited to 14 characters, *gzip.msdos.exe* is compressed to *gzi.msdx.exe.gz*. Names are not truncated on systems which do not have a limit on file name length.

By default, *gzip* keeps the original file name and timestamp in the compressed file. These are used when decompressing the file with the **-N** option. This is useful when the compressed file name was truncated or when the time stamp was not preserved after a file transfer.

Compressed files can be restored to their original form using *gzip -d* or *gunzip* or *zcat*. If the original name saved in the compressed file is not suitable for its file system, a new name is constructed from the original one to make it legal.

gunzip takes a list of files on its command line and replaces each file whose name ends with **.gz**, **-gz**, **.z**, **-z**, **_z** or **.Z** and which begins with the correct magic number with an uncompressed file without the original extension. *gunzip* also recognizes the special extensions **.tgz** and **.taz** as shorthands for **.tar.gz** and **.tar.Z** respectively. When compressing, *gzip* uses the **.tgz** extension if necessary instead of truncating a file with a **.tar** extension.

gunzip can currently decompress files created by *gzip*, *zip*, *compress*, *compress -H* or *pack*. The detection of the input format is automatic. When using the first two formats, *gunzip* checks a 32 bit CRC. For *pack*, *gunzip* checks the uncompressed length. The standard compress format was not designed to allow consistency checks. However *gunzip* is sometimes able to detect a bad **.Z** file. If you get an error when uncompressing a **.Z** file, do not assume that the **.Z** file is correct simply because the standard *uncompress* does not complain. This generally means that the standard *uncompress* does not check its input, and happily generates garbage output. The SCO *compress -H* format (lzh compression method) does not include a CRC but also allows some consistency checks.

Files created by *zip* can be uncompressed by *gzip* only if they have a single member compressed with the 'deflation' method. This feature is only intended to help conversion of tar.zip files to the tar.gz format. To extract zip files with several members, use *unzip* instead of *gunzip*.

zcat is identical to *gunzip -c*. (On some systems, *zcat* may be installed as *gzcat* to preserve the original link to compress.) *zcat* uncompresses either a list of files on the command line or its standard input and writes the uncompressed data on standard output. *zcat* will uncompress files that have the correct magic number whether they have a **.gz** suffix or not.

Gzip uses the Lempel-Ziv algorithm used in *zip* and PKZIP. The amount of compression obtained depends on the size of the input and the distribution of common substrings. Typically, text such as source code or English is reduced by 60–70%. Compression is generally much better than that achieved by LZW (as used in *compress*), Huffman coding (as used in *pack*), or adaptive Huffman coding (*compact*).

Compression is always performed, even if the compressed file is slightly larger than the original. The worst case expansion is a few bytes for the gzip file header, plus 5 bytes every 32K block, or an expansion ratio of 0.015% for large files. Note that the actual number of used disk blocks almost never increases. *gzip* preserves the mode, ownership and timestamps of files when compressing or decompressing.

OPTIONS

-a --ascii

Ascii text mode: convert end-of-lines using local conventions. This option is supported only on some non-Unix systems. For MSDOS, CR LF is converted to LF when compressing, and LF is converted to CR LF when decompressing.

-c --stdout --to-stdout

Write output on standard output; keep original files unchanged. If there are several input files, the output consists of a sequence of independently compressed members. To obtain better compression, concatenate all input files before compressing them.

-d --decompress --uncompress

Decompress.

-f --force

Force compression or decompression even if the file has multiple links or the corresponding file already exists, or if the compressed data is read from or written to a terminal. If the input data is not in a format recognized by *gzip*, and if the option `--stdout` is also given, copy the input data without change to the standard output: let *zcat* behave as *cat*. If `-f` is not given, and when not running in the background, *gzip* prompts to verify whether an existing file should be overwritten.

-h --help

Display a help screen and quit.

-l --list For each compressed file, list the following fields:

compressed size: size of the compressed file
uncompressed size: size of the uncompressed file
ratio: compression ratio (0.0% if unknown)
uncompressed_name: name of the uncompressed file

The uncompressed size is given as -1 for files not in gzip format, such as compressed .Z files. To get the uncompressed size for such a file, you can use:

```
zcat file.Z | wc -c
```

In combination with the `--verbose` option, the following fields are also displayed:

method: compression method
crc: the 32-bit CRC of the uncompressed data
date & time: time stamp for the uncompressed file

The compression methods currently supported are deflate, compress, lzh (SCO compress -H) and pack. The crc is given as ffffffff for a file not in gzip format.

With `--name`, the uncompressed name, date and time are those stored within the compress file if present.

With `--verbose`, the size totals and compression ratio for all files is also displayed, unless some sizes are unknown. With `--quiet`, the title and totals lines are not displayed.

-L --license

Display the *gzip* license and quit.

-n --no-name

When compressing, do not save the original file name and time stamp by default. (The original name is always saved if the name had to be truncated.) When decompressing, do not restore the original file name if present (remove only the *gzip* suffix from the compressed file name) and do not restore the original time stamp if present (copy it from

the compressed file). This option is the default when decompressing.

-N --name

When compressing, always save the original file name and time stamp; this is the default. When decompressing, restore the original file name and time stamp if present. This option is useful on systems which have a limit on file name length or when the time stamp has been lost after a file transfer.

-q --quiet

Suppress all warnings.

-r --recursive

Travel the directory structure recursively. If any of the file names specified on the command line are directories, *gzip* will descend into the directory and compress all the files it finds there (or decompress them in the case of *gunzip*).

-S .suf --suffix .suf

Use suffix *.suf* instead of *.gz*. Any suffix can be given, but suffixes other than *.z* and *.gz* should be avoided to avoid confusion when files are transferred to other systems. A null suffix forces *gunzip* to try decompression on all given files regardless of suffix, as in:

```
gunzip -S "" *      (*. * for MSDOS)
```

Previous versions of *gzip* used the *.z* suffix. This was changed to avoid a conflict with *pack(1)*.

-t --test

Test. Check the compressed file integrity.

-v --verbose

Verbose. Display the name and percentage reduction for each file compressed or decompressed.

-V --version

Version. Display the version number and compilation options then quit.

-# --fast --best

Regulate the speed of compression using the specified digit #, where **-1** or **—fast** indicates the fastest compression method (less compression) and **-9** or **—best** indicates the slowest compression method (best compression). The default compression level is **-6** (that is, biased towards high compression at expense of speed).

ADVANCED USAGE

Multiple compressed files can be concatenated. In this case, *gunzip* will extract all members at once. For example:

```
gzip -c file1 > foo.gz
```

```
gzip -c file2 >> foo.gz
```

Then

```
gunzip -c foo
```

is equivalent to

```
cat file1 file2
```

In case of damage to one member of a *.gz* file, other members can still be recovered (if the damaged member is removed). However, you can get better compression by compressing all members at once:

```
cat file1 file2 | gzip > foo.gz
```

compresses better than

```
gzip -c file1 file2 > foo.gz
```

If you want to recompress concatenated files to get better compression, do:

```
gzip -cd old.gz | gzip > new.gz
```

If a compressed file consists of several members, the uncompressed size and CRC reported by the `--list` option applies to the last member only. If you need the uncompressed size for all members, you can use:

```
gzip -cd file.gz | wc -c
```

If you wish to create a single archive file with multiple members so that members can later be extracted independently, use an archiver such as tar or zip. GNU tar supports the `-z` option to invoke gzip transparently. gzip is designed as a complement to tar, not as a replacement.

ENVIRONMENT

The environment variable **GZIP** can hold a set of default options for *gzip*. These options are interpreted first and can be overwritten by explicit command line parameters. For example:

```
for sh:  GZIP="-8v --name"; export GZIP
```

```
for csh: setenv GZIP "-8v --name"
```

```
for MSDOS: set GZIP=-8v --name
```

On Vax/VMS, the name of the environment variable is GZIP_OPT, to avoid a conflict with the symbol set for invocation of the program.

SEE ALSO

znew(1), zcmp(1), zmore(1), zforce(1), gzexe(1), zip(1), unzip(1), compress(1), pack(1), compact(1)

DIAGNOSTICS

Exit status is normally 0; if an error occurs, exit status is 1. If a warning occurs, exit status is 2.

Usage: gzip [-cdfhLnNrtvV19] [-S suffix] [file ...]

Invalid options were specified on the command line.

file: not in gzip format

The file specified to *gunzip* has not been compressed.

file: Corrupt input. Use zcat to recover some data.

The compressed file has been damaged. The data up to the point of failure can be recovered using

```
zcat file > recover
```

file: compressed with *xx* bits, can only handle *yy* bits

File was compressed (using LZW) by a program that could deal with more *bits* than the decompress code on this machine. Recompress the file with gzip, which compresses better and uses less memory.

file: already has .gz suffix -- no change

The file is assumed to be already compressed. Rename the file and try again.

file already exists; do you wish to overwrite (y or n)?

Respond "y" if you want the output file to be replaced; "n" if not.

gunzip: corrupt input

A SIGSEGV violation was detected which usually means that the input file has been corrupted.

xx.x%

Percentage of the input saved by compression. (Relevant only for `-v` and `-l`.)

-- not a regular file or directory: ignored

When the input file is not a regular file or directory, (e.g. a symbolic link, socket, FIFO, device file), it is left unaltered.

-- has *xx* other links: unchanged

The input file has links; it is left unchanged. See *ln(1)* for more information. Use the **-f** flag to force compression of multiply-linked files.

CAVEATS

When writing compressed data to a tape, it is generally necessary to pad the output with zeroes up to a block boundary. When the data is read and the whole block is passed to *gunzip* for decompression, *gunzip* detects that there is extra trailing garbage after the compressed data and emits a warning by default. You have to use the **--quiet** option to suppress the warning. This option can be set in the **GZIP** environment variable as in:

```
for sh: GZIP="-q" tar -xfz --block-compress /dev/rst0
```

```
for csh: (setenv GZIP -q; tar -xfz --block-compr / dev/rst0
```

In the above example, *gzip* is invoked implicitly by the **-z** option of GNU *tar*. Make sure that the same block size (**-b** option of *tar*) is used for reading and writing compressed data on tapes. (This example assumes you are using the GNU version of *tar*.)

BUGS

The **--list** option reports incorrect sizes if they exceed 2 gigabytes. The **--list** option reports sizes as **-1** and **crc** as **ffffff** if the compressed file is on a non seekable media.

In some rare cases, the **--best** option gives worse compression than the default compression level (**-6**). On some highly redundant files, *compress* compresses better than *gzip*.

NAME

`mt` – magnetic tape control

SYNOPSIS

`mt [-f tapename] command [count]`

DESCRIPTION

`mt` sends commands to a magnetic tape drive. If *tapename* is not specified, the environment variable **TAPE** is used. If **TAPE** does not exist, `mt` uses the device `/dev/rmt12`. *tapename* refers to a raw tape device. By default, `mt` performs the requested operation once; multiple operations may be performed by specifying *count*.

The available commands are listed below. Only as many characters as are required to uniquely identify a command need be specified.

`mt` returns a 0 exit status when the operation(s) were successful, 1 if the command was unrecognized or if `mt` was unable to open the specified tape drive, and 2 if an operation failed.

OPTIONS

eof,weof	Write <i>count</i> EOF marks at the current position on the tape.
fsf	Forward space over <i>count</i> EOF marks. The tape is positioned on the first block of the file.
fsr	Forward space <i>count</i> records.
bsf	Back space over <i>count</i> EOF marks. The tape is positioned on the beginning-of-tape side of the EOF mark.
bsr	Back space <i>count</i> records.
nbsf	Back space <i>count</i> files. The tape is positioned on the first block of the file. This is equivalent to <i>count</i> +1 bsf 's followed by one fsf .
asf	Absolute space to <i>count</i> file number. This is equivalent to a rewind followed by a fsf <i>count</i> .

For the following commands, *count* is ignored:

eom	Space to the end of recorded media on the tape. This is useful for appending files onto previously written tapes.
rewind	Rewind the tape.
offline,rewoffl	Rewind the tape and, if appropriate, take the drive unit off-line by unloading the tape.
status	Print status information about the tape unit.
retension	Rewind the cartridge tape completely, then wind it forward to the end of the reel and back to beginning-of-tape to smooth out tape tension.
erase	Erase the entire tape.

FILES

<code>/dev/rmt*</code>	magnetic tape interface
<code>/dev/rar*</code>	Archive cartridge tape interface
<code>/dev/rst*</code>	SCSI tape interface

SEE ALSO

`ar(4S)`, `mtio(4)`, `st(4S)`, `tm(4S)`, `xt(4S)` `environ(5V)`

BUGS

Not all devices support all options. Some options are hardware-dependent. Refer to the corresponding device manual page.

WARNINGS

The **bsf** option for SCSI tape in SunOS 4.0.3 is incompatible with this release and releases prior to 4.0.3.

NAME

rexecd, in.rexecd – remote execution server

SYNOPSIS

/usr/etc/in.rexecd host.port

AVAILABILITY

This program is available with the *Networking* software installation option. Refer to **INSTALL** for information on how to install optional software.

DESCRIPTION

rexecd is the server for the **rexec(3N)** routine. The server provides remote execution facilities with authentication based on user names and encrypted passwords. It is invoked automatically as needed by **inetd(8C)**, and then executes the following protocol:

- o The server reads characters from the socket up to a null (**\0**) byte. The resultant string is interpreted as an ASCII number, base 10.
- o If the number received in step 1 is non-zero, it is interpreted as the port number of a secondary stream to be used for the **stderr**. A second connection is then created to the specified port on the client's machine.
- o A null terminated user name of at most 16 characters is retrieved on the initial socket.
- o A null terminated, encrypted, password of at most 16 characters is retrieved on the initial socket.
- o A null terminated command to be passed to a shell is retrieved on the initial socket. The length of the command is limited by the upper bound on the size of the system's argument list.
- o **rexecd** then validates the user as is done at login time and, if the authentication was successful, changes to the user's home directory, and establishes the user and group protections of the user. If any of these steps fail the connection is aborted with a diagnostic message returned.
- o A null byte is returned on the connection associated with the **stderr** and the command line is passed to the normal login shell of the user. The shell inherits the network connections established by **rexecd**.

SEE ALSO

rexec(3N) **inetd(8C)**,

DIAGNOSTICS

All diagnostic messages are returned on the connection associated with the **stderr**, after which any network connections are closed. An error is indicated by a leading byte with a value of 1 (0 is returned in step 7 above upon successful completion of all the steps prior to the command execution).

username too long

The name is longer than 16 characters.

password too long

The password is longer than 16 characters.

command too long

The command line passed exceeds the size of the argument list (as configured into the system).

Login incorrect.

No password file entry for the user name existed.

Password incorrect.

The wrong password was supplied.

No remote directory.

The **chdir** command to the home directory failed.

Try again.

A **fork** by the server failed.

/usr/bin/sh:...

The user's login shell could not be started.

BUGS

Indicating '**Login incorrect**' as opposed to '**Password incorrect**' is a security breach which allows people to probe a system for users with null passwords.

A facility to allow all data exchanges to be encrypted should be present.

NAME

rshd, in.rshd – remote shell server

SYNOPSIS

/usr/etc/in.rshd host.port

DESCRIPTION

rshd is the server for the **rcmd**(3N) routine and, consequently, for the **rsh**(1C) program. The server provides remote execution facilities with authentication based on privileged port numbers.

rshd is invoked by **inetd**(8C) each time a shell service is requested, and executes the following protocol:

- o The server checks the client's source port. If the port is not in the range 512-1023, the server aborts the connection. The client's host address (in hex) and port number (in decimal) are the argument passed to **rshd**.
- o The server reads characters from the socket up to a null (**\0**) byte. The resultant string is interpreted as an ASCII number, base 10.
- o If the number received in step 1 is non-zero, it is interpreted as the port number of a secondary stream to be used for the **stderr**. A second connection is then created to the specified port on the client's machine. The source port of this second connection is also in the range 512-1023.
- o The server checks the client's source address. If the address is associated with a host for which no corresponding entry exists in the host name data base (see **hosts**(5)), the server aborts the connection.
- o A null terminated user name of at most 16 characters is retrieved on the initial socket. This user name is interpreted as a user identity to use on the **server's** machine.
- o A null terminated user name of at most 16 characters is retrieved on the initial socket. This user name is interpreted as the user identity on the **client's** machine.
- o A null terminated command to be passed to a shell is retrieved on the initial socket. The length of the command is limited by the upper bound on the size of the system's argument list.
- o **rshd** then validates the user according to the following steps. The remote user name is looked up in the password file and a **chdir** is performed to the user's home directory. If the lookup fails, the connection is terminated. If the **chdir** fails, it does a **chdir** to / (**root**). If the user is not the super-user, (user ID 0), the file **/etc/hosts.equiv** is consulted for a list of hosts considered ***(|equivalent|)***(r). If the client's host name is present in this file, the authentication is considered successful. If the lookup fails, or the user is the super-user, then the file **.rhosts** in the home directory of the remote user is checked for the machine name and identity of the user on the client's machine. If this lookup fails, the connection is terminated.
- o A null byte is returned on the connection associated with the **stderr** and the command line is passed to the normal login shell of the user. The shell inherits the network connections established by **rshd**.

FILES

/etc/hosts.equiv

SEE ALSO

rsh(1C), **rcmd**(3N), **syslogd**(8)

BUGS

The authentication procedure used here assumes the integrity of each client machine and the

connecting medium. This is insecure, but is useful in an “open” environment.
A facility to allow all data exchanges to be encrypted should be present.

DIAGNOSTICS

The following diagnostic messages are returned on the connection associated with the **stderr**, after which any network connections are closed. An error is indicated by a leading byte with a value of 1 (0 is returned in step 9 above upon successful completion of all the steps prior to the command execution).

locuser too long

The name of the user on the client’s machine is longer than 16 characters.

remuser too long

The name of the user on the remote machine is longer than 16 characters.

command too long

The command line passed exceeds the size of the argument list (as configured into the system).

Hostname for your address unknown.

No entry in the host name database existed for the client’s machine.

Login incorrect.

No password file entry for the user name existed.

Permission denied.

The authentication procedure described above failed.

Can’t make pipe.

The pipe needed for the **stderr**, was not created.

Try again.

A *fork* by the server failed.

/usr/bin/sh:...

The user’s login shell could not be started.

In addition, daemon’s status messages and internal diagnostics are logged to the appropriate system log using the **syslogd**(8) facility.

NAME

tar – create tape archives, and add or extract files

SYNOPSIS

```
tar [ - ] c|r|t|u|x [ bBefFhilmopvwX014578 ] [ tarfile ] [ blocksize ] [ exclude-file ] [-I include-file]  
filename1 filename2 ... -C directory filenameN ...
```

DESCRIPTION

tar archives and extracts multiple files onto a single **tar**, file archive, called a *tarfile*. A *tarfile* is usually a magnetic tape, but it can be any file. **tar**'s actions are controlled by the first argument, the key, a string of characters containing exactly one function letter from the set **crtux**, and one or more of the optional function modifiers listed below. Other arguments to **tar** are file or directory names that specify which files to archive or extract. In all cases, the appearance of a directory name refers recursively to the files and subdirectories of that directory.

FUNCTION LETTERS

- c** Create a new *tarfile* and write the named files onto it.
- r** Write the named files on the end of the *tarfile*. Note: this option does not work with quarter-inch archive tapes.
- t** List the table of contents of the *tarfile*.
- u** Add the named files to the *tarfile* if they are not there or if they have been modified since they were last archived. Note: this option *does not work* with quarter-inch archive tapes.
- x** Extract the named files from the *tarfile*. If a named file matches a directory with contents written onto the tape, this directory is (recursively) extracted. The owner, modification time, and mode are restored (if possible). If no *filename* arguments are given, all files in the archive are extracted. Note: if multiple entries specifying the same file are on the tape, the last one overwrites all earlier versions.

FUNCTION MODIFIERS

- b** Use the next argument as the blocking factor for tape records. The default blocking factor is 20 blocks. The block size is determined automatically when reading tapes (key letters **x** and **t**). This determination of the blocking factor may be fooled when reading from a pipe or a socket (see the **B** key letter below). The maximum blocking factor is determined only by the amount of memory available to **tar** when it is run. Larger blocking factors result in better throughput, longer blocks on nine-track tapes, and better media utilization.
- B** Force **tar** to perform multiple reads (if necessary) so as to read exactly enough bytes to fill a block. This option exists so that **tar** can work across the Ethernet, since pipes and sockets return partial blocks even when more data is coming.
- e** If any unexpected errors occur **tar** will exit immediately with a positive exit status.
- f** Use the next argument as the name of the *tarfile*. If **f** is omitted, use the device indicated by the **TAPE** environment variable, if set. Otherwise, use **/dev/rmt8** by default. If *tarfile* is given as '-', **tar** writes to the standard output or reads from the standard input, whichever is appropriate. Thus, **tar** can be used as the head or tail of a filter chain. **tar** can also be used to copy hierarchies with the command:

```
example% cd fromdir; tar cf - . | (cd todir; tar xfBp -)
```
- F** With one **F** argument specified, exclude all directories named **SCCS** from *tarfile*. With two arguments **FF**, exclude all directories named **SCCS**, all files with **.o** as their suffix, and all files named **errs**, **core**, and **a.out**.
- h** Follow symbolic links as if they were normal files or directories. Normally, **tar** does not

- follow symbolic links.
- i** Ignore directory checksum errors.
- I** Display error messages if all links to archived files cannot be resolved. If **I** is not used, no error messages are printed.
- m** Do not extract modification times of extracted files. The modification time will be the time of extraction.
- o** Suppress information specifying owner and modes of directories which tar normally places in the archive. Such information makes former versions of tar generate an error message like:
 - filename:/cannotcreate**
 when they encounter it.
- p** Restore the named files to their original modes, ignoring the present **umask(2V)**. SetUID and sticky information are also extracted if you are the super-user. This option is only useful with the **x** key letter.
- v** Verbose. Normally **tar** does its work silently; this option displays the name of each file tar treats, preceded by the function letter. When used with the **t** function, **v** displays the *tarfile* entries in a form similar to '**ls -l**'.
- w** Wait for user confirmation before taking the specified action. If you use **w**, **tar** displays the action to be taken followed by the file name, and then waits for a **y** response to proceed. No action is taken on the named file if you type anything other than a line beginning with **y**.
- X** Use the next argument as a file containing a list of named files (or directories) to be excluded from the *tarfile* when using the key letters **c**, **x**, or **t**. Multiple **X** arguments may be used, with one *exclude file* per argument.

014578

Select an alternate drive on which the tape is mounted. The numbers **2**, **3**, **6**, and **9** do not specify valid drives. The default is **/dev/rmt8**.

If a file name is preceded by **-I** then the filename is opened. A list filenames, one per line, is treated as if each appeared separately on the command line. Be careful of trailing white space in both include and exclude file lists.

In the case where excluded files (see **X** option) also exist, excluded files take precedence over all included files. So, if a file is specified in both the include and exclude files (or on the command line), it will be excluded.

.ne "6"

If a file name is preceded by **-C** in a **c** (create) or **r** (replace) operation, **tar** will perform a **chdir** (see **csh(1)**) to that file name. This allows multiple directories not related by a close common parent to be archived using short relative path names. See EXAMPLES below.

Note: the **-C** option only applies to *one* following directory name and one following file name.

EXAMPLES

To archive files from **/usr/include** and from **/etc**, one might use:

```
example% tar c -C /usr include -C /etc .
```

If you get a table of contents from the resulting *tarfile*, you will see something like:

```
include/
```

```
include/a.out.h
and all the other files in /usr/include...
/chown
and all the other files in /etc
```

Here is a simple example using **tar** to create an archive of your home directory on a tape mounted on drive **/dev/rmt0**:

```
example% cd
example% tar cvf /dev/rmt0 .
messages from tar
```

The **c** option means create the archive; the **v** option makes tar tell you what it is doing as it works; the **f** option means that you are specifically naming the file onto which the archive should be placed (**/dev/rmt0** in this example).

Now you can read the table of contents from the archive like this:

```
example% tar tvf /dev/rmt0    display table of contents of the archive
(access user-id/group-id    size  mod. date    filename)
rw-r--r-- 1677/40          2123  Nov 7 18:15:1985    ./archive/test.c
...
example%
```

You can extract files from the archive like this:

```
example% tar xvf /dev/rmt0    extract files from the archive
messagesfromtar
example%
```

If there are multiple archive files on a tape, each is separated from the following one by an EOF marker. **tar** does not read the EOF mark on the tape after it finishes reading an archive file because **tar** looks for a special header to decide when it has reached the end of the archive. Now if you try to use **tar** to read the next archive file from the tape, **tar** does not know enough to skip over the EOF mark and tries to read the EOF mark as an archive instead. The result of this is an error message from **tar** to the effect:

tar: blocksize=0

This means that to read another archive from the tape, you must skip over the EOF marker before starting another **tar** command. You can accomplish this using the **mt(1)** command, as shown in the example below. Assume that you are reading from **/dev/nrmt0**.

```
example% tar xvfp /dev/nrmt0    read first archive from tape
messages from tar
example% mt fsf1                skip over the end-of-file marker
example% tar xvfp /dev/nrmt0    read second archive from tape
messages from tar
example%
```

Finally, here is an example using **tar** to transfer files across the Ethernet. First, here is how to archive files from the local machine (**example**) to a tape on a remote system (**host**):

```
example% tar cvfb - 20 filenames | rsh host dd of=/dev/rmt0 obs=20b
messages from tar
example%
```

In the example above, we are *creating a tarfile* with the **c** key letter, asking for *verbose* output from tar with the **v** option, specifying the name of the output *tarfile* using the **f** option (the standard output is where the *tarfile* appears, as indicated by the '-' sign), and specifying the blocksize (20) with the **b** option. If you want to change the blocksize, you must change the blocksize arguments both on the **tar** command and on the **dd** command.

Now, here is how to use **tar** to get files from a tape on the remote system back to the local

system:

```
example% rsh -n host dd if=/dev/rmt0 bs=20b | tar xvBfb - 20 filenames
messages from tar
example%
```

In the example above, we are *extracting* from the *tarfile* with the **x** key letter, asking for *verbose* output from **tar** with the **v** option, telling **tar** it is reading from a pipe with the **B** option, specifying the name of the input *tarfile* using the **f** option (the standard input is where the *tarfile* appears, as indicated by the '-' sign), and specifying the blocksize (20) with the **b** option.

FILES

/dev/rmt? half-inch magnetic tape interface
/dev/rst? SCSI tape interface
/tmp/tar*

ENVIRONMENT

TAPE If specified, in the environment, the value of **TAPE** indicates the default tape device.

SEE ALSO

cpio(1), **cs(1)**, **mt(1)**, **umask(2V)**, **tar(5)**, **dump(8)**, **restore(8)**

BUGS

Neither the **r** option nor the **u** option can be used with quarter-inch archive tapes, since these tape drives cannot backspace.

There is no way to ask for the *n*th occurrence of a file.

Tape errors are handled ungracefully.

The **u** option can be slow.

There is no way selectively to follow symbolic links.

When extracting tapes created with the **r** or **u** options, directory modification times may not be set correctly.

Files with names longer than 100 characters cannot be processed.

Filename substitution wildcards do not work for extracting files from the archive. To get around this, use a command of the form:

```
tar xvf ... /dev/rst0 'tar tf ... /dev/rst0 | grep 'pattern''
```

