

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Append

Enables programs to open data files in specified directories as if these files were in the current directory.

The specified directories are called *appended directories* because, for the sake of opening data files, they can be found as if they were appended to the current directory.

append [**;**] [[*drive:*]*path*[**;**...]] [**/x**:{**on** | **off**}] [**/path**:{**on** | **off**}] [**/e**]

Parameters

;

Cancels the list of appended directories.

[*drive:*]*path*

Specifies the drive (if other than the current drive) and directory that you want to append to the current directory. You can specify multiple entries of [*drive:*]*path*, separating the entries with semicolons. When used by itself, **append** cancels the existing list of appended directories.

/x:{**on** | **off**}

Specifies whether MS-DOS subsystem is to search appended directories when executing programs. If you use the **/x:on** switch, the program does search appended directories. If you use the **/x:off** switch, the program does not search appended directories. You can abbreviate **/x:on** to **/x**. If you want to specify **/x:on**, you must do it the first time you use **append** after starting your system. After that, you can switch between **/x:on** and **/x:off**.

/path:{**on** | **off**}

Specifies whether a program is to search appended directories for a data file when a path is already included with the name of the file the program is looking for. The default setting is **/path:on**.


/e


Assigns the list of appended directories to an environment variable named **append**. This switch can be used only the first time you use **append** after starting your system. If you use **/e**, you can use the **set** command to display the list of appended directories. For information about environment variables, see the **set** command.

See Also

 [Path](#)

More Information About Append

 [Append--Notes](#)

 [Append--Examples](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Append--Notes

Storing the list of appended directories in the environment

You can use the `/e` switch with **append** to assign the list of appended directories to an environment variable named **append**. To do this, first use the **append** command with only the `/e` switch. Then use **append** again, this time including the directories you want to append. You cannot specify `/e` and `[drive:]path` on the same command line.

Specifying multiple appended directories

To append more than one directory, separate multiple entries with semicolons. If you use the **append** command with the `[drive:]path` parameters again, the specified directory or directories replace any directories specified in a previous **append** command.

Appended directories and the `dir` command

The **dir** command does not add filenames from appended directories to directory listings produced by the **dir** command.

Filename conflicts

If a file in an appended directory has the same name as a file in the current directory, programs open the file in the current directory.


Using `append` with programs that create new files


When a program opens a file in an appended directory, the file can be found as if it were in the current directory. If the program then saves the file by creating a new file with the same name, the new file is created in the current directory (not the appended directory). The **append** command is appropriately used for data files that are not to be modified or that are to be modified without creating new copies of the files. Database programs often modify data files without making new copies. Text editors and word processors, however, usually save modified data files by making new copies. To avoid confusion, you might want to avoid using **append** with these programs.

Using the `/x:on` switch and the `path` command

When `/x:on` is specified, you can run a program located in an appended directory by typing the program name at the command prompt. Usually, you use the **path** command to specify directories that contain programs. However, when your program is in an appended directory, you do not need to use the **path** command to specify that directory. Windows NT finds a program in an appended directory by following the usual order in which Windows NT searches for a program; that is, first in the current directory, then in the appended directories, and then in the search path.

More Information About Append

 [Append--Examples](#)

 [Append](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Append--Examples

To allow programs to open data files in a directory named LETTERS on the disk in drive B and in a directory named REPORTS on the disk in drive A as if the files were in the current directory, type the following command:

```
append b:\letters;a:\reports
```

To append the same directories and keep a copy of the list of appended directories in the Windows NT environment, type the following commands:

```
append /e
```


```
append b:\letters;a:\reports
```


These must be the first **append** commands you use after starting your system.

Note

Do not use the file extension .EXE when typing the **append** command.

More Information About Append

 [Append--Notes](#)

 [Append](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Assoc

Displays or modifies file extension associations.

assoc [.ext]=[*filetype*]]

Parameters

.ext

Specifies the file extension to associate with the file type.

filetype

Specifies the file type to associate with the file extension.

Type **assoc** without options to display the current file associations.

Type **assoc .ext** to display the current association for a file extension.

Type **assoc .ext=** to delete the association for a file extension.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

At

The **at** command schedules commands and programs to run on a computer at a specified time and date. The Schedule service must be running to use the **at** command.

at [*\\computername*] [*id*] [/delete [/yes]]

at [*\\computername*] *time* [/interactive] [/every:date[,...] | /next:date[,...]] "*command*"

Parameters

none

Used without parameters, **at** lists scheduled commands.

\\computername

Specifies a remote computer. If this parameter is omitted, the commands are scheduled on the local computer.

id

Is an identification number assigned to a scheduled command.

/delete

Cancels a scheduled command. If *id* is omitted, all the scheduled commands on the computer are canceled.

/yes

Forces a yes answer to all queries from the system when deleting scheduled events.

time

Specifies the time when command is to run. Time is expressed as hours:minutes in 24-hour notation (00:00 [midnight] through 23:59).

/interactive

Allows the job to interact with the desktop of the user who is logged on at the time the job runs.

/every:date[,...]

Runs the command on every specified day(s) of the week or month (for example, every Thursday, or the third day of every month). Specify date as one or more days of the week (M,T,W,Th,F,S,Su) or one or more days of the month (using numbers 1 through 31). Separate multiple date entries with commas. If date is omitted, the current day of the month is assumed.


/next:date[,...]


Runs the specified command on the next occurrence of the day (for example, next Thursday). Specify date as one or more days of the week (M,T,W,Th,F,S,Su) or one or more days of the month (using numbers 1 through 31). Separate multiple date entries with commas. If date is omitted, the current day of the month is assumed.

"command"

Is the Windows NT command, program (.EXE or .COM file) or batch program (.BAT or .CMD file) to be run. When the command requires a path as an argument, use the absolute path, that is, the entire pathname beginning with the drive letter. If command is on a remote computer, specify the server and sharename, rather than a remote drive letter. You may use quotation marks around the command, whether you are using **at** at the command line or in a batch file. If the command includes switches that are used by both the command and **at**, you must enclose command in quotation marks. If the command is not an executable (.EXE) file, you must precede the command with **cmd /c**; for example **cmd /c dir > c:\test.out**.

More Information About At

 [At--Notes](#)

 [At--Examples](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

At--Notes

The **at** command requires you be a member of the local Administrator group. Also, the **at** command does not automatically load **cmd**, the command interpreter, before running commands. Unless you are running an executable (.EXE) file, you must explicitly load CMD.EXE at the beginning of the command; for example, **cmd /c dir > c:\test.out**.

Viewing Scheduled Commands

The **at** command has two displays. When you use **at** without options, you see a display similar to the following:

STATUS	ID	Day	Time	Command Line
	0	Each F	04:30PM	net send group leads status due
	2	Each M	12:00AM	chkstor > check.fil
	3	Each F	11:59PM	backup2.bat

When you include the identification number for the command, the display provides information for a single entry and is similar to the following:

```
Task ID:      0
Status:      OK
Schedule:    Each F
Time of Day: 04:30PM
Command:     net send group leads status due
```

After you schedule a command with **at** (especially a command that has option variables), check that the command is entered properly by typing **at** without options. If the information in the "Command Line" column isn't correct, delete the command and retype it. If it still isn't correct, retype the command with fewer option variables.

Viewing Results

Commands scheduled with the **at** command run as background processes, so no output is displayed on the computer's screen. To redirect output to a file, use the redirection symbol (>). If you redirect output to a file, enclose the name of the command you are scheduling in quotation marks. The current directory for the executing command is the %systemroot% directory.

Changing System Time

If you change the system time at a computer after scheduling a command to run with **at**, synchronize the **at** scheduler with the revised time by typing **at** without options.

Storing Commands

Scheduled commands are stored in the registry, so scheduled tasks are not lost if you have to restart the Schedule service.

Connecting to Network Drives

Scheduled jobs that access the network should not use redirected drives made by the user. The Schedule service may not be able to access these drives and they may not be present if a different user is logged on at the time the scheduled job runs. Instead, scheduled jobs should use UNC paths. For example,

```
at 1:00pm my_backup \\server\share
```

and not


```
at 1:00pm my_backup x:
```

where x: is a connection made by the user.

If you schedule an **at** command that uses a drive letter to connect to a shared directory, you should include an **at** command to disconnect the drive when you are finished using it. If the drive is not disconnected, the drive letter assigned will not be available or seen at the command prompt.

More Information About At

 [At](#)

 [At--Examples](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

At--Examples

To display a list of commands scheduled on the server MARKETING, type

at \\marketing

To learn more about a command with the identification number 3 on the server CORP, type

at \\corp 3

To schedule a net share command to run on the CORP server at 8:00 A.M., and redirect the listing to the server MAINTENANCE, shared directory REPORTS, and file CORP.TXT, type

at \\corp 08:00 "cmd /c net share reports=d:\marketing\reports >> \\maintenance\reports\corp.txt"


To back up the MARKETING server's hard disk to a tape drive at midnight every five days, create a batch program (ARCHIVE.CMD) containing the backup commands. Then schedule the batch program to run by typing

at \\marketing 00:00 /every:5,10,15,20,25,30 archive

To cancel all commands scheduled on the current server, clear the **at** schedule information by typing

at /delete

More Information About At

 [At--Notes](#)

 [At](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Attrib

Displays or changes file attributes.

This command displays, sets, or removes the read-only, archive, system, and hidden attributes assigned to files or directories.

attrib [+r|-r] [+a|-a] [+s|-s] [+h|-h][[drive:][path] filename] [/s]

Parameters

[[drive:][path] filename]

Specifies the location and name of the directory, file, or set of files you want to process.

+r

Sets the read-only file attribute.

-r

Clears the read-only file attribute.

+a

Sets the archive file attribute.

-a

Clears the archive file attribute.

+s

Sets the file as a system file.

-s

Clears the system file attribute.

+h

Sets the file as a hidden file.

-h


Clears the hidden file attribute.

/s


Processes files in the current directory and all of its subdirectories.


See Also

 [Copy](#)

 [Xcopy](#)

More Information About Attrib

 [Attrib--Notes](#)

 [Attrib--Examples](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Attrib--Notes


Groups of files

You can use wildcards (? and *) with the *filename* parameter to display or change the attributes for a group of files. If a file has the system or hidden attribute set, you must clear that attribute before you can change any other attributes for that file.

Archive attributes

The archive attribute (**+a**) is used to mark files that have changed since they were previously backed up. The **backup**, **restore** and **xcopy** commands use these archive attributes. For information about archive attributes, see the **backup**, **restore** and **xcopy** commands.

More Information About Attrib

 [Attrib--Examples](#)

 [Attrib](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Attrib--Examples

To display the attributes of a file named NEWS86 located on the current drive, type the following command:

```
attrib news86
```

To assign the read-only attribute to the file REPORT.TXT, type the following command:

```
attrib +r report.txt
```

To remove the read-only attribute from files in the \PUBLIC\JONES directory on a disk in drive B and from files in any subdirectories of \PUBLIC\JONES, type the following command:

```
attrib -r b:\public\jones\*.*/s
```

As a final example, suppose you want to give an associate a disk containing all files in the default directory on a disk in drive A except files with the .BAK extension. Because you can use **xcopy** to copy only those files marked with the archive attribute, you need to set the archive attribute for those files you want to copy. To do this, you use the following two commands to set the archive attribute for all files on drive A and then to clear the attribute for those files with the .BAK extension:

```
attrib +a a:*.*
```

```
attrib -a a:*.bak
```


Next, use the **xcopy** command to copy the files from the disk in drive A to the disk in drive B. The **/a** switch in the following command causes **xcopy** to copy only those files marked with the archive attribute:

```
xcopy a: b: /a
```

If you want **xcopy** to clear each file's archive attribute after it copies the file, use the **/m** switch instead of **/a**, as in the following example:

```
xcopy a: b: /m
```

More Information About Attrib

 [Attrib--Notes](#)

 [Attrib](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Backup

Backs up one or more files from one disk onto another.

You can back up files onto either a hard disk or floppy disk(s). Files can also be backed up from one floppy disk onto another, even if the disks have different numbers of sides or sectors. Windows NT displays the name of each file it backs up.

backup *source destination-drive: [/s] [/m] [/a][/f[:size]] [/d:date [/t:time]][/l[:drive:][path]logfile]*

Parameters

source

Specifies the location of files you want to back up. Source can consist of a drive letter and colon, a directory name, a filename, or a combination.

destination-drive:

Specifies the drive that contains the disk on which you want to store any backup files. The backup files are stored in the *BACKUP.nnn* and *CONTROL.nnn* files. That is, **backup** assigns the names *BACKUP.001* and *CONTROL.001* to the files it creates on the first backup disk you use, *BACKUP.002* and *CONTROL.002* to the files it creates on the second backup disk, and so on.

/s

Backs up the contents of all subdirectories.

/m

Backs up only files that have changed since the last backup, and turns off the archive attribute of the original files.

/a

Adds backup files to an existing backup disk without deleting existing files. (The **/a** switch is ignored if the existing backup disk contains backup files that were created by using the **backup** command from MS-DOS version 3.2 or earlier.)

/f[:size]

Formats the backup disk to the size you specify. (The **format** command must be present in the current path.) With this switch, you direct **backup** to format floppy disks that do not match the default size of the drive. The **backup** command formats an unformatted destination disk even if you do not specify the **/f** switch. When **backup** finishes formatting, it begins backing up files onto the last disk it formatted. *Size* specifies the size in kilobytes of the disk to be formatted. If you do not specify size, the **/f** switch uses the default size of the drive. The following list shows the valid values for size and a brief description of each size:

160 or 160k or 160kb 160K, single-sided, double-density, 5.25-inch disk

180 or 180k or 180kb 180K, single-sided, double-density, 5.25-inch disk

320 or 320k or 320kb 320K, double-sided, double-density, 5.25-inch disk

360 or 360k or 360kb 360K, double-sided, double-density, 5.25-inch disk

720 or 720k or 720kb 720K, double-sided, double-density, 3.5-inch disk

1200 or 1200k or 1200kb or 1.2 or 1.2m or 1.2mb 1.2-MB, double-sided, quadruple-density, 5.25-inch disk

1440 or 1440k or 1440kb or 1.44 or 1.44m or 1.44mb 1.44-MB, double-sided, quadruple-density, 3.5-inch disk

2880 or 2880k or 2880kb or 2.88 or 2.88m or 2.88mb 2.88-MB, double-sided, 3.5-inch disk

/d:date

Backs up only files modified on or after the specified date. The date format depends on the setting you are using

for the **country** command.

/t:time


Backs up only files modified at or after the specified time. The time format depends on the setting you are using for the **country** command.

/l:[drive:][path]logfile]

Creates a log file and adds an entry to that file to record the backup operation. If you do not specify a location for the log file, **backup** puts the file in the root directory of the source drive. If you do not specify logfile, **backup** names the file BACKUP.LOG. You should not specify a removable drive (such as a floppy disk drive) for this parameter; but once the backup is complete, you can copy the log file to a floppy disk.

More Information About Backup

 [Backup--Notes](#)

 [Backup--Examples](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Backup--Notes

Backing up onto a disk with files

Unless you use the **/a** switch, **backup** deletes old files (including read-only files) from a backup disk before adding new files to it.

Backup log file

If you use the **/l** switch and do not specify a name and location for the log file, the **backup** command adds a file named BACKUP.LOG to the root directory of the source drive. If the BACKUP.LOG file already exists, **backup** adds the current entry to the file. A backup log-file entry uses the following format:

- The date and time of the backup appear on the first line.
- Each filename appears on a separate line with the number of the backup disk that contains the file.

The backup log file can assist you later, when you need to identify the files you want to restore. The **restore** command always returns a file to the original directory or subdirectory recorded in the backup log, creating the subdirectory if necessary.

Labeling backup disks

It is important to label and number backup disks consecutively. As each disk is filled, **backup** prompts you for the next disk. When you restore files, you need to insert the backup disks into the disk drive in the same sequence. To check the sequence of backup disks (MS-DOS version 3.3 or later), use the **dir** command to check the disk number.

Backup and system files

The **backup** command cannot back up the system files IO.SYS, MSDOS.SYS, and CMD.EXE. You can use the **sys** command to copy these files onto a floppy disk.

Using an old version of the restore command

You cannot use an old version of the **restore** command (MS-DOS version 3.2 or earlier) for files backed up with MS-DOS version 3.3 or later. If you attempt this, Windows NT displays the following message:

Source does not contain backup files

This error occurs because the format of old backup files differs from the format of files backed up with MS-DOS versions 3.3 and later.

Backup exit codes


The following list shows each exit code and given a brief description of its meaning:

- 0 The backup was successful.
- 1 No files were found to back up.
- 2 Some files were not backed up because of file-sharing conflicts.
- 3 The user pressed CTRL+C to stop the process.
- 4 The process stopped because of an error.

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned

by the **backup** command. For an example of a batch program that processes exit codes, see [Backup--Examples](#).

More Information About Backup

 [Backup--Examples](#)

 [Backup](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Backup--Examples

Suppose you want to back up all the files in the \PUBLIC\SMITH directory on drive C onto a blank, formatted disk in drive A. To do so, type the following:

```
backup c:\public\smith\*. * a:
```

Suppose you need to back up all files in the \PUBLIC\SMITH directory on drive C onto a 720K floppy disk in drive B. If the floppy disk is unformatted, **backup** formats it before backing up any files. Because the **/s** switch is not specified in the following command, files in subdirectories are not backed up:


```
backup c:\public\smith\*. * b: /f:720k
```

To write a simple batch program named SMITH that supports the **backup** command's exit codes and the **/s** switch, create a file containing the following commands using any editor.

```
echo off
rem Smith's backup command
backup c:\public\smith\*. * b: /s
if errorlevel 4 goto error
if errorlevel 3 goto abort
if errorlevel 2 goto conflict
if errorlevel 1 goto no_files
if errorlevel 0 goto success
:error
echo Backup stopped the process due to an error
goto exit
:abort
echo You just pressed CTRL+C to stop the backup
goto exit
:conflict
echo One or more files were not backed up due to a sharing conflict
goto exit
:no_files
echo Sorry, but there were no files to backup
goto exit
:success
echo The backup was successful
goto exit
:exit
```

For more information about using the **if** command in batch programs, see the **if** command.

More Information About Backup

 [Backup--Notes](#)

 [Backup](#)

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Break

Windows NT does not use this command. It is accepted only for compatibility with files from MS-DOS.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Buffers

Windows NT and the MS-DOS subsystem do not use this command. It is accepted only for compatibility with files from MS-DOS.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Cacls

Displays or modifies access control lists (ACLs) of files.

cacls *filename* [/t] [/e] [/c] [/g *user:perm*] [/r *user* [...]] [/p *user:perm* [...]] [/d *user* [...]]

Parameters

filename

Displays ACLs of specified file or files.

/t

Changes ACLs of specified files in the current directory and all subdirectories.

/e

Edit ACL instead of replacing it.

/c

Continue changing ACLs, ignoring errors.

/g *user:perm*

Grant specified user access rights. *Perm* can be:

r Read

c Change (write)

f Full control

/r *user*

Revoke specified user's access rights.

/p *user:perm*

Replace specified user's access rights. *Perm* can be:

n None

r Read

c Change (write)

f Full control

/d *user*

Deny specified user access.

You can specify more than one file or user in a command.



Call

Calls one batch program from another without causing the parent batch program to stop.

call [*drive:*][*path*] *filename* [*batch-parameters*]

Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of the batch program you want to call. *Filename* must have a .BAT or .CMD extension.

batch-parameters

Specifies any command-line information required by the batch program.

More Information About Call



[Call--Notes](#)



[Call--Examples](#)



Call--Notes

Using batch-parameters

Batch-parameters can contain any information that you can pass to a batch program, including switches, filenames, the replaceable parameters %1 through %9, and variables such as %baud%.

Using pipes and redirection symbols

Do not use pipes and redirection symbols with the **call** command.

Making a recursive call

You can create a batch program that calls itself; however, you must provide an exit condition. Otherwise, the parent and child batch programs can loop endlessly.

More Information About Call



[Call--Examples](#)



[Call](#)



Call--Examples

To run the CHECKNEW.BAT program from another batch program, include the following command in the parent batch program:

call checknew

Suppose the parent batch program accepts two replaceable parameters and you want it to pass those parameters to CHECKNEW.BAT. You can use the following command in the parent batch program:

```
call checknew %1 %2
```

More Information About Call



[Call--Notes](#)



[Call](#)



Chcp

Displays the number of the active console code page, or changes the active console code page that Windows NT is to use for the console. Note: Only the OEM code page installed with Windows NT will display correctly in a command prompt window using Raster fonts. Other code pages will display correctly in full-screen mode or command prompt windows that are using TrueType fonts.

chcp [*nnn*]

Parameter

none

Used without parameters, **chcp** displays the number of the active console code page.

nnn

Specifies the code page. The following list shows each code page that Windows NT supports and its country or language:

437	United States
850	Multilingual (Latin I)
852	Slavic (Latin II)
855	Cyrillic (Russian)
857	Turkish
860	Portuguese
861	Icelandic
863	Canadian-French
865	Nordic
866	Russian
869	Modern Greek

See Also

[Country](#)

[Device](#)

[Devicehigh](#)

[Mode \(set device code pages\)](#)

More Information About Chcp

[Chcp--Notes](#)

[Chcp--Examples](#)



Chcp--Notes

Code pages need not be prepared

Code pages do not need to be prepared, as in MS-DOS.

Assigning a new code page

After assigning a new code page, programs you start will use that new code page. However, any program (except CMD.EXE) that you started before assigning the new code page will probably attempt to use the original code page.

More Information About Chcp



[Chcp--Examples](#)



[Chcp](#)



Chcp--Examples

To view the active code-page setting, type the following command:

chcp

Windows NT responds with a message similar to the following:

Active code page: 437

To change the active code page to 850 (Multilingual), type the following command:

chcp 850

Windows NT alerts you if the specified code page is invalid. The following error message appears:

Invalid code page

More Information About Chcp



[Chcp--Notes](#)



[Chcp](#)



Chdir (Cd)

Displays the name of the current directory or changes the current directory.

chdir [/d] [drive:][path] [..]

-or-

cd [/d] [drive:][path] [..]

Parameters

none

Used without parameters, **chdir** displays the names of the current drive and directory.

[/d]

Changes the current drive in addition to changing the current directory for a drive.

[drive:][path]

Specifies the drive (if other than the current drive) and directory to which you want to change.

[..]

Specifies that you want to change to the parent directory.

More Information About Chdir



[Chdir--Notes](#)



[Chdir--Examples](#)



Chdir--Notes

Changing to the root directory

The root directory is the top of the directory hierarchy for a drive. To return to the root directory, type the following command:

```
cd\
```

Using the current directory from a different drive

If you are working in the \PUBLIC\JONES directory on drive C and you change to drive D, you can copy files to and from the \PUBLIC\JONES directory by specifying only the drive letter C.

Changing the directory on another drive

You can change the current directory on another drive by specifying the drive name on the command line when you use **chdir**.

More Information About Chdir



[Chdir--Examples](#)



[Chdir](#)



Chdir--Examples

Either of the following commands changes your current directory to the directory named REPORTS:

```
chdir \reports  
cd \reports
```

Suppose you have a directory named SPECIALS with a subdirectory named SPONSORS. To change your current directory to \SPECIALS\SPONSORS, type the following command:

```
cd \specials\sponsors
```

Or, if your current directory is \SPECIALS, you can use the following command to change to the \SPECIALS\SPONSORS directory:

```
cd sponsors
```

To change from a subdirectory back to the next higher directory, type the following command:

```
cd ..
```

To display the name of the current directory, you can use **chdir** or **cd** without a parameter. For example, if your current directory is \PUBLIC\JONES on the disk in drive B, type **chdir** to see the following response:

```
B:\PUBLIC\JONES
```


If you are working on drive D and you want to copy all files in the \PUBLIC\JONES and \PUBLIC\LEWIS directories on drive C to the root directory on drive D, type the following commands:

```
chdir c:\public\jones  
copy c:*. * d:\  
chdir c:\public\lewis  
copy c:*. * d:\
```

If, instead, you want to copy all files in the \PUBLIC\JONES and \PUBLIC\LEWIS directories to your current location on drive D, type the following commands:

```
chdir c:\public\jones  
copy c:*. * d:  
chdir c:\public\lewis  
copy c:*. * d:
```

More Information About Chdir

 [Chdir--Notes](#)

 [Chdir](#)



Chkdsk

Creates and displays a status report for a disk based on the file system used. **Chkdsk** also lists and corrects errors on the disk. If **chkdsk** cannot lock the drive it will offer to check it the next time the computer reboots. Issuing the **chkdsk** command on a fixed disk requires you be a member of the Administrators group.

chkdsk [*drive:*][*[path] filename*] [**/f**] [**/v**] [**/r**]

Parameters

none

Used without parameters, **chkdsk** displays the status of the disk in the current drive.

drive:

Specifies the drive that contains the disk that you want **chkdsk** to check.

[path] filename

Specifies the location and name of a file or set of files that you want **chkdsk** to check for fragmentation. You can use wildcards (* and ?) to specify multiple files.

/f

Fixes errors on the disk. The disk must be locked. If **chkdsk** cannot lock the drive it will offer to check it the next time the computer reboots.

/v

Displays the name of each file in every directory as the disk is checked.

/r

Locates bad sectors and recovers readable information. The disk must be locked.

More Information About Chkdsk



[Chkdsk--Notes](#)



[Chkdsk--Examples](#)



Chkdsk--Notes

Chkdsk examines disk space and use for the NTFS and FAT file systems. Information specific to each file system is provided in a status report. The status report shows errors found in the file system.

If errors exist on the disk, **chkdsk** alerts you with a message and corrects the errors if the **/f** switch was used.

Chkdsk must be able to lock the drive to correct errors. If errors are found but the drive cannot be locked, an error message is displayed. If you run **chkdsk** without the **/f** switch on an active partition, it may report spurious errors, since it will not lock the volume.

You should use **chkdsk** occasionally on each disk to check for errors.

Checking a locked drive at reboot

Files cannot be open on a drive when **chkdsk** corrects disk errors it finds. If files are open **chkdsk** offers to check the drive the next time the computer reboots. If you choose to check the drive the next time the computer reboots, the drive will be checked and errors will be corrected automatically at reboot. If the drive partition is a boot partition, upon completion it will reboot the computer.

Checking a FAT disk

Windows NT displays **chkdsk** status reports for a FAT disk in the following format:

```
Volume Serial Number is B1AF-AFBF
 72214528 bytes total disk space
   73728 bytes in 3 hidden files
   30720 bytes in 12 directories
11493376 bytes in 386 user files
   61440 bytes in bad sectors
60555264 bytes available on disk
   2048 bytes in each allocation unit
   35261 total allocation units on disk
   29568 available allocation units on disk
```

Checking an NTFS disk

Windows NT displays **chkdsk** status reports for an NTFS disk in the following format:

The type of the file system is NTFS.

CHKDSK is verifying files...

File verification completed.

CHKDSK is verifying indexes...

Index verification completed.

CHKDSK is verifying security descriptors...

Security descriptor verification completed.

```
 12372 kilobytes total disk space.
    3 kilobytes in 1 user files.
    2 kilobytes in 1 indexes.
 4217 kilobytes in use by the system.
 8150 kilobytes available on disk.
   512 bytes in each allocation unit.
24745 total allocation units on disk.
16301 allocation units available on disk.
```

Fixing disk errors

The **chkdsk** command corrects disk errors only if you specify the **/f** switch. Since repairs usually change a disk's file allocation table and sometimes cause a loss of data, **chkdsk** first prompts you with a confirmation message similar to the following:

10 lost allocation units found in 3 chains.

Convert lost chains to files?

If you press **Y**, Windows NT saves each lost chain in the root directory as a file with a name in the format **FILEnnnn.CHK**. When **chkdsk** finishes, you can check these files to see if they contain any data you need. If you press **N**, Windows NT fixes the disk but does not save the contents of the lost allocation units.

If you do not use the **/f** switch, **chkdsk** alerts you with a message if a file needs to be fixed but does not fix the error(s).

If you use **chkdsk /f** on a very large disk (for example, 70 GB) or a disk with a very large number of files (millions), **chkdsk** may take a long time (perhaps days) to complete. The computer will not be available during this time, as **chkdsk** does not relinquish control until it is done.

Using chkdsk with open files

If you specify the **/f** switch, **chkdsk** shows an error if open files are found on the disk. If you do not specify the **/f** switch and open files exist, **chkdsk** might report lost allocation units on the disk. This could happen if open files have not yet been recorded in the file allocation table. If **chkdsk** reports the loss of a large number of allocation units, consider repairing the disk.

Physical disk errors

Use the **/r** switch to find physical disk errors in the file system. For information about recovering physically damaged files, see the **recover** command.


Bad disk sectors


Bad sectors reported by **chkdsk** were marked as "bad" when your disk was first prepared for operation. They pose no danger.

Exit codes

- 0 No errors were found
- 1 Errors were found and fixed.
- 2 Could not check the disk, errors could not be fixed, or errors were not fixed because **/f** was not specified.

More Information About Chkdsk

 [_Chkdsk--Examples](#)

 [_Chkdsk](#)



Chkdsk--Examples

If you want to check the disk in drive D and have Windows NT fix any errors encountered, type the following command:

chkdsk d: /f

Chkdsk pauses and displays messages if it encounters errors. **Chkdsk** finishes by displaying a report showing the status of the disk. Also, no files can be open on the drive specified.

On a FAT disk, to check all files in the current directory for non-contiguous blocks, type

chkdsk *.*

Chkdsk displays a status report, then lists the files matching the file specification that have non-contiguous blocks.

More Information About Chkdsk



[Chkdsk--Notes](#)



[Chkdsk](#)



Cls

Clears the screen.

The cleared screen shows only the command prompt and cursor.

cls



Cmd

Starts a new instance of the Windows NT command interpreter, CMD.EXE.

A command interpreter is a program that displays the command prompt at which you type commands. Use the **exit** command to stop the new command interpreter and return control to the old one.

cmd [/x | /y] [/a | /u] [/q] [/t:fg] [[/c | /k] *string*]

Parameters

/c

Carries out the command specified by *string* and then stops.

/k

Carries out the command specified by *string* and continues.

/q

Turns the echo off.

/a

Creates ANSI output.

/u

Creates Unicode output.

/t:fg

Sets the foreground and background colors. (For more information, see the **color** command.)

/x

Enables extensions to the Windows NT version of CMD.EXE, to provide a richer shell programming environment. The following commands use the extensions: **del** (**erase**), **color**, **cd** (**chdir**), **md** (**mdir**), **prompt**, **pushd**, **popd**, **set**, **setlocal**, **endlocal**, **if**, **for**, **call**, **shift**, **goto**, **start**, **assoc**, and **ftype**. For details, see the Help for each command.

/y

Disables extensions to the Windows NT version of CMD.EXE, for backward compatibility reasons. The extensions are enabled by default.

string

Specifies the command you want to carry out.



Codepage

Selects the code pages that the system will use for the MS OS/2 subsystem. To use this command, place it in your OS/2 C:\CONFIG.SYS file.

codepage=xxx[,yyy]

Parameters

xxx

Specifies the first code page. This must be a three-digit number from this list:

437	United States
850	Multilingual (Latin I)
852	Slavic (Latin II)
855	Cyrillic (Russian)
857	Turkish
860	Portuguese
861	Icelandic
863	Canadian-French
865	Nordic
866	Russian
869	Modern Greek

yyy

This parameter is not used by the OS/2 subsystem. It is accepted only for compatibility with files from MS OS/2 version 1.3 or earlier.

More Information About Codepage



[Codepage--Notes](#)



[Codepage--Examples](#)



Codepage--Notes

Codepage, **keyboard**, and **country** are interrelated. A code page is a set of characters that are available to your system for use on the screen, for printing, and for sending to any other sort of output device. Your keyboard layout tells your system which characters from the character set correspond to which keystrokes; this can be different from country to country. The country you are working in (or for which you want to set up your system) determines which two code pages you should use. (For a list of country codes and their corresponding code pages, see the **country** command.) You can change keyboard layouts without having to change code pages. These are the available code pages:

437	United States
850	Multilingual (Latin I)
852	Slavic (Latin II)
855	Cyrillic (Russian)
857	Turkish
860	Portuguese
861	Icelandic
863	Canadian-French
865	Nordic
866	Russian
869	Modern Greek

If you are setting up a keyboard or screen to use code pages, you must also use the **devinfo** command to specify the code pages the outside device is to use. For more information, see the [devinfo](#) command.

More Information About Codepage



[Codepage--Examples](#)



[Codepage](#)



Codepage--Examples

To set up your MS OS/2 subsystem to use code pages 437 and 850, include the following line in your C:\CONFIG.SYS file:

codepage=437,850

More Information About Codepage



[Codepage--Notes](#)



[Codepage](#)



Color

Sets the default console foreground and background colors.

color *bf*

Parameters

bf

Specifies color attributes of console output. *b* is a hex digit that specifies the background color; *f* specifies the foreground. Hex digits have the following values:

0	Black
1	Blue
2	Green
3	Aqua
4	Red
5	Purple
6	Yellow
7	White
8	Gray
9	Light blue
A	Light green
B	Light aqua
C	Light red
D	Light purple
E	Light yellow
F	Bright white

For example, **color fc** produces light red on bright white.

If no argument is given, this command restores the color to what it was when CMD.EXE started.

If the foreground and background values are the same, **color** returns ERRORLEVEL 1.



Comp

Compares the contents of two files or sets of files byte by byte.

Comp can compare files on the same drive or on different drives, in the same directory or in different directories. As **comp** compares the files, it displays their locations and filenames.

comp [*data1*] [*data2*] [/d] [/a] [/l] [/n=*number*] [/c]

Parameters

data1

Specifies the location and name of the first file or set of files you want to compare. You can use wildcards (* and ?) to specify multiple files.

data2

Specifies the location and name of the second file or set of files you want to compare. You can use wildcards (* and ?) to specify multiple files.

/d

Displays differences in decimal format. (The default format is hexadecimal.)

/a

Displays differences as characters.

/l

Displays the number of the line on which a difference occurs, instead of displaying the byte offset.

/n=*number*

Compares the first *number* of lines of both files, even if the files are different sizes.

/c


Performs a comparison that is not case-sensitive.


See Also

 [Diskcomp](#)

 [Fc](#)

More Information About Comp

 [Comp--Notes](#)

 [Comp--Examples](#)



Comp--Notes

Comparing files with the same names

The files you want to compare can have the same filename, provided they are in different directories or on different drives. If you do not specify a filename for *data2*, the default filename for *data2* is the same as the filename in *data1*. You can use wildcards (* and ?) to specify filenames.

Special cases for *data1* and *data2*

If you omit necessary components of either *data1* or *data2* or if you omit *data2*, **comp** prompts you for the missing information. If *data1* contains only a drive letter or a directory name with no filename, **comp** compares all the files in the specified directory to the file specified in *data2*. If *data2* contains only a drive letter or a directory name, the default filename for *data2* is the same as that in *data1*.

How the **comp** command identifies mismatching information

During the comparison, **comp** displays messages to identify the locations of unequal information in the two files. Each message indicates the offset memory address of the unequal bytes and the contents of the bytes themselves (in hexadecimal notation unless you specify the **/a** or **/d** switch). The message has the following format:

```
Compare error at OFFSET xxxxxxxx
file1 = xx
file2 = xx
```

After 10 unequal comparisons, **comp** stops comparing the files and displays the following message:

```
10 Mismatches - ending compare
```

Comparing files of different sizes

You cannot compare files of different sizes unless you specify the **/n** switch. If the file sizes are different, **comp** displays the following message:

```
Files are different sizes
Compare more files (Y/N)?
```

Press **Y** to compare another pair of files. Press **N** to stop the **comp** command.

If you press **Y** in response to the prompt, **comp** includes any switches you specified on the command line in every comparison it makes, until you press **N** or retype the command.

When comparing files of different sizes, use the **/n** switch to compare only the first portion of each file.

Comparing files sequentially

If you use wildcards to specify multiple files, **comp** finds the first file matching *data1* and compares it with the corresponding file in *data2*, if it exists. The **comp** command reports the results of the comparison, then does the same for each file matching *data1*. When finished, **comp** displays the following message:

```
Compare more files (Y/N)?
```

To compare more files, press Y. The **comp** command prompts you for the locations and names of the new files. To stop the comparisons, press N. When you press Y, **comp** prompts you for switches to use. If you don't specify any switches, **comp** uses the ones you specified before.

If **comp** cannot find the files

If **comp** cannot find the file(s) you specify, it prompts you with a message to determine whether you want to compare more files.

More Information About Comp



[Comp--Examples](#)



[Comp](#)



Comp--Examples

To compare the contents of the directory C:\REPORTS with the backup directory \\SALES\BACKUP\APRIL, type

```
comp c:\reports \\sales\backup\april
```

To compare the first ten lines of the text files in the \INVOICE directory and display the result in decimal format, type

```
comp \invoice\*.txt \invoice\backup\*.txt /n=10 /d
```

More Information About Comp



[Comp--Notes](#)



[Comp](#)



Compact

Displays and alters the compression of files or directories

compact [/c] [/u] [/s] [/i] [/f] [/l] *filename*

Parameters

none

Used without parameters, **compact** displays the compression state of the current directory.

/c

Compresses the specified directory or file.

/u

Uncompresses the specified directory or file.

/s:directory

Specifies that the requested action (compress or uncompress) be applied to all subdirectories of the specified directory, or of the current directory if none is specified.

/i

Ignores errors.

/f

Forces compression or uncompression of the specified directory or file. This is used in the case of a file that was partly compressed when the operation was interrupted by a system crash. Use the /c and /f parameters and specify the partially compressed file to force the file to be compressed in its entirety.

filename

Specifies the file or directory. You can use multiple filenames and wildcards

More Information About Compact



[Compact--Notes](#)



[Compact--Examples](#)



Compact--Notes

Compact, the command line version of Windows NT's compression feature, displays and alters the compression attribute of files and directories on NTFS partitions. The compression state of a directory indicates whether files added to the directory will be automatically compressed; when you set the compression state of a directory you do not necessarily change the compression state of files that are already there.

You cannot use compact to read, write, or mount volumes that have been compressed using DriveSpace or DoubleSpace.

More Information About Compact



[Compact](#)



[Compact--Examples](#)



Compact--Examples

To compress the files in the current directory and all subdirectories, and set the compressed attribute on the current directory and all subdirectories, type

```
compact /c /s
```

To compress all files that end in .BMP in the \TMP directory and all subdirectories of \TMP, but not modify the compressed attribute of these directories, type

```
compact /c /s:\tmp *.bmp
```

To force complete compression of the file ZEBRA.BMP, which was partially compressed at the time of a system crash, type

```
compact /c /f zebra.bmp
```

To remove the compressed attribute from the directory C:\TMP, but not change the compression state of any files in that directory, type

```
compact /u c:\tmp
```

More Information About Compact



[Compact](#)



[Compact--Notes](#)



Convert

Converts FAT volumes to NTFS. You cannot convert the current drive. If **convert** cannot lock the drive it will offer to convert it the next time the computer reboots.

convert [*drive:*] **/fs:ntfs** [**/v**] [**/nametable:filename**]

Parameters

drive

Specifies the drive to convert to NTFS.

/fs:ntfs

Specifies to convert the volume to NTFS.

/v

Specifies verbose mode. All messages will be displayed during conversion.

/nametable:filename

Creates a name-translation table in the root directory of the converted volume using the specified filename. Use this switch if you encounter difficulty converting files with unusual filenames.

More Information About Convert



[Convert--Notes](#)



[Convert--Examples](#)



Convert--Notes

If an automatic conversion fails at boot, look for details in Event Viewer in the application log under WINLOGON. The application log will list the cause of the convert failure. If the conversion fails due to difficulty in converting files with strange filenames, reissue the **convert** command using the **/nametable** switch and attempt automatic conversion. The system will consult the nametable created by **convert** during the automatic conversion.

More Information About Convert



[Convert](#)



[Convert--Examples](#)



Convert--Examples

To convert the volume on drive E and display all messages, type

```
convert e: /fs:ntfs /v
```

To convert the volume on drive E and create the nametable file CONVFILE.LOG in the root directory of the converted volume, type

```
convert e: /fs:ntfs /nametable:convfile.log
```

More Information About Convert



[Convert](#)



[Convert--Notes](#)



Copy

Copies one or more files to another location.

This command can also be used to combine files. When more than one file is copied, Windows NT displays each filename as the file is copied.

copy [/a]/b] source [/a]/b] [+ source [/a]/b] [+ ...] [destination [/a]/b]] [/v] [/n] [/z]

Parameters

source

Specifies the location and name of a file or set of files from which you want to copy. *Source* can consist of a drive letter and colon, a directory name, a filename, or a combination.

destination

Specifies the location and name of a file or set of files to which you want to copy. *Destination* can consist of a drive letter and colon, a directory name, a filename, or a combination.

/a

Indicates an ASCII text file. When the **/a** switch precedes the list of filenames on the command line, it applies to all files whose names follow the **/a** switch, until **copy** encounters a **/b** switch, in which case the **/b** switch applies to the file whose name precedes the **/b** switch.

When the **/a** switch follows a filename, it applies to the file whose name precedes the **/a** switch and to all files whose names follow the **/a** switch, until **copy** encounters a **/b** switch, in which case the **/b** switch applies to the file whose name precedes the **/b** switch. An ASCII text file can use an end-of-file character (CTRL+Z) to indicate the end of the file. When combining files, **copy** treats files as ASCII text files by default.

/b

Indicates a binary file. When the **/b** switch precedes the list of filenames on the command line, it applies to all files whose names follow the **/b** switch, until **copy** encounters an **/a** switch, in which case the **/a** switch applies to the file whose name precedes the **/a** switch.

When the **/b** switch follows a filename, it applies to the file whose name precedes the **/b** switch and to all files whose names follow the **/b** switch, until **copy** encounters an **/a** switch, in which case the **/a** switch applies to the file whose name precedes the **/a** switch.

The **/b** switch specifies that the command interpreter is to read the number of bytes specified by the file size in the directory. The **/b** switch is the default value for **copy** unless **copy** is combining files.

/v

Verifies that new files are written correctly.

/n

Uses a short filename, if available, when copying a file with a non-8dot3 name.

/z

Copies over a network in restartable mode.

See Also



[Xcopy](#)

More Information About Copy



[Copy--Notes](#)



[Copy--Examples](#)



Copy--Notes

Copying to and from devices

You can substitute a device name for one or more occurrences of *source* or for *destination*.

Using or omitting the /b switch when copying to a device

When *destination* is a device (for example, COM1 or LPT1), the **/b** switch causes Windows NT to copy data to the device in binary mode. In binary mode, all characters (including such special characters as CTRL+C, CTRL+S, CTRL+Z, and carriage return) are copied to the device as data. Whereas, omission of the **/b** switch causes Windows NT to copy data to the device in ASCII mode. In ASCII mode, such special characters as those previously listed may cause Windows NT to take special action during the copying process, such as

Using the default destination file

If you do not specify a destination file, Windows NT creates a copy with the same name, creation date, and creation time as the original file, placing the new copy in the current directory on the current drive. If the source file is on the current drive and in the current directory and you do not specify a different drive or directory for the destination file, the **copy** command stops and Windows NT displays the following error message:

```
File cannot be copied onto itself
0 File(s) copied
```

Using the /v switch

If Windows NT cannot verify a write operation, it displays an error message. Although recording errors rarely occur with the **copy** command, the **/v** switch lets you verify that critical data has been correctly recorded. The **/v** switch also slows down the **copy** command, because Windows NT must check each sector recorded on the disk.

Using the /a and /b switches

The effect of an **/a** or **/b** switch depends upon its position on the command line. When the **/a** or **/b** switch follows the source filename, **copy** performs as shown in the following list:

/a Treats the file as an ASCII (text) file and copies data that precedes the first end-of-file character. **Copy** does not copy the first end-of-file character or the remainder of the file.

/b Copies the entire file, including any end-of-file character.

When the **/a** or **/b** switch follows the destination filename, **copy** performs as shown in the following list:

/a Adds an end-of-file character as the last character of the file.

/b Does not add an end-of-file character.

Combining files with the copy command

If you specify more than one *source*, separating entries with a plus sign (+), **copy** combines the files, creating a single file. If you use wildcards in *source* but specify a single filename in *destination*, **copy** combines all files matching the filename in *source* and creates a single file with the filename specified in *destination*.

In either case, **copy** assumes the combined files are ASCII files unless you specify the **/b** switch.

Copying files in subdirectories

To copy all of a directory's files and subdirectories, you should use the **xcopy** command.

Copying zero-length files


Copy does not copy files that are 0 bytes long. Use xcopy to copy these files.


Changing the time and date of a file

If you want to assign the current time and date to a file without modifying the file, use a command in the following format. The commas indicate the omission of the *destination* parameter.

```
copy /b source+,,
```

More Information About Copy

 Copy--Examples

 Copy



Copy--Examples

The following command copies a file and ensures that an end-of-file character is at the end of the copied file:

```
copy memo.doc letter.doc /a
```

To copy a file named ROBIN.TYP from the current drive and directory to an existing directory named BIRDS that is located on drive C, type the following command:

```
copy robin.typ c:\birds
```

If the BIRDS directory doesn't exist, Windows NT copies the file ROBIN.TYP into a file named BIRDS that is located in the root directory on the disk in drive C.

To copy several files into one file, list any number of files as *source* parameters on the **copy** command line. Separate filenames with a plus sign (+) and specify a filename for the resulting combined file, as the following example shows:

```
copy mar89.rpt + apr89.rpt + may89.rpt report
```

This command combines the files named MAR89.RPT, APR89.RPT, and MAY89.RPT from the current drive and directory and places them in a file named REPORT in the current directory on the current drive. When files are combined, the destination file is created with the current date and time. If you omit *destination*, Windows NT combines the files and stores them under the name of the first specified file. For example, if a file named REPORT already exists, you can use the following command to combine all four files in REPORT:

```
copy report + mar89.rpt + apr89.rpt + may89.rpt
```

You can also combine several files into one by using wildcards, as the following example shows:

```
copy *.txt combin.doc
```

This command combines all files in the current directory on the current drive that have the extension .TXT into one file named COMBIN.DOC, also in the current directory on the current drive.

If you want to combine several binary files into one by using wildcards, include the **/b** switch, as the following example shows:

```
copy /b *.exe combin.exe
```

This prevents Windows NT from treating CTRL+Z as an end-of-file character.

Caution

If you combine binary files, the resulting file might not be usable due to internal formatting.


In the following example, **copy** combines each file that has a .TXT extension with its corresponding .REF file. The result is a file with the same filename but with a .DOC extension. Thus, **copy** combines FILE1.TXT with FILE1.REF to form FILE1.DOC. Then **copy** combines FILE2.TXT with FILE2.REF to form FILE2.DOC, and so on.

```
copy *.txt + *.ref *.doc
```

The following **copy** command combines first all files with the .TXT extension, then all files with the .REF extension into one file named COMBIN.DOC:

```
copy *.txt + *.ref combin.doc
```

More Information About Copy

 [Copy--Notes](#)

 [Copy](#)



Country

Enables the MS-DOS subsystem to use international time, dates, currency, case conversions, and decimal separators.

The country command configures the MS-DOS subsystem to recognize the character set and punctuation conventions observed when using one of the supported languages.

country=xxx[,yyy][,[drive:][path] filename]

Parameters

xxx

Specifies the country code.

yyy

Specifies the code page for the country.

[drive:][path] filename

Specifies the location and name of the file containing country information.

More Information About Country



[Country--Notes](#)



[Country--Examples](#)



Country--Notes

Specifying supported languages

The following table lists each country or language supported by the Windows NT MS-DOS subsystem. The table also lists the code pages you can use with each country code. For example, if you use country code 003, you can use only code page 437 or 850 for the *yyy* parameter. The first of the two code pages listed for each country or language is its default code page.

Country or language	Country code	Code pages
United States	001	437, 850
Canadian-French	002	863, 850
Netherlands	031	850, 437
Belgium	032	850, 437
France	033	850, 437
Spain	034	850, 437
Hungary	036	852, 850
Italy	039	850, 437
Switzerland	041	850, 437
Czechoslovakia	042	852, 850
United Kingdom	044	437, 850
Denmark	045	850, 865
Sweden	046	850, 437
Norway	047	850, 865
Poland	048	852, 850
Germany	049	850, 437
Brazil	055	850, 437
International English	061	437, 850
Portugal	351	850, 860
Finland	358	850, 437

Specifying international time and date formats


The country code specifies the time and date formats used by the following commands: **backup**, **date**, **restore**, and **time**.


The following table lists the date and time formats related to each country code. For each country code, the "Date format" column shows the display for January 3, 1994, and the "Time format" column shows the display for 5:35 P.M. (with 0 seconds and 0 hundredths of a second).

Country or language	Country code	Date format	Time format
United States	001	01/03/1994	5:35:00.00p
Canadian-French	002	1994-01-03	17:35:00
Latin America	003	03/01/1994	5:35:00.00p
Netherlands	031	03-01-94	17:35:00

Belgium	032	03/01/94	17:35:00
France	033	03.01.1994	17:35:00
Spain	034	3/01/94	17:35:00
Hungary	036	1994.01.03	17:35:00
Italy	039	03/01/94	17.35.00
Switzerland	041	03.01.94	17 35.00
Czechoslovakia	042	03.01.1994	17:35:00
United Kingdom	044	03/01/94	17:35:00.00
Denmark	045	03-01-94	17.35.00
Sweden	046	1994-01-03	17.35.00
Norway	047	03.01.94	17:35:00
Poland	048	1994-01-03	17:35:00
Germany	049	03.01.1994	17:35:00
Brazil	055	03/01/94	17:35:00
International English	061	03/01/1994	17:35:00.00
Portugal	351	03-01-1994	17:35:00
Finland	358	3.1.1994	17.35.00

More Information About Country

 [Country--Examples](#)

 [Country](#)



Country--Examples

To convert international currency, time, date, and case to French conventions, add the following command to your CONFIG.SYS file:

```
country=033
```

To specify a code page with the country code for France, type the following:

```
country=033,850
```

If you omit the code page but include the *[drive:][path] filename* parameter, you must still type the comma that would have preceded the code page, as the following example shows:

```
country=033,,c:\winnt\system32\country.sys
```

More Information About Country



Country--Notes



Country



Date

Displays the date or allows you to change the date from your terminal or from a batch program.

date [*mm-dd-yy*]

Parameter

mm-dd-yy

Sets the date you specify. Values for day, month, and year must be separated by periods (.), hyphens (-), or slash marks (/).

mm can be 1 through 12

dd can be 1 through 31


yy can be 80 through 99 or 1980 through 2099

See Also

 [Time](#)

More Information on Date

 [Date--Notes](#)

 [Date--Example](#)



Date--Notes

Adjusting for days in a month

Windows NT is programmed to change months and years correctly, whether the month has 28, 29, 30, or 31 days.

More Information on Date



[Date--Example](#)



[Date](#)



Date--Example

To set the date to January 12, 1994, type
date 01-12-1994

More Information on Date

 [Date--Notes](#)

 [Date](#)



Del (Erase)

Deletes specified files.

del [*drive:*][*path*] *filename* [*;* ...] [**/p**] [**/f**] [**/s**] [**/q**] [**/a:attributes**]

erase [*drive:*][*path*] *filename* [*;* ...] [**/p**] [**/f**] [**/s**] [**/q**] [**/a:attributes**]

Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of the file or set of files you want to delete. Multiple filenames can be used. Filenames can be separated by spaces, commas, or semicolons.

/p

Prompts you for confirmation before deleting the specified file.

/f

Forces deletion of read-only files.

/s

Deletes specified files from the current directory and all subdirectories.

/q

Quiet mode; does not prompt for delete confirmation.

/a


Deletes files based on specified attributes.

attributes

Can be any of the following file attributes:


- r** Read-only
- h** Hidden
- s** System
- a** Archive
- Prefix meaning “not”

See Also

 [Rmdir](#)

More Information About Del

 [Del--Notes](#)

 [Del--Examples](#)



Del--Notes

Using the /p switch

If you use the **/p** switch, **del** displays the name of a file and prompts you with a message in the following format:
filename, Delete (Y/N)?

Press Y to confirm the deletion, N to cancel the deletion and display the next filename (if you specified a group of files), or CTRL+C to stop the **del** command.

Deleting more than one file at a time

You can delete all the files in a directory by typing the **del** command followed by [*drive:*]path. For example,
del \work

will delete all files in the directory \WORK.

You can also use wildcards (* and ?) to delete more than one file at a time. However, you should use wildcards cautiously with the **del** command to avoid deleting files unintentionally. Suppose you type the following command:

```
del *.*
```

Del displays the following prompt:

All files in directory will be deleted! Are you sure (Y/N)?

Press Y and then ENTER to delete all files in the current directory, or press N and then ENTER to cancel the deletion.

Before you use wildcards with the **del** command to delete a group of files, you can use the same wildcards with the **dir** command to see a list of the names of all the files included in the group.

Caution

Once you delete a file from your disk, you may not be able to retrieve it.

If Command Extensions are enabled, the **/s** option displays as its output the names of the files that are deleted. Ordinarily, it shows the names of files it could not find.

More Information About Del



[Del--Examples](#)



[Del](#)



Del--Examples

To delete all the files in a directory named TEST on drive C, you can use either of the following commands:

```
del c:\test
```

```
del c:test*.*
```

More Information About Del



[Del--Notes](#)



[_Del](#)



Device

Loads into memory the device driver you specify.

Use the %systemroot%\SYSTEM32\CONFIG.NT file, or the equivalent startup file specified in a program's PIF, to load device drivers for the MS-DOS subsystem.

device=[drive:][path] filename [dd-parameters]

Parameters

[drive:][path] filename

Specifies the location and name of the device driver you want to load.

[dd-parameters]

Specifies any command-line information required by the device driver.

See Also



[_Devicehigh](#)

More Information About Device



[_Device--Examples](#)



Device--Example

To use an ANSI escape sequence to control the screen and keyboard for the MS-DOS subsystem, add the following command to your CONFIG.NT or equivalent startup file:

```
device=c:\winnt\system32\ansi.sys
```

More Information About Device



Device



Devicehigh

Loads device drivers into the upper memory area. Loading a device driver into the upper memory area frees more bytes of conventional memory for other programs.

devicehigh=[*drive:*][*path*] *filename* [*dd-parameters*]

To specify the minimum amount of memory that must be available before **devicehigh** attempts to load a device driver into the upper memory area, use the following syntax:

devicehigh size=*hexsize* [*drive:*][*path*] *filename* [*dd-parameters*]

Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of the device driver you want to load into the upper memory area.

dd-parameters

Specifies any command-line information required by the device driver.

hexsize

Specifies the minimum amount of memory (the number of bytes, in hexadecimal format) that must be available before **devicehigh** attempts to load a device driver into the upper memory area. You must use both **size** and *hexsize*, as shown in the second syntax line.

See Also



[Loadhigh](#)



[Device](#)

More Information About Devicehigh



[Devicehigh--Notes](#)



[Devicehigh--Examples](#)



Devicehigh--Notes

Using the dos=umb command

To use the **devicehigh** command, you must also include the **dos=umb** command in your CONFIG.NT or equivalent startup file. If you do not specify this command, all device drivers are loaded into conventional memory, as if you had used the **device** command. For more information about the **umb** switch, see the **dos** command.

Installing HIMEM.SYS

Before you can load a device driver into the upper memory area, you must use the **device** command to install the HIMEM.SYS device driver. This command must appear before any **devicehigh** command in your CONFIG.NT or equivalent startup file.

Specifying a size limit

If the device driver you specify on the **devicehigh** command line attempts to allocate more buffer space than is available in a block of the upper memory area, your system may lock up. You can try using the *hexsize* parameter to avoid this problem. In *hexsize*, indicate, in hexadecimal format, the amount of memory the device driver needs. To find this value for a particular device driver, load the driver into conventional memory and use the **mem /debug** command. This method is usually, but not always, effective.

If no upper memory area is available

If there is not enough upper memory area available to load the device driver you specified with the **devicehigh** command, the MS-DOS subsystem will load it into conventional memory (as if you had used the **device** command).

More Information About Devicehigh



[Devicehigh](#)



[Devicehigh--Examples](#)



Devicehigh--Example

If you include the following commands in your CONFIG.NT or equivalent startup file, the MS-DOS subsystem attempts to load a device driver named MYDRIV.SYS into the upper memory area

```
device=c:\winnt\system32\himem.sys
```

```
dos=umb
```

```
devicehigh=mydriv.sys
```

More Information About Devicehigh



[Devicehigh--Notes](#)



[Devicehigh](#)



Devinfo

Prepares a device to use code pages. To use this command, place it in your OS/2 C:\CONFIG.SYS file.

devinfo=*devtype,subtype,[drive:][path]filename [,ROM=[[({}xxx[,yyy)]][,...]]*

Parameters

devtype

Specifies the type of device: keyboard, monitor, or parallel printer. See below for possible values.

subtype

Specifies the style or model of the device. For a keyboard, this argument would specify the keyboard layout. See the list below for possible values.

filename

Specifies the file that contains information about the code pages for that device. See the list below for possible values.

ROM=

This parameter is not used by the OS/2 subsystem. It is accepted only for compatibility with files from MS OS/2 version 1.3 or earlier.

xxx

This parameter is not used by the OS/2 subsystem. It is accepted only for compatibility with files from MS OS/2 version 1.3 or earlier.

yyy

This parameter is not used by the OS/2 subsystem. It is accepted only for compatibility with files from MS OS/2 version 1.3 or earlier.

More Information About Devinfo



[Devinfo--Notes](#)



[Devinfo--Examples](#)



Devinfo--Notes

You must include a separate **devinfo** command in your OS/2 C:\CONFIG.SYS file for each device connected to your system, including the keyboard and the monitor, if you want to be able to switch code pages. The **devinfo** command specifies the kind of device you have connected to your system and the location of the code-page or keyboard information for that device.

The following list shows the values you can give to *devtype*, *subtype*, and *filename*:

Argument	Keyboard	Monitor
<i>devtype</i>	KBD	SCR
<i>subtype</i>	keyboard code	EGA, VGA
<i>filename</i>	KEYBOAR D.DCP	VIOTBL. DCP

Keyboard code is a code that identifies the keyboard layout for a particular country. For a list of the possible keyboard codes, see the [keyb](#) command. If you specify a keyboard for a country that has more than one enhanced keyboard, include the subcode in this setting, but do not include a space between the two-letter code and the subcode. For example, to specify the French enhanced keyboard 120, use FR120 for the keyboard code.

More Information About Devinfo



[Devinfo--Examples](#)



[Devinfo](#)



Devinfo--Examples

To prepare your keyboard to use the code pages you have specified with the codepage command, to use the United Kingdom keyboard layout, and to specify that the file containing code-page information is in the OS2 directory on your hard disk (drive C), include the following line in your C:\CONFIG.SYS file:

```
devinfo=kbd,uk,c:\os2\keyboard.dcp
```

More Information About Devinfo



[Devinfo--Notes](#)



[Devinfo](#)



Dir

Displays a list of a directory's files and subdirectories.

dir [*drive:*][*path*][*filename*] [; ...] [/p] [/w] [/d] [/a[:]*attributes*][/o[:]*sortorder*] [/t[:]*timefield*] [/s] [/b] [/l] [/n] [/x]

Parameters

none

Used without parameters or switches, **dir** displays the disk's volume label and serial number; one directory or filename per line, including the filename extension, the file size in bytes, and the date and time the file was last modified; and the total number of files listed, their cumulative size, and the free space (in bytes) remaining on the disk.

[*drive:*][*path*]

Specifies the drive and directory for which you want to see a listing.

[*filename*]

Specifies a particular file or group of files for which you want to see a listing. Multiple filenames can be used. Filenames can be separated by spaces, commas, or semicolons.

/p

Displays one screen of the listing at a time. To see the next screen, press any key.

/w

Displays the listing in wide format, with as many as five filenames or directory names on each line.

/d

Same as wide but files are sorted by column.

/a[:] *attributes*

Displays only the names of those directories and files with the attributes you specify. If you omit this switch, **dir** displays the names of all files except hidden and system files. If you use this switch without specifying *attributes*, **dir** displays the names of all files, including hidden and system files. The following list describes each of the values you can use for *attributes*. The colon (:) is optional. Use any combination of these values, and do not separate the values with spaces.

h Hidden files

s System files

d Directories

a Files ready for archiving (backup)

r Read-only files

-h Files that are not hidden

-s Files other than system files

-d Files only (not directories)

-a Files that have not changed since the last backup

-r Files that are not read-only

/o[:] *sortorder*

Controls the order in which **dir** sorts and displays directory names and filenames. If you omit this switch, **dir** displays the names in the order in which they occur in the directory. If you use this switch without specifying *sortorder*, **dir** displays the names of the directories, sorted in alphabetic order, and then displays the names of files, sorted in alphabetic order. The colon (:) is optional. The following list describes each of the values you can use for *sortorder*. Use any combination of the values, and do not separate these values with spaces.

n In alphabetic order by name

- e** In alphabetic order by extension
- d** By date and time, earliest first
- s** By size, smallest first
- g** With directories grouped before files
- n** In reverse alphabetic order by name (Z through A)
- e** In reverse alphabetic order by extension (Z through A)
- d** By date and time, latest first
- s** By size, largest first
- g** With directories grouped after files

/t[[:] *timefield*]

The following list describes each of the values you can use for timefield. Controls which time field is displayed or used for sorting.

- c** Creation
- a** Last access
- w** Last written

/s

Lists every occurrence, in the specified directory and all subdirectories, of the specified filename.

/b

Lists each directory name or filename, one per line (including the filename extension). This switch displays no heading information and no summary. The **/b** switch overrides the **/w** switch.

/l

Displays unsorted directory names and filenames in lowercase. This switch does not convert extended characters to lowercase.

/n

Displays long list format with filenames on far right.

/x

Displays the shortnames generated for files on NTFS and FAT volumes. Display is the same as with the **/n** switch, but shortnames are displayed after the longname.

See Also

 [Tree](#)

More Information About Dir

 [Dir--Notes](#)

 [Dir--Examples](#)



Dir--Notes

Using wildcards with dir

You can use wildcards (? and *) to display a listing of a subset of files and subdirectories. For an example illustrating the use of a wildcard, see the "Examples" screen.

Specifying file display attributes

If you specify the **/a** switch with more than one value in *attributes*, **dir** displays the names of only those files with all the specified attributes. For example, if you specify the **/a** switch with the **r** and **-h** values for attributes by using either **/a:r-h** or **/ar-h**, **dir** displays only the names of read-only files that are not hidden.

Specifying filename sorting

If you specify more than one *sortorder* value, **dir** sorts the filenames by the first criterion first, then by the second criterion, and so on. For example, if you specify the **/o** switch with the **e** and **-s** values for *sortorder* by using either **/o:e-s** or **/oe-s**, **dir** sorts the names of directories and files by extension, with the largest first, and displays the final result. The alphabetic sorting by extension causes filenames with no extensions to appear first, then directory names, then filenames with extensions.

Using redirection symbols and pipes

When you use a redirection symbol (>) to send **dir** output to a file, or a pipe (|) to send **dir** output to another command, use the **/a:-d** and **/b** switches to list only the filenames. You can use the *filename* parameter with the **/b** and **/s** switches to specify that **dir** is to search the current directory and its subdirectories for all filenames that match *filename*. **Dir** lists only the drive letter, directory name, filename, and filename extension, one path per line, for each filename it finds.

Before using a pipe for redirection, you should set the TEMP environment variable in your AUTOEXEC.NT file.

Presetting dir parameters and switches

You can preset **dir** parameters and switches by including the **set** command with the **dircmd** environment variable in your AUTOEXEC.NT file. You can use any valid combination of **dir** parameters and switches with the **set dircmd** command, including the location and name of a file.

For example, to use the **dircmd** environment variable to set the wide display format (**/w**) as the default format, include the following command in your AUTOEXEC.NT file:

```
set dircmd=/w
```

For a single use of the **dir** command, you can override a switch set by using the **dircmd** environment variable. To do so, you use the same switch on the **dir** command line, but you must also precede the switch letter with a minus sign, as the following example shows:

```
dir /-w
```

You can change the **dircmd** default settings by typing the **set** command at the command prompt with a new parameter or switch after the equal sign (=). The new default settings are effective for all subsequent **dir** commands until you use **set dircmd** again on the command line or until you restart Windows NT.

To clear all default settings, type the following command:

```
set dircmd=
```

You can view the current settings of the **dircmd** environment variable by typing the following command:

```
set
```

Windows NT displays a list of environment variables and their settings. For more information about setting environment variables, see the **set** command.

More Information About Dir



[Dir--Examples](#)



[Dir](#)



Dir--Examples

Suppose you want **dir** to display one directory listing after another, until it has displayed the listing for every directory on the disk in the current drive. Suppose also that you want **dir** to alphabetize each directory listing, display it in wide format, and pause after each screen.

To specify such a display, be sure the root directory is the current directory and then type the following command:

```
dir /s/w/o/p
```

Dir lists the name of the root directory, the names of the subdirectories of the root directory, and the names of the files in the root directory (including extensions). Then **dir** lists the subdirectory names and filenames in each subdirectory in the directory tree.

To alter the preceding example so that **dir** displays the filenames and extensions but omits the directory names, type the following command:

```
dir /s/w/o/p/a:-d
```

To print a directory listing, type the redirection symbol and PRN after any form of the **dir** command, as the following example shows:

```
dir > prn
```

When you specify PRN on the **dir** command line, the directory listing is sent to the printer attached to the LPT1 port. If your printer is attached to a different port, you must replace PRN with the name of the correct port.

You can also redirect output of the **dir** command to a file by replacing PRN with a filename. A path is also accepted on the command line. For example, to direct **dir** output to the file DIR.DOC in the RECORDS directory, type the following command:

```
dir > \records\dir.doc
```

If DIR.DOC does not exist, Windows NT creates it, unless the directory RECORDS also does not exist. In that case, Windows NT displays the following message:


```
File creation error
```

To display a list of all the filenames with the .TXT extension in all directories on drive C, type the following command:

```
dir c:\*.txt /w/o/s/p
```

Dir displays, in wide format, an alphabetized list of the matching filenames in each directory and pauses each time the screen fills, until you press a key to continue.

More Information About Dir

 [Dir--Notes](#)

 [Dir](#)



Diskcomp

Compares the contents of two floppy disks.

diskcomp [*drive1*: [*drive2*:]]

Parameters

drive1

Specifies the drive containing one of the floppy disks.

drive2


Specifies the drive containing the other floppy disk.


See Also

 [_Comp](#)

 [_Fc](#)

More Information About Diskcomp

 [Diskcomp--Notes](#)

 [Diskcomp--Examples](#)



Diskcomp--Notes

Invalid drive for diskcomp

The **diskcomp** command works only with floppy disks. You cannot use **diskcomp** with a hard disk. If you specify a hard disk drive for *drive1* or *drive2*, **diskcomp** displays the following error message:

```
Invalid drive specification  
Specified drive does not exist  
or is non-removable
```

Diskcomp messages

If all tracks on the two disks being compared are the same, **diskcomp** displays the following message:

```
Compare OK
```

If the tracks are not the same, **diskcomp** displays a message similar to the following:

```
Compare error on  
side 1, track 2
```

When **diskcomp** completes the comparison, it displays the following message:

```
Compare another diskette (Y/N)?
```

If you press Y, **diskcomp** prompts you to insert disks for the next comparison. If you press N, **diskcomp** stops the comparison.

When **diskcomp** makes the comparison, it ignores a disk's volume number.

Omitting drive parameters

If you omit the *drive2* parameter, **diskcomp** uses the current drive for *drive2*. If you omit both drive parameters, **diskcomp** uses the current drive for both. If the current drive is the same as *drive1*, **diskcomp** prompts you to swap disks as necessary.

Using one drive for the comparison

If you specify the same floppy disk drive for *drive1* and *drive2*, **diskcomp** does a comparison by using one drive and prompts you to insert the disks as necessary. You might have to swap the disks more than once, depending on the capacity of the disks and the amount of available memory.

Comparing different types of disks

Diskcomp cannot compare a single-sided disk with a double-sided disk, nor a high-density disk with a double-density disk. If the disk in *drive1* is not of the same type as the disk in *drive2*, **diskcomp** displays the following message:

```
Drive types or diskette types not compatible
```

Using diskcomp with networks and redirected drives

Diskcomp does not work on a network drive or on a drive created by the **subst** command. If you attempt to use **diskcomp** with a drive of any of these types, **diskcomp** displays an error message.

Comparing an original disk with a copy

When you use **diskcomp** with a disk that you made with the **copy** command, **diskcomp** may display a message similar to the following:

```
Compare error on  
side 0, track 0
```

This type of error can occur even if the files on the disks are identical. Although the **copy** command duplicates information, it doesn't necessarily place it in the same location on the destination disk. For more information about comparing individual files on two disks, see the **fc** command.


Diskcomp exit codes

The following list shows each exit code and a brief description of its meaning:

- 0 The disks are the same.
- 1 Differences were found.
- 3 A hard error occurred.
- 4 An initialization error occurred.

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by **diskcomp**.

More Information About Diskcomp

 [Diskcomp--Examples](#)

 [Diskcomp](#)



Diskcomp--Examples

If your system has only one floppy disk drive, drive A, and you want to compare two disks, type the following command:

```
diskcomp a: a:
```

Diskcomp **prompts you to insert each disk, as required.**

Following is an example of a batch program that uses the **errorlevel** parameter on the **if** command line to process a **diskcomp** exit code:

```
rem CHECKOUT.BAT compares the disks in drive A and B
echo off
diskcomp a: b:
if errorlevel 4 goto ini_error
if errorlevel 3 goto hard_error
if errorlevel 1 goto no_compare
if errorlevel 0 goto compare_ok
:ini_error
echo ERROR: Insufficient memory or command invalid
goto exit
:hard_error
echo ERROR: An irrecoverable error occurred
goto exit
:break
echo "You just pressed CTRL+C" to stop the comparison
goto exit
:no_compare
echo Disks are not the same
goto exit
:compare_ok
echo The comparison was successful; the disks are the same
goto exit
:exit
```

More Information About Diskcomp



[Diskcomp--Notes](#)



[Diskcomp](#)



Diskcopy

Copies the contents of the floppy disk in the source drive to a formatted or unformatted floppy disk in the destination drive.

diskcopy [*drive1*: [*drive2*:]] [/v]

Parameters

drive1

Specifies the drive containing the source disk.

drive2


Specifies the drive containing the destination disk.

/v

Verifies that the information is copied correctly. Use of this switch slows the copying process.

See Also


 [Copy](#)


 [Xcopy](#)

For information about copying one or more files, see the [copy](#) command.

For information about copying directories and subdirectories, see the [xcopy](#) command.

More Information About Diskcopy

 [Diskcopy--Notes](#)

 [Diskcopy--Example](#)



Diskcopy--Notes

Invalid drive for diskcopy

The **diskcopy** command works only with removable disks, such as floppy disks. You cannot use **diskcopy** with a hard disk. If you specify a hard disk drive for *drive1* or *drive2*, **diskcopy** displays the following error message:

```
Invalid drive specification  
Specified drive does not exist  
or is non-removable
```

Diskcopy messages

The **diskcopy** command prompts you to insert the source and destination disks and waits for you to press any key before continuing.

After copying, **diskcopy** displays the following message:

```
Copy another diskette (Y/N)?
```

If you press Y, **diskcopy** prompts you to insert source and destination disks for the next copy operation. To stop the **diskcopy** process, press N.

If you are copying to an unformatted floppy disk in *drive2*, **diskcopy** formats the disk with the same number of sides and sectors per track as are on the disk in *drive1*. **Diskcopy** displays the following message while it formats the disk and copies the files:

```
Formatting while copying
```

Disk serial numbers

If the source disk has a volume serial number, **diskcopy** creates a new volume serial number for the destination disk and displays the number when the copy operation is complete.

Omitting drive parameters

If you omit the *drive2* parameter, **diskcopy** uses the current drive as the destination drive. If you omit both drive parameters, **diskcopy** uses the current drive for both. If the current drive is the same as *drive1*, **diskcopy** prompts you to swap disks as necessary. Note that you currently cannot lock the current drive, but **diskcopy** requires that a drive be locked before copying files.

Using one drive for copying

If *drive1* and *drive2* are the same, **diskcopy** prompts you whenever you should switch disks. If you omit both *drive* parameters and the current disk drive is a floppy disk drive, **diskcopy** prompts you each time you should insert a disk in the drive. If the disks contain more information than available memory can hold, **diskcopy** cannot read all of the information at once. The **diskcopy** command reads from the source disk, writes to the destination disk, and prompts you to insert the source disk again. This process continues until the entire disk has been copied.

Avoiding disk fragmentation

Because **diskcopy** makes an exact copy of the source disk on the destination disk, any fragmentation on the source disk is transferred to the destination disk. Fragmentation is the presence of small areas of unused disk space between existing files on a disk.

A fragmented source disk can slow down the finding, reading, or writing of files. To avoid transferring fragmentation from one disk to another, use either the **copy** command or the **xcopy** command to copy your disk. Because **copy** and **xcopy** copy files sequentially, the new disk is not fragmented.

Caution

You cannot use **xcopy** to copy a startup disk.



Diskcopy exit codes

The following list briefly describes the meaning of each **diskcopy** exit code (**errorlevel**):

- 0 The copy operation was successful.
- 1 A nonfatal read/write error occurred.
- 3 A fatal hard error occurred.
- 4 An initialization error occurred.

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by **diskcopy**. For an example of a batch program that processes exit codes, see the **diskcomp** command.

More Information About Diskcopy


-  [Diskcopy--Example](#)
-  [Diskcopy](#)



Diskcopy--Example

To create an exact copy of the disk in drive B on the disk in drive A, type
diskcopy b: a:

More Information About Diskcopy

 [Diskcopy--Notes](#)

 [Diskcopy](#)



Diskperf

Starts and stops system disk performance counters.

diskperf [-y|-n] [\\computername]

none

Used without the **-y** or **-n** switches, **diskperf** reports whether disk performance counters are enabled on the local or specified computer.

-y

Sets the system to start disk performance counters when the system is rebooted.

-n

Sets the system to not use disk performance counters when the system is rebooted.

computername

Is the name of the computer on which you want to see or set disk performance counter use.

More Information About Diskperf



[Diskperf--Notes](#)




[Diskperf--Example](#)



Diskperf--Notes

Diskperf **must start at boot to monitor disk activity with Performance Monitor.**

More Information About Diskperf

 [Diskperf--Notes](#)

 [Diskperf--Example](#)



Diskperf--Examples

To set the disk performance counters on the computer PRODUCTION to start at boot, type

```
diskperf -y \\production
```

To stop the disk performance counters from starting at boot on the local computer, type

```
diskperf -n
```

More Information About Diskperf



[Diskperf--Notes](#)



[Diskperf](#)



Dos

Specifies that the MS-DOS subsystem is to maintain a link to the upper memory area or is to load part of itself into the high memory area (HMA).

dos=high|low[,umb|,noumb]

dos=[high,|low,]umb|noumb

Parameters

umb|noumb


Specifies whether the MS-DOS subsystem should maintain a link between conventional memory and the upper memory area. The **umb** parameter provides this link. The **noumb** parameter disconnects this link. The default setting is **noumb**.

high|low


Specifies whether the MS-DOS subsystem should attempt to load a part of itself into the HMA. Use the **high** parameter to enable the MS-DOS subsystem to load itself into the HMA. Use the **low** parameter to keep all of the MS-DOS subsystem in conventional memory. The default setting is **low**.

See Also

 [Devicehigh](#)

 [Loadhigh](#)

More Information About Dos

 [_Dos--Notes](#)



Dos--Notes

Must install HIMEM.SYS for dos=umb or dos=high.

You must install the HIMEM.SYS device driver before you specify either **dos=umb** or **dos=high**.

Using the UMB parameter

You must specify the **dos=umb** command in order to load programs and device drivers into the upper memory area. Using the upper memory area frees more space in conventional memory for programs. In addition to using this command, you must install an upper-memory-block (UMB) provider.

Using the high parameter

If you specify the **high** parameter, the MS-DOS subsystem attempts to load part of itself into the HMA. Loading part of MS-DOS into the HMA frees conventional memory for programs.

Combining parameters

You can include more than one parameter on a single **dos** command line, using commas to separate them. For example, the following command lines are valid:

dos=umb,low

dos=high,umb

You can place the **dos** command anywhere in your CONFIG.NT or equivalent startup file.

More Information About Dos



[_Dos](#)



Doskey

Calls the Doskey program, which recalls Windows NT commands, edits command lines, and creates macros.

doskey [/reinstall] [/listsize=*size*] [/macros:[**all** | *exename*] [/history] [/insert|/overstrike] [/exename=*exename*] [/macrofile=*filename*] [*macroname*]=[*text*]]

Parameters

/reinstall

Clears the command history buffer.

/listsize=*size*

Specifies the maximum number of commands in the history buffer.

/macros

Displays a list of all Doskey macros. You can use a redirection symbol (>) with the **/macros** switch to redirect the list to a file. You can abbreviate the **/macros** switch as **/m**.

all

Displays Doskey macros for all executables.

exename

Displays Doskey macros for the specified executable.

/history

Displays all commands stored in memory. You can use a redirection symbol (>) with the **/history** switch to redirect the list to a file. You can abbreviate the **/history** switch as **/h**.

/insert | **/overstrike**

Specifies whether new text you type is to replace old text. If you use the **/insert** switch, new text that you type on a line is inserted into old text (as if you had pressed the INSERT key). If you use the **/overstrike** switch, new text replaces old text. The default setting is **/overstrike**.

/exename=*exename*

Specifies the program (executable) in which the Doskey macro will run.

/macrofile=*filename*

Specifies a file containing macros to install.

macroname=[*text*]

Creates a macro that carries out the commands specified by *text*. *Macroname* specifies the name you want to assign to the macro. *Text* specifies the commands you want to record. If *text* is left blank, *macroname* is cleared of any assigned commands.

More Information About Doskey



[Doskey--Notes](#)



[Doskey--Examples](#)



Doskey--Notes

Using Doskey

The Doskey program is always available for all character-based, interactive programs (such as programming debuggers or file transfer programs), and maintains a command history buffer and macros for each program started. You cannot use doskey switches from a program; **doskey** switches must be issued before starting a program. Program key assignments override **doskey** key assignments.

Recalling a command

To recall a command, you can use any of the following keys after loading Doskey into memory. If using the Doskey program within a program, that program's key assignments take precedence.

UP ARROW	Recalls the command you used before the one displayed.
DOWN ARROW	Recalls the command you used after the one displayed.
PAGE UP	Recalls the oldest command you used in the current session.
PAGE DOWN	Recalls the most recent command you used.

Editing the command line

With the Doskey program, you can edit the current command line. If using the Doskey program within a program, that program's key assignments take precedence and not all Doskey editing keys may work.

The following list describes the Doskey editing keys and their functions:

LEFT ARROW	Moves the cursor back one character.
RIGHT ARROW	Moves the cursor forward one character.
CTRL+LEFT ARROW	Moves the cursor back one word.
CTRL+RIGHT ARROW	Moves the cursor forward one word.
HOME	Moves the cursor to the beginning of the line.
END	Moves the cursor to the end of the line.
ESC	Clears the

	command from the display.
F1	Copies one character from the same column in the template to the same column in the Windows NT command prompt. (The template is a memory buffer that holds the last command you typed.)
F2	Searches forward in the template for the next key you type after pressing F2. Doskey inserts the text from the template up to but not including the character you specify.
F3	Copies the remainder of the template to the command line. Doskey begins copying characters from the position in the template that corresponds to the position indicated by the cursor on the command line.
F4	Deletes characters from the current cursor position up to a character you specify. To use this editing key, you press F4 and type a character. Doskey deletes

	the characters from the current cursor position to the first letter specified.
F5	Copies the template into the current command line.
F6	Places an end-of-file character (CTRL+Z) at the current cursor position.
F7	Displays all commands for this program stored in memory in a pop-up box. Use the up and down arrow keys to select the command you want and press ENTER. The command will run. You can also note the sequential number in front of the command and use this number in conjunction with the F9 key.
ALT+F7	Deletes all commands stored in memory for the current history buffer.
F8	Displays all commands in the history buffer that start with the characters in the current command.
F9	Prompts you for a history buffer

	command number, then displays the command associated with the number you specify. Press ENTER to run the command. To display all the numbers and their associated commands, press F7.
ALT+F10	Deletes all macro definitions.

Using Doskey within a program

Certain character-based, interactive programs, such as programming debuggers or file transfer programs (FTP), automatically use the Doskey program. To use Doskey a program must be a console process and use buffered input. Any program key assignments override Doskey key assignments. For example, if the program uses the F7 key for some function, you will be unable to get a Doskey command history in a popup window.

Doskey maintains a command history for each program you start, allows you to repeat and edit previous commands at the program's prompt, and allows you to start Doskey macros created for the program. If you exit then restart a program from the same command prompt window, the command history from the previous program session will be available.

Doskey switches must be issued before starting a program. You cannot use Doskey switches from a program's command prompt, even if the program has a shell command.

If you use a program frequently and want to customize how Doskey works with the program and to create Doskey macros for that program, you can create a batch program that modifies Doskey and starts the program.

Specifying a default insert mode

If you press the INSERT key, you can type text on the Doskey command line in the middle of old text without replacing the old text. However, once you press ENTER, Doskey returns your keyboard to replace mode. You must press INSERT again to return to insert mode.

The **/insert** switch puts your keyboard in insert mode each time you press ENTER. Your keyboard effectively remains in insert mode until you use the **/overstrike** switch. You can temporarily return to replace mode by pressing the INSERT key; but once you press ENTER, Doskey returns your keyboard to insert mode.

The cursor changes shape when you use the INSERT key to change from one mode to the other.

Creating a macro

You can use the Doskey program to create macros that carry out one or more commands.

You can use the following special characters to control command operations when defining a macro:

\$G or \$g	Redirects output. Use either of these special characters to send output to a device or a file instead of to the screen. This character is equivalent to the
------------	---

	redirection symbol for output (>).
\$G\$G or \$g\$g	Appends output to the end of a file. Use either of these special double characters to append output to an existing file rather than replace the data in the file. These double characters are equivalent to the "append" redirection symbol for output (>>).
\$L or \$l	Redirects input. Use either of these special characters to read input from a device or a file instead of from the keyboard. This character is equivalent to the redirection symbol for input (<).
\$B or \$b	Sends macro output to a command. Using one of these special characters is equivalent to using the pipe () on a command line.
\$T or \$t	Separates commands. Use either of these special characters to separate commands when you are creating macros or typing commands on the Doskey command line. Using one of these special characters is equivalent to using the ampersand (&) on a command line.
\$\$	Specifies the dollar-sign character (\$).
\$1 through \$9	Represent any command-line information you want to specify when you run the macro. The special characters \$1

through \$9 are batch parameters, which make it possible for you to use different data on the command line each time you run the macro. The \$1 character in a **doskey** command is similar to the %1 character in a batch program.

\$*

Represents all the command-line information you want to specify when you type the macro name. The special character \$* is a replaceable parameter that is similar to the batch parameters \$1 through \$9, with one important difference. Here, everything you type on the command line after the macro name is substituted for the \$* in the macro.

For example, to create a macro that performs a quick and unconditional format of a disk, type the following command:

```
doskey qf=format $1 /q  
/u
```

For information about quick and unconditional formatting, see the [format](#) command.

Running a Doskey macro

To run a macro, type the macro name starting at the first position on the command line. If the macro was defined with \$* or any of the batch parameters \$1 through \$9, use a space to separate parameters.

You could run the QF macro created in the previous example to format a disk in drive A quickly and unconditionally. To do so, you would type the following command:

qf a:

You cannot run a Doskey macro from a batch program.

Creating a macro with the same name as a Windows NT command

You might want to create a macro that has the same name as a Windows NT command. This can be useful, for example, if you always use a certain command with specific switches. To specify whether you want to run the macro or the Windows NT command, follow these guidelines:

- To run the macro, begin typing the macro name immediately after the command prompt, with no space between the prompt and the command name.
- To carry out the command, insert one or more spaces between the command prompt and the command name.

Deleting a macro

To delete a macro, type the following command:

```
doskey macroname=
```

More Information About Doskey

■ [_Doskey--Examples](#)

■ [_Doskey](#)



Doskey--Examples

The **/macros** and **/history** switches are useful for creating batch programs to save macros and commands. For example, to store all current Doskey macros, type the following command:

```
doskey /macros > macinit
```

To then use the macros stored in MACINIT, type

```
doskey /macrofile=macinit
```

To create a batch program named TMP.BAT that contains recently used commands, type the following command:

```
doskey /history > tmp.bat
```

To define a macro with multiple commands, use **\$t** to separate commands, as follows:

```
doskey tx=cd temp$tdir/w $*
```

In the preceding example, the TX macro changes the current directory to TEMP and then displays a directory listing, using the wide display format. You can use **\$*** at the end of the macro to append other switches to the **dir** command when you run TX.

The following macro uses a batch parameter for a new directory name. The macro first creates a new directory and then changes to it from the current directory.

```
doskey mc=md $1$tc d $1
```

To use the preceding macro to create and change to a directory named BOOKS, you type the following:

```
mc books
```

To create a Doskey macro for a specific program (in this case FTP.EXE) include the **/exename** switch.

```
doskey /exename:ftp.exe go=open 126.127.1.100$tmget *.TXT c:\reports$bye
```

To use the preceding macro, start **ftp**. At the **ftp** prompt type **go**. **Ftp** will execute the **open**, **mget**, and **bye** commands.

More Information About Doskey



[Doskey--Notes](#)



[Doskey](#)





Dosonly

Prevents starting applications other than MS-DOS-based applications from the COMMAND.COM prompt.

dosonly

More Information About Dosonly

 [Dosonly--Notes](#)

 [Dosonly--Examples](#)



Dosonly--Notes

Normally, when you exit an MS-DOS-based application, Windows NT returns to the Windows NT command interpreter, CMD.EXE. However, by default, when you run a TSR or temporarily suspend an MS-DOS-based application to return to the command prompt, Windows NT runs COMMAND.COM, the command interpreter for the MS-DOS environment. This preserves the MS-DOS environment, allowing you to use the TSR immediately.

Because starting and running other types of applications from the COMMAND.COM prompt can disrupt a TSR or suspended MS-DOS-based application, Windows NT provides the **dosonly** command. The **dosonly** command allows only MS-DOS-based applications to be started from the COMMAND.COM prompt. You can include the **dosonly** command in your CONFIG.NT file or the equivalent custom startup file in an application's PIF.

More Information About Dosonly



[Dosonly--Examples](#)



[_Dosonly](#)




Dosonly--Examples

Include dosonly in your CONFIG.NT file, or the Config startup file specified in the PIF.

dosonly

More Information About Dosonly

 [Dosonly--Notes](#)

 [Dosonly](#)

**Driveparm**

Windows NT and the MS-DOS subsystem do not take action for this command. It is accepted only for compatibility with MS-DOS files.



Echo

Turns the command-echoing feature on or off, or displays a message.

echo [**on** | **off**] [*message*]

Parameters

on | **off**

Specifies whether to turn the command-echoing feature on or off. To display the current **echo** setting, use the **echo** command without a parameter.

message

Specifies text you want Windows NT to display on the screen.

See Also



[Pause](#)

More Information About Echo



[Echo--Notes](#)



[Echo--Examples](#)



Echo--Notes

Using a message with the echo command

The **echo** *message* command is useful when **echo** is off. To display a message that is several lines long without displaying other commands, you can include several **echo** *message* commands after the **echo off** command in your batch program.

Hiding the command prompt

If you use the **echo off** command on the command line, the command prompt does not appear on your screen. To redisplay the command prompt, type **echo on**.

Preventing Windows NT from echoing a line

You can insert an at sign (@) in front of a command in a batch program to prevent Windows NT from echoing that line.

Echoing a blank line

To echo a blank line on the screen, you can type **echo** and then a period (**echo.**). There must be no intervening space.

Displaying pipes and redirection characters

You cannot display a pipe (|) or redirection character (< or >) when using the **echo** command.

More Information About Echo



[Echo--Examples](#)



[Echo](#)



Echo--Examples

The following example shows a batch program that includes a three-line message preceded and followed by a blank line:

```
echo off
echo.
echo This batch program
echo formats and checks
echo new disks
echo.
```

If you want to turn **echo** off and you do not want to echo the **echo** command itself, include an at sign (@) before the command, as follows:

```
@echo off
```

You can use the **if** and **echo** commands on the same command line, as follows:

```
if exist *.rpt echo The report has arrived.
```

More Information About Echo



[Echo--Notes](#)



[Echo](#)

**Echoconfig**

Displays messages during the processing of the MS-DOS subsystem CONFIG.NT and AUTOEXEC.NT when the MS-DOS subsystem is invoked. If this command is not present, messages will not be displayed.

This command must be in the MS-DOS subsystem CONFIG.NT file.

echoconfig



Edit

Starts MS-DOS Editor, which creates and changes ASCII text files.

edit [[*drive:*][*path*] *filename*] [**/b**] [**/g**] [**/h**] [**/nohi**]

Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of an ASCII text file. If the file does not exist, MS-DOS Editor creates it. If the file exists, MS-DOS Editor opens it and displays its contents on the screen.

/b

Displays MS-DOS Editor in black and white. You use this option if MS-DOS Editor isn't displayed correctly on a monochrome monitor.

/g

Uses the fastest screen updating possible for a CGA monitor.

/h

Displays the maximum number of lines possible for the monitor you are using.

/nohi

Enables you to use 8-color monitors with MS-DOS Editor. Usually, Windows NT uses 16 colors.

Caution



MS-DOS Editor does not work if the file QBASIC.EXE is not in the current directory, in the search path, or in the same directory as the file EDIT.COM. If you delete QBASIC.EXE to save space on your hard disk, you cannot use MS-DOS Editor.

Note



Some monitors may not support the display of shortcut keys by default. If your monitor does not display shortcut keys, use the **/b** switch (for CGA monitors) and the **/nohi** switch (for systems that do not support bold characters).



Endlocal

Ends localization of environment changes in a batch file. Each **setlocal** command must have an **endlocal** command to restore environment variables.

endlocal

See Also



[Setlocal](#)

More Information About Endlocal



[Endlocal--Example](#)



Endlocal--Example

You can localize environment variables in a batch file.

```
@echo off
```

```
rem This program starts the superapp batch program on the network,
```

```
rem directs the output to a file, and displays the file
```

```
rem in Notepad.
```

```
setlocal
```

```
path=g:\programs\superapp;%path%
```

```
call superapp>c:\superapp.out
```

```
endlocal
```

```
start notepad c:\superapp.out
```

More Information About Endlocal



[Endlocal](#)



Exe2bin

Converts .EXE (executable) files to binary format.

Exe2bin is included with Windows NT as a courtesy to software developers. It is not useful for general users.

exe2bin [*drive1:*][*path1*]*input-file* [[*drive2:*][*path2*]*output-file*]

Parameters

[*drive1:*][*path1*]*input-file*

Specifies the location and name of the input file.

[*drive2:*][*path2*]*output-file*

Specifies the location and name of the output file.

More Information About Exe2bin



[Exe2bin--Notes](#)



Exe2bin--Notes

Restrictions on using exe2bin

The following restrictions apply when you use the **exe2bin** command:

1bmct bullet.bmp} The input file must be in valid .EXE format produced by the linker and must not be packed.



The resident, or actual, code and data portions of the file combined must be less than 64K.



There must be no **stack** segment.

Default values for parameters

The **exe2bin** command takes specific actions, depending upon the values you use for the *input-file* and *output-file* parameters.



The default filename extension for the filename you specify for *input-file* is .EXE. The **exe2bin** command converts the input .EXE file to an output file in .BIN format (a memory image of the program) and uses the location and filename you specify for [*drive2:*][*path2*]*output-file* to store that output file.



If you do not specify *drive2* or *path2*, **exe2bin** writes the output file to the current drive and directory.



If you do not specify an output filename, **exe2bin** uses the input filename.



The default extension for the filename specified for the output-file parameter is .BIN.

Types of conversion available with exe2bin

Two types of conversion are possible, depending upon whether the initial CS:IP (Code Segment:Instruction Pointer) is specified in the .EXE file. The following list presents the two types:



If the CS:IP is not specified in the .EXE file, **exe2bin** performs a pure binary conversion. If segment fixups are necessary (that is, if the program contains instructions requiring segment relocation), **exe2bin** prompts you for the fixup value. This value is the absolute segment at which the program is to be loaded. The resulting program is usable only when loaded at the absolute memory address specified by your program. The command interpreter cannot load the program.



If the CS:IP is specified as 0000:100H, the file runs as a .COM file with the instruction pointer set at 100H by the assembler statement ORG. Include the .COM extension in the *output-file* parameter. No segment fixups are allowed, because .COM files must be segment-relocatable; that is, they must assume the entry conditions explained in the Microsoft Macro Assembler manuals. The command interpreter can then load and run the program in the same way as it loads and runs the .COM programs supplied on your Windows NT disk.

More Information About Exe2bin



[Exe2bin](#)

**Exit**

Quits the CMD.EXE program (the command interpreter) and returns to the program that started CMD.EXE, if one exists, or to the Program Manager.

exit



Expand

Expands one or more compressed files. This command is used to retrieve compressed files from distribution disks.

expand [-r] *source* [*destination*]

Parameters

-r

Renames expanded files.

source

Specifies the files to expand. *Source* can consist of a drive letter and colon, a directory name, a filename, or a combination. Wildcards may be used.

destination

Specifies where files are to be expanded. If *source* is multiple files and **-r** is not specified, *destination* must be a directory. *Destination* can consist of a drive letter and colon, a directory name, a filename, or a combination.

**Fastopen**

Windows NT and the MS-DOS subsystem do not use this command. It is accepted only for compatibility with MS-DOS files.



Fc

Compares two files and displays the differences between them.

fc [/a] [/b] [/c] [/l] [/lbn] [/n] [/t] [/u] [/w] [/nnnn] [drive1:][path1]filename1 [drive2:][path2]filename2

Parameters

/a

Abbreviates the output of an ASCII comparison. Instead of displaying all the lines that are different, **fc** displays only the first and last line for each set of differences.

/b

Compares the files in binary mode. Fc compares the two files byte by byte and does not attempt to resynchronize the files after finding a mismatch. This is the default mode for comparing files that have extensions of .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN.

/c

Ignores the case of letters.

/l

Compares the files in ASCII mode. Fc compares the two files line by line and attempts to resynchronize the files after finding a mismatch. This is the default mode for comparing files that do not have extensions of .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN.

/lbn

Sets the number of lines for the internal line buffer. The default length of the line buffer is 100 lines. If the files being compared have more than this number of consecutive differing lines, **fc** cancels the comparison.

/n

Displays the line numbers during an ASCII comparison.

/t

Does not expand tabs to spaces. The default behavior is to treat tabs as spaces, with stops at each eighth character position.

/u

Compares files as Unicode text files.

/w

Compresses white space (tabs and spaces) during the comparison. If a line contains many consecutive spaces or tabs, the **/w** switch treats these characters as a single space. When used with the **/w** switch, **fc** ignores (and does not compare) white space at the beginning and end of a line.

/nnnn

Specifies the number of consecutive lines that must match before fc considers the files to be resynchronized. If the number of matching lines in the files is less than this number, **fc** displays the matching lines as differences. The default value is 2.

[drive1:][path1]filename1

Specifies the location and name of the first file you want to compare.

[drive2:][path2]filename2

Specifies the location and name of the second file you want to compare.

More Information About Fc



[Fc--Notes](#)



[Fc--Examples](#)



Fc--Notes

Reporting differences between files for an ASCII comparison

When you use `fc` for an ASCII comparison, Windows NT reports differences between two files by displaying the name of the first file, followed by the lines from *filename1* that differ between the files, followed by the first line to match in both files. Windows NT then displays the name of the second file, followed by the lines from *filename2* that differ, followed by the first line to match.

Using the `/b` switch for binary comparisons

Windows NT uses the following format to report mismatches found during a binary comparison:

```
xxxxxxx: yy zz
```

The value of `xxxxxxx` specifies the relative hexadecimal address for the pair of bytes, measured from the beginning of the file. Addresses start at 00000000; the hexadecimal values for `yy` and `zz` represent the mismatched bytes from *filename1* and *filename2*, respectively.

Using wildcards

You can use wildcards (`?` and `*`) in either of the filenames you specify with the `fc` command. If you use a wildcard in *filename1*, `fc` compares all the specified files to the file specified by *filename*. If you use a wildcard in *filename2*, `fc` uses the corresponding value from *filename1*.

How `fc` uses memory

When comparing ASCII files, `fc` uses an internal buffer (large enough to hold 100 lines) as storage. If the files are larger than the buffer, `fc` compares what it can load into the buffer. If `fc` does not find a match in the loaded portions of the files, it stops and displays the following message:

```
Resynch failed. Files are too different.
```

When comparing binary files that are larger than available memory, `fc` compares both files completely, overlaying the portions in memory with the next portions from the disk. The output is the same as that for files that fit completely in memory.

More Information About Fc



[Fc--Examples](#)



[Fc](#)



Fc--Examples

Suppose you want to make an ASCII comparison of two text files that are named MONTHLY.RPT and SALES.RPT, and you want to display the results in abbreviated format. To make this comparison, type the following command:

```
fc /a monthly.rpt sales.rpt
```

To make a binary comparison of two batch files named PROFITS.BAT and EARNINGS.BAT, type the following command:

```
fc /b profits.bat earnings.bat
```

The results of this command will be similar to the following:

```
00000002: 72 43
00000004: 65 3A
0000000E: 56 92
00000012: 6D 5C
00000013: 0D 7C
00000014: 0D 0A
00000015: 0A 0D
0000001E: 43 7A
0000001F: 09 0A
00000022: 72 44
```

...

...

...

```
000005E0: 00 61
000005E1: 00 73
000005E2: 00 73
000005E3: 00 69
000005E4: 00 67
000005E5: 00 6E
000005E6: 00 6D
000005E7: 00 65
000005E8: 00 6E
```

FC: EARNINGS.BAT longer than PROFITS.BAT

If the PROFITS.BAT and EARNINGS.BAT files were identical, **fc** would display the following message:

FC: no differences encountered

To compare every .BAT file in the current directory with the file NEW.BAT, type the following command:

```
fc *.bat new.bat
```

To compare the file NEW.BAT on drive C with the file NEW.BAT on drive D, type the following command.

```
fc c:new.bat d:*.bat
```

To compare each batch file in the root directory on drive C to the file with the same name in the root directory on drive D, type the following command:

```
fc c:*.bat d:*.bat
```

More Information About Fc



[Fc--Notes](#)



[Fc](#)

**Fcbs**

Specifies the number of file control blocks (FCBs) that the MS-DOS subsystem can have open at the same time.

A file control block is a data structure that stores information about a file.


fcbs=x


Parameter

x

Specifies the number of file control blocks that the MS-DOS subsystem can have open at one time. Valid values for x are in the range 1 through 255. The default value is 4.

More Information About Fcbs

 [Fcbs--Notes](#)

 [Fcbs--Examples](#)



Fcbs--Notes

Limitation on opening files

If a program tries to open more than x files by using file control blocks, the MS-DOS subsystem might close the files that were opened earlier.

Recommended use of the fcbs command

You should use the **fcbs** command only if a program requires you to do so. Most newer programs do not require file control blocks. However, some older programs might require you to use the **fcbs** command in your CONFIG.NT file.

More Information About Fcbs



[Fcbs--Examples](#)



[Fcbs](#)




Fcbs--Example

To specify that the MS-DOS subsystem can have up to eight file control blocks open at the same time, add the following line to your CONFIG.NT file:

fcbs=8

More Information About Fcbs

 [Fcbs--Notes](#)

 [Fcbs](#)



Files

Sets the number of files that the MS-DOS subsystem can access at one time.

files=x

Parameter

x

Specifies the number of files that the MS-DOS subsystem can access at one time. Valid values for *x* are in the range 8 through 255. The default value is 8.

More Information About Files



[Files--Notes](#)




[Files--Examples](#)



Files--Note

Although the default setting for the x parameter is 8, some programs require a larger value. A typical setting is 20.

More Information About Files

 [Files--Examples](#)

 [Files](#)



Files--Example

To specify that the MS-DOS subsystem can access up to 20 files at one time, add the following line to your CONFIG.NT file:

files=20

More Information About Files

 [Files--Notes](#)

 [Files](#)



Find

Searches for a specific string of text in a file or files.

After searching the specified files, **find** displays any lines of text that contain the specified string.

find [/v] [/c] [/n] [/i] "string" [[drive:][path]filename[...]]

Parameters

"string"

Specifies the group of characters you want to search for. You must enclose the text for *string* in quotation marks.

[drive:][path] filename

Specifies the location and name of the file in which to search for the specified string.

/v

Displays all lines not containing the specified string.

/c

Displays only a count of the lines that contain the specified string.

/n

Precedes each line with the file's line number.

/i

Specifies that the search is not to be case-sensitive.

See Also



[Filter commands](#)



[Redirection symbols](#)

More Information About Find



[Find--Notes](#)



[Find--Examples](#)



Find--Notes

Specifying a string

Unless you specify the **/i** switch, **find** searches for exactly what you specify for string. For example, to the **find** command, the characters "a" and "A" are different. If you were to use the **/i** switch, however, **find** would ignore case and search for "a" and "A" as if they were the same character.

If the string you want to search for contains quotation marks, you must use two quotation marks for each quotation mark contained within the string.

Using find as a filter

If you omit a filename, **find** acts as a filter, taking input from the Windows NT standard source (usually the keyboard, a pipe, or a redirected file) and displaying any lines that contain string.

Ordering the command

Parameters and switches for the **find** command can be typed in any order.

Using wildcards with find

You cannot use wildcards (* and ?) in filenames or extensions that you specify with the **find** command. To search for a string in a set of files you specify with wildcards, you can use the **find** command in a **for** command.

Using the /v or /n switch with the /c switch

If you specify the **/c** and **/v** switches in the same command, **find** displays a count of the lines that do not contain the specified string. If you specify the **/c** and **/n** switches in the same command, **find** ignores the **/n** switch.

Using find in files with carriage returns

The **find** command does not recognize carriage returns. When you use **find** to search for text in a file that includes carriage returns, you must limit the search string to text that can be found between carriage returns--that is, a string that is not likely to be interrupted by a carriage return. For example, **find** does not report a match for the string "tax file" wherever a carriage return occurs between the word "tax" and the word "file".

More Information About Find



[Find--Examples](#)



[Find](#)



Find--Examples

To display all lines from the file PENCIL.AD that contain the string "Pencil Sharpener", type the following command:

```
find "Pencil Sharpener" pencil.ad
```

To find a string that contains text within quotation marks, you must enclose the entire string in quotation marks and, in addition, use two quotation marks for each quotation mark contained within the string, as shown in the following example:

```
find "The scientists labeled their paper ""for discussion only."" It is not a final report." report.doc
```

If you want to search for a set of files, you can use the **find** command with the **for** command. The following command uses this method to search the current directory for files that have the extension .BAT; in each file found, the command searches for the string "PROMPT":

```
for %f in (*.bat) do find "PROMPT" %f
```

Suppose you want **find** to search your hard disk to find and display the filenames on drive C that contain the string "CPU". To do this you can use the pipe (|) to direct the results of a **dir** command to **find**, as shown in the following example:

```
dir c:\ /s /b | find "CPU"
```

Since **find** searches are case-sensitive and since **dir** produces uppercase output, you must either type the string "CPU" in uppercase letters or use the **/i** switch with **find**.

More Information About Find



[Find--Notes](#)



[Find](#)



Findstr

Searches for strings in files using literal text or regular expressions. See **findstr** notes in this file for a list of the regular expression symbols accepted by **findstr**.

findstr [/b] [/e] [/l] [/c:string] [/r] [/s] [/i] [/x] [/v] [/n] [/m] [/o] [/g:file] [/f:file] strings files

Parameters

/b

Matches pattern if at the beginning of a line.

/e

Matches pattern if at the end of a line.

/l

Uses search strings literally.

/c

Uses specified text as a literal search string.

/r

Uses search strings as regular expressions. This switch is not required; **findstr** will interpret all metacharacters as regular expressions unless the **/l** switch is used.

/s

Searches for matching files in the current directory and all subdirectories.

/i

Specifies that the search is not to be case-sensitive.

/x

Prints lines that match exactly.

/v

Prints only lines that do not contain a match.

/n

Prints the line number before each line that matches.

/m

Prints only the filename if a file contains a match.

/o

Prints seek offset before each matching line.

/g

Gets search strings from the specified file.

/f

Reads file list from the specified file.

strings

Text to be searched for.

files

Files to be searched.

Use spaces to separate multiple search strings unless the argument is prefixed with **/e**. For example,

`findstr "hello there" x.y`


searches for "hello" or "there" in file x.y. On the other hand,

findstr /c:"hello there" x.y'

searches for "hello there" in file x.y.

More Information About Findstr

 [Findstr--Notes](#)

 [Findstr--Examples](#)



Findstr--Notes

Findstr is capable of finding the exact text you are looking for in any ASCII file or files. Sometimes, however, you have only part of the information that you want to match, or you want to find a wider range of information. In such cases, **findstr** has the powerful capability to search for patterns of text using regular expressions.

Regular expressions are a notation for specifying patterns of text, as opposed to exact strings of characters. The notation uses literal characters and metacharacters. Every character that does not have special meaning in the regular-expression syntax is a literal character and matches an occurrence of that character. For example, letters and numbers are literal characters. A metacharacter is a symbol with special meaning (an operator or delimiter) in the regular-expression syntax. These are the metacharacters accepted by **findstr**:

.	Wildcard: any character
*	Repeat: zero or more occurrences of previous character or class
^	Line position: beginning of line
\$	Line position: end of line
[<i>class</i>]	Character class: any one character in set
[^ <i>class</i>]	Inverse class: any one character not in set
[<i>x-y</i>]	Range: any characters within the specified range
\ <i>x</i>	Escape: literal use of metacharacter <i>x</i>
\< <i>xyz</i>	Word position: beginning of word
<i>xyz</i> >	Word position: end of word

The special characters in regular expression syntax are most powerful when they are used together. For example, the following combination of the wildcard character (.) and repeat (*) characters

.*

matches any string of characters. This expression is useful when it is part of a larger expression, such as

b.*ing

which matches any string beginning with B and ending with ing.

More Information About Findstr



[Findstr--Examples](#)



[Findstr](#)



Findstr--Examples

To find all occurrences of the word Windows (with an initial capital w) in the file PROPOSAL.TXT, type

```
findstr Windows proposal.txt
```

To search every file in the current directory and all subdirectories that contained the word Windows, regardless of the letter case, type

```
findstr /s /i Windows *.*
```

To find all occurrences of lines that contain the word FOR, preceded by any number of spaces, (as in a computer program loop) and to include the line number where each occurrence is found, type

```
findstr /b /n /c:" *FOR" *.bas
```

If you want to search for several different items in the same set of files, create a text file that contains each search criteria on a new line. You can also list the exact files you want to search in a text file. To use the search criteria in the file FINDDATA.TXT and search the files listed in FILELIST.TXT then store the results in the file RESULTS.OUT, type

```
findstr /g:finddata.txt /f:filelist.txt > results.out
```

Assume you wanted to find every file in the current directory and all subdirectories that contained the word computer, regardless of the letter case. To list every file containing the word computer, type

```
findstr /s /i /m "\<computer\>" *.*
```

Now assume you want to find not only the word computer, but also any other words that begin with the letters comp, such as compliment and compete. Type

```
findstr /s /i /m "\<comp.*" *.*
```

More Information About Findstr



[Findstr--Notes](#)



[Findstr](#)



For

Runs a specified command for each file in a set of files.

You can use the **for** command within a batch program or directly from the command prompt.

To use **for** in a batch program, use the following syntax:

```
for %%variable in (set) do command [command-parameters]
```

To use **for** from the command prompt, use the following syntax:

```
for %variable in (set) do command [command-parameters]
```

Parameters

%%variable or %variable

Represents a replaceable variable. The **for** command replaces %%variable (or %variable) with each text string in the specified set until the command (specified in the command parameters) processes all the files. Use %variable to carry out the **for** command within a batch program. Use %variable to carry out for from the command prompt.

(set)

Specifies one or more files or text strings that you want to process with the specified command. The parentheses are required.

command

Specifies the command that you want to carry out on each file included in the specified set.

command-parameters

Specifies any parameters or switches that you want to use with the specified command (if the specified command uses any parameters or switches).

More Information About For



[For--Notes](#)



[For--Examples](#)



For--Notes

Using the in and do keywords

In and **do** are not parameters, but they are required in the **for** command. If you omit either of these keywords, Windows NT displays an error message.

Using the replaceable variable

To avoid confusion with the batch parameters **%0** through **%9**, you can use any character for *variable* except the numerals 0 through 9. For simple batch programs, a single character such as **%%f** may be all that is necessary.

You can use multiple values for *variable* in complex batch programs to distinguish different replaceable variables.

Specifying a group of files

The set parameter can represent a single group of files or several groups of files. You can use wildcards (* and ?) to specify a file set. The following are valid file sets:

(* .doc)


(* .doc *.txt *.me)

(jan*.doc jan*.rpt feb*.doc feb*.rpt)

(ar??1991.* ap??1991.*)

When you use the **for** command, the first value in *set* replaces **%%variable** (or **%variable**) and Windows NT carries out the specified command in order to process this value; this continues until Windows NT has processed all the files (or groups of files) that correspond to the value (or values) in *set*.

More Information About For

 [For--Examples](#)

 [_For](#)



For--Examples

Suppose you want to use the **type** command to display the contents of all the files in the current directory that have the extension .DOC or .TXT. To do this and to use the replaceable variable **%f**, type the following command at the command prompt:


```
for %f in (*.doc *.txt) do type %f
```

In this example, each file that has the .DOC or .TXT extension in the current directory is substituted for the **%f** variable until the contents of every file are displayed. To use this command in a batch file, you simply replace every occurrence of **%f** with **%%f**. Otherwise, Windows NT ignores the variable and displays an error message.

Windows NT supports command switches, pipes, and redirection that you may want to use with the specified command. For example, to redirect the output of the previous example to PRN (the default printer port), you would type the following command:

```
for %f in (*.doc *.txt) do type %f > prn:
```

More Information About For

 [For--Notes](#)

 [For](#)



Forcedos

Starts the specified program in the MS-DOS subsystem. This command is necessary only for those MS-DOS programs not recognized as such by Windows NT.

forcedos [/d *directory*] *filename* [*parameters*]

Parameters

/d directory

Specifies the current directory for the specified program to use.

filename

Specifies the program to start. If not in the current directory or Windows NT path, you must specify the drive letter or directories to the program.

parameters

Specifies parameters to pass to the program.

More Information About Forcedos



[Forcedos--Notes](#)



[Forcedos--Examples](#)




Forcedos--Notes

Some MS-DOS programs are not recognized by Windows NT as MS-DOS programs. The **forcedos** command allows you to start these programs.

All programs started by the program will be launched as MS-DOS applications.

More Information About Forcedos

 [Forcedos--Examples](#)

 [Forcedos](#)





Forcedos--Example

To start the program MYPROG in the \OLDAPPS directory and use the \REPORTS directory type

```
forcedos /d reports \oldapps\myprog
```

More Information About Forcedos

 [Forcedos--Notes](#)

 [Forcedos](#)



Format

Formats the disk in the specified drive to accept Windows NT files. You must be a member of the Administrators group to format a hard drive.

format *drive*: [/fs:*file-system*] [/v[:*label*]] [/a:*unitsize*] [/q] [/f:*size*] [/t:*tracks* /n:*sectors*] [/1] [/4] [/8]

Parameters

drive:

Specifies the drive containing the disk you want to format. If you do not specify any of the following switches, **format** uses the drive type to determine the default format for the disk.

/fs:file-system

Specifies the file system to use, FAT or NTFS. Floppy disks can use only the FAT file system.

/v:label

Specifies the volume label. If you omit the */v* switch, or use it without specifying a volume label, Windows NT prompts you for the volume label after the formatting is completed. Use the syntax, */v:* to prevent the prompt for a volume label. If you format more than one disk by using one **format** command, all of the disks will be given the same volume label. The */v* switch is not compatible with the */8* switch. For more information about disk volume labels, see the **dir**, **label**, and **vol** commands.

/a:unitsize

Specifies the allocation unit size to use on NTFS disks. Use one of the following values for *unitsize*. If *unitsize* is not specified, it will be chosen based on disk size.

512	Creates 512 bytes per cluster, 1024 bytes per file record. Default if disk is less than 512 meg.
1024	Creates 1024 bytes per cluster, 1024 bytes per file record. Default if disk is 512MB to 1G.
2048	Creates 2048 bytes per cluster, 2048 bytes per file record. Default if disk is 1G to 2G.
4096	Creates 4096 bytes per cluster, 4096 bytes per file record. Default if disk is over 2G.

/q

Deletes the file table and the root directory of a previously formatted disk, but does not scan the disk for bad areas. You should use the */q* switch to format only previously formatted disks that you know are in good condition.

/f:size

Specifies the size of the floppy disk to format. When possible, use this switch instead of the */t* and */n* switches. Use one of the following values for size:

160 or 160k or 160kb

160K, single-sided, double-density, 5.25-inch disk

180 or 180k or 180kb

180K, single-sided, double-density, 5.25-inch disk

320 or 320k or 320kb

320K, double-sided, double-density, 5.25-inch disk

360 or 360k or 360kb

360K, double-sided, double-density, 5.25-inch disk

720 or 720k or 720kb

720K, double-sided, double-density, 3.5-inch disk

1200 or 1200k or 1200kb or 1.2 or 1.2m or 1.2mb

1.2-MB, double-sided, quadruple-density, 5.25-inch disk

1440 or 1440k or 1440kb or 1.44 or 1.44m or 1.44mb

1.44-MB, double-sided, quadruple-density, 3.5-inch disk

2880 or 2880k or 2880kb or 2.88 or 2.88m or 2.88mb

2.88-MB, double-sided, 3.5-inch disk

20.8 or 20.8m or 20.8mb

20.8-MB, 3.5-inch floptical disk

/t:tracks

Specifies the number of tracks on the disk. When possible, use the **/f** switch instead of this switch. If you use the **/t** switch, you must also use the **/n** switch. These two switches provide an alternative method of specifying the size of the disk being formatted. You cannot use the **/f** switch with the **/t** switch.

/n:sectors

Specifies the number of sectors per track. When possible, use the **/f** switch instead of this switch. If you use the **/n** switch, you must also use the **/t** switch. These two switches provide an alternative method of specifying the size of the disk being formatted. You cannot use the **/f** switch with the **/n** switch.

/1

Formats a single side of a floppy disk.


/4


Formats a 5.25-inch, 360K, double-sided, double-density floppy disk on a 1.2-MB disk drive. Some 360K drives cannot reliably read disks formatted with this switch. When used with the **/1** switch, this switch formats a 5.25-inch, 180K, single-sided floppy disk.

/8

Formats a 5.25-inch disk with 8 sectors per track. This switch formats a floppy disk to be compatible with MS-DOS versions prior to 2.0.

More Information About Format

 [Format--Notes](#)

 [Format--Examples](#)



Format--Notes

Using Format

The format command creates a new root directory and file system for the disk. It can also check for bad areas on the disk, and it can delete all data on the disk. In order for Windows NT to be able to use a new disk, you must first use this command to format the disk.

Typing a volume label

After formatting a floppy disk, format displays the following message:

Volume label (11 characters, ENTER for none)?

The volume label can be a maximum of 11 characters (including spaces). If you do not want your disk to have a volume label, just press ENTER. For information about volume labels, see the label command.

Formatting a hard disk

When you use the format command to format a hard disk, Windows NT displays a message of the following form before attempting to format the hard disk:

```
WARNING, ALL DATA ON NON-REMOVABLE DISK  
DRIVE x: WILL BE LOST!  
Proceed with Format (Y/N)?_
```

To format the hard disk, press Y; if you do not want to format the disk, press N. You must have Administrator rights to format a hard disk.

Format messages

When formatting is complete, Windows NT displays messages showing the total disk space, any space marked as defective, and the space available for your files.

Quick formatting

You can speed up the formatting process by using the /q switch. Use this switch only if there are no bad sectors on your hard disk.

Using format with a reassigned drive or a network drive

You should not use the format command on a drive prepared by using the subst command. You cannot format disks over a network.

Format exit codes


The following list shows each exit code and a brief description of its meaning:

- | | |
|---|---|
| 0 | The format operation was successful. |
| 1 | Incorrect parameters were supplied. |
| 4 | A fatal error occurred (any error other than 0, 1, or 5). |
| 5 | The user pressed N in response to the |

prompt
"Proceed
with Format
(Y/N)?" to
stop the
process.

You can check these exit codes by using the **errorlevel** condition with the **if** batch command. For an example of a batch program that supports **errorlevel** conditions, see the **backup** command.

More Information About Format

 [Format--Examples](#)

 [Format](#)



Format--Examples

To format a new floppy disk in drive A, using the default size, type the following command:

format a:

To perform a quick format operation on a previously formatted disk in drive A, type the following command:

format a: /q

To format a floppy disk in drive A and assign it the volume label "DATA", type the following command:

format a: /v:DATA

More Information About Format



[Format--Notes](#)



[Format](#)



Ftype

Displays or modifies file types used in file extension associations.

Ftype [**filetype**[=**command**]]

Parameter

filetype

Specifies the file type you want to display or change.

command

Specifies the open command to use when launching files of this type.

Type **ftype** without options to display the current file types that have open command strings defined.

Type **ftype filetype** to display the current open command string for this file type.

Type **ftype filetype=** to delete the open command string for the file type.

Within an open command string, **ftype** substitutes the following variables:

%0 or %1 are substituted with the file name being launched.

%* is substituted with all the parameters

%3 is substituted with the first parameter, %4 with the second, and so forth.



Goto

Directs Windows NT to a line in a batch program marked by a label you specify.

The **goto** command directs Windows NT within a batch program to a line identified by a label. When Windows NT finds the label, it processes the commands beginning on the next line.

goto *label*

Parameter

label

Specifies the line in a batch program to which Windows NT should go.

More Information About Goto



[Goto--Notes](#)



[Goto--Examples](#)



Goto--Notes

Valid values for label

The *label* parameter can include spaces but cannot include other separators, such as semicolons or equal signs.

Goto uses the first eight characters of each label

The **goto** command uses only the first eight characters of a label. Therefore, the labels ":hithere01" and ":hithere02" are both equivalent to ":hithere0".

Matching the label parameter with the label in the batch program

The label value you specify on the **goto** command line must match a label in the batch program. The label within the batch program must begin with a colon.

If your batch program does not contain the label that you specify, the batch program stops and Windows NT displays the following message:

Label not found

Windows NT recognizes a batch-program line beginning with a colon (:) as a label and does not process it as a command. If a line begins with a colon, Windows NT ignores any commands on that line.

Using goto for conditional operations

Goto is often used on the same command line with other commands to perform conditional operations. For more information about using **goto** for conditional operations, see the **if** command.

More Information About Goto



[Goto--Examples](#)



[Goto](#)



Goto--Example

The following batch program formats a disk in drive A as a system disk. If the operation is successful, the **goto** command directs Windows NT to a label named "end".

```
echo off
format a: /s
if not errorlevel 1 goto end
echo An error occurred during formatting.
:end
echo End of batch program.
```

More Information About Goto



[Goto--Notes](#)



[Goto](#)



Graftabl

Enables Windows NT to display the extended characters of a specified code page in full-screen mode. Extended characters will not display in window mode.

graftabl [xxx] [/status]

Parameters

xxx

Specifies the code page for which you want Windows NT to define the appearance of extended characters in graphics mode. The following list shows each valid code-page identification number and its country or language:

437 United States

850 Multilingual (Latin I)

852 Slavic (Latin II)

855 Cyrillic (Russian)

857 Turkish

860 Portuguese

861 Icelandic

863 Canadian-French

865 Nordic

866 Russian

869 Modern Greek

/status

Identifies the code page selected for use by **graftabl**.

See Also

For information about using code pages, see the [chcp](#) and [mode \(set device code pages\)](#) commands.

More Information About Graftabl



[Graftabl--Notes](#)



[Graftabl--Examples](#)



Graftabl--Notes

Graftabl does not change the console input code page. **Graftabl** affects only the monitor display of extended characters of the code page you specify. To change the code page you are using, use the **mode** or **chcp** command.

Graftabl exit codes

The following list shows each exit code and a brief description of its meaning:

- | | |
|---|--|
| 0 | Character set was loaded successfully; no previous code page was loaded. |
| 1 | An incorrect parameter was specified; no action was taken. |
| 2 | A file error occurred. |

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by **grftabl**. For an example of a batch program that processes exit codes, see the **backup** command.

More Information About Graftabl



[Graftabl--Examples](#)



[Graftabl](#)



Graftabl--Example

To load the graphics character set for code page 437 (United States) into memory, type

graftabl 437

To load the graphics character set for code page 850 (multilingual) into memory, type

graftabl 850

More Information About Graftabl



[Graftabl--Notes](#)



[Graftabl](#)



Graphics

Loads a program into memory that allows Windows NT to print on a printer the displayed contents of the screen when you are using a color or graphics adapter.

graphics [*type*] [[*drive:*][*path*] *filename*] [/r] [/b] [/lcd] [/printbox:std | /printbox:lcd]

Parameters

type

Specifies the type of printer. The following list shows each valid value for this parameter and a brief description of its meaning:

Parameter	Description
color1	An IBM Personal Computer Color Printer with black ribbon
color4	An IBM Personal Computer Color Printer with RGB (red, green, blue, and black) ribbon
color8	An IBM Personal Computer Color Printer with CMY (cyan, magenta, yellow, and black) ribbon
hpdefault	Any Hewlett-Packard PCL printer
deskjet	A Hewlett-Packard DeskJet printer
graphics	An IBM Personal Graphics Printer, IBM Proprinter, or IBM Quietwriter printer
graphicswide	An IBM Personal Graphics Printer with an 11-inch-wide carriage
laserjet	A Hewlett-Packard LaserJet printer
laserjetii	A Hewlett-Packard LaserJet II printer
paintjet	A Hewlett-Packard PaintJet printer
quietjet	A Hewlett-Packard QuietJet printer
quietjetplu	A Hewlett-Packard QuietJet Plus printer
ruggedwriter	A Hewlett-Packard RuggedWriter printer
ruggedwriterwide	A Hewlett-Packard RuggedWriterwide

	printer
thermal	An IBM PC-convertible Thermal Printer
thinkjet	A Hewlett-Packard ThinkJet printer

[drive:][path] filename

Specifies the location and name of the printer profile that contains information about all supported printers. If this parameter is omitted, Windows NT looks for a file called GRAPHICS.PRO in the current directory and in the directory that contains the GRAPHICS.COM file.

/r

Prints the image as it appears on the screen (white characters on a black background) rather than reversed (black characters on a white background). The latter occurs by default.

/b

Prints the background in color. This switch is valid for **color4** and **color8** printers.

/lcd

Prints an image by using the liquid crystal display (LCD) aspect ratio instead of the CGA aspect ratio. The effect of this switch is the same as that of **/printbox:lcd**.

/printbox:std | /printbox:lcd

Selects the print-box size. You can abbreviate **printbox** as **pb**. You should check the first operand of the **printbox** statement in your GRAPHICS.PRO file and specify the **/printbox:std** switch if that operand is **std** or the **/printbox:lcd** switch if that operand is **lcd**.

See Also

 [Print](#)

More Information About Graphics

 [Graphics--Notes](#)

 [Graphics--Examples](#)



Graphics--Notes

Printing the contents of the screen

To print the contents of the screen, press the SHIFT+PRINT SCREEN key combination. If the computer is in 320 x 148 x 200 color graphics mode and if the printer type is **color1** or **graphics**, the **graphics** command prints the screen contents with as many as four shades of gray. If the computer is in 640 x 148 x 200 color graphics mode, **graphics** prints the screen contents sideways on the paper (landscape orientation). You cannot use the SHIFT+PRINT SCREEN key combination to print the contents of a screen to a PostScript printer.

Effect on memory

The **graphics** command decreases the amount of available conventional memory.

Loading a new profile

If you have already loaded a printer profile and you want to load another one by using the **graphics** command, the new profile must be smaller than the one already loaded.


To load a new profile that is larger than the one currently loaded, you must restart your system and then use the **graphics** command to load the new profile.

If you try to use only the **graphics** command to load a new profile that is larger than the currently loaded profile, Windows NT displays the following message:

Unable to reload with profile supplied

More Information About Graphics

 [Graphics--Examples](#)

 [Graphics](#)



Graphics--Example

To prepare to print a graphics screen on your printer, type the following command:

graphics

After you display the information you want to print, press SHIFT+PRINT SCREEN. Windows NT scans the information displayed on the screen and sends it to the printer.

More Information About Graphics



[Graphics--Notes](#)



[Graphics](#)



Help

Provides online information about Windows NT commands (non-network).

help [*command*]

Parameter

command

Specifies the name of the command about which you want information. If you do not specify a command name, the **help** command lists and briefly describes every Windows NT system command.

Note



There are two ways to get online Help for a command. You can specify the name of the command on the **help** command line, or you can type the name of the command and the **/?** switch at the command prompt. For example, you can type either of the following commands to get information about the **xcopy** command:

help xcopy

xcopy /?

The second command is slightly faster.

See **net help** for information about help with network commands.



If

Performs conditional processing in batch programs. If the condition specified in an **if** command is true, Windows NT carries out the command that follows the condition. If the condition is false, Windows NT ignores the command.

if [**not**] **errorlevel** *number* *command*

if [**not**] *string1*==*string2* *command*

if [**not**] **exist** *filename* *command*

Parameters

not

Specifies that Windows NT should carry out the command only if the condition is false.

errorlevel *number*

Specifies a true condition only if the previous program run by CMD.EXE returned an exit code equal to or greater than *number*.

command

Specifies the command that Windows NT should carry out if the preceding condition is met.


string1==*string2*


Specifies a true condition only if *string1* and *string2* are the same. These values can be literal strings or batch variables (%**1**, for example). Literal strings do not need quotation marks.

exist *filename*

Specifies a true condition if *filename* exists.

More Information About If

 [If--Notes](#)


 [If--Examples](#)



If--Notes

When a program stops, it returns an exit code to Windows NT. The **errorlevel** parameter lets you use exit codes as conditions.

More Information About If

 [If--Examples](#)

 [If](#)



If--Examples

The following example displays the message "Can't find data file" if Windows NT cannot find the file PRODUCT.DAT:

```
if not exist product.dat echo Can't find data file
```


The following example displays an error message if an error occurs during formatting of the disk in drive A. If no error occurs, the error message is skipped.

```
:begin
@echo off
format a: /s
if not errorlevel 1 goto end
echo An error occurred during formatting.
:end
echo End of batch program.
```

The following example tests for the existence of a directory. The **if** command cannot be used to test directly for a directory, but the null (NUL) device does exist in every directory. Therefore, you can test for the null device to determine whether a directory exists.

```
if exist c:\mydir\nul goto process
```

More Information About If

 [If--Notes](#)

 [_If](#)



Install

Loads a memory-resident program into memory.

install=[*drive:*][*path*] *filename* [*command-parameters*]

Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of the memory-resident program you want to run.

command-parameters

Specifies parameters for the program you specify for filename.

More Information About Install



[Install--Notes](#)



[Install--Examples](#)



Install--Note

Install should be used to load a memory resident program into memory just prior to starting the application that requires the memory resident program. If a memory resident program is loaded into memory using the AUTOEXEC.NT file, the memory resident program will be loaded into memory every time the command prompt is started, wasting memory.

Do not use **Install** to load programs that use environment variables or shortcut keys or that require CMD.EXE to be present to handle critical errors.

More Information About Install



[Install--Example](#)



[Install](#)



Install--Example

To load a program into memory in the CONFIG.NT file, type

install

More Information About Install

 [Install--Notes](#)

 [Install](#)



ipxroute

Displays and modifies information about the routing tables used by the IPX protocol. The command has different options for IPX routing and for source routing. Separate all options with spaces.

IPX Routing Options

ipxroute servers [/type=x]

ipxroute stats [/show] [/clear]

ipxroute table

Parameters

servers [/type=x]

Displays the SAP table for the specified server type. x is an integer. For example, /**type=4** displays all file servers. If no /**type** is specified, then servers of all types are shown. The list is sorted by server name.

stats [/show] [/clear]

Displays or clears IPX router interface statistics. /**show** is the default. /**clear** clears the statistics.

table

Displays the IPX routing table, sorted by network number.

Source Routing Options **ipxroute board=n** [clear] [def] [gbr] [mbr] [remove=xxxxx]

ipxroute config

Parameters

board=n

Specifies the network adapter card for which to query or set parameters.

clear

Clears the source routing table

def

Sends packets to the ALL ROUTES broadcast. If a packet is transmitted to a unique mac address that is not in the source routing table, the default is to send the packet to the SINGLE ROUTES broadcast.

gbr

Sends packets to the ALL ROUTES broadcast. If a packet is transmitted to the broadcast address (FFFFFFFFFFFF), the default is to send the packet to the SINGLE ROUTES broadcast.

mbr

Sends packets to the ALL ROUTES broadcast. If a packet is transmitted to a multicast address (C000xxxxxxx), the default is to send the packet to the SINGLE ROUTES broadcast.

remove=xxxxx

Removes the given node address from the source routing table.

config

Displays information on all the bindings for which IPX is configured.



Keyb

Starts the **keyb** program, which configures a keyboard for a specific language.

Use **keyb** to configure a keyboard for a language other than United States English.

keyb [xx[,yyy][, [drive:][path] filename]] [/e] [/id:nnn]

In your CONFIG.NT file, use the following syntax:

install=[drive:]path**keyb.com** [xx[,yyy][, [drive:][path] filename]] [/e] [/id:nnn]

Parameters

xx

Specifies the keyboard code. See the table at the end of this topic for a list of valid values for this parameter.

yyy

Windows NT and the MS-DOS subsystem do not use this parameter. It is accepted only for compatibility with MS-DOS files.

[drive:][path] filename

Windows NT and the MS-DOS subsystem do not use this parameter. It is accepted only for compatibility with MS-DOS files.

[drive:]path

Windows NT and the MS-DOS subsystem do not use this parameter. It is accepted only for compatibility with MS-DOS files.

/e

Windows NT and the MS-DOS subsystem do not use this parameter. It is accepted only for compatibility with MS-DOS files.

/id:nnn

Windows NT and the MS-DOS subsystem do not use this parameter. It is accepted only for compatibility with MS-DOS files.

Values for xx

The following table shows the valid values for xx for each country or language:


Country or language	Keyboard code (xx value)
Belgium	be
Brazil	br
Canadian-French	cf
Czech	cz
Slovak	sl
Denmark	dk
Finland	su
France	fr
Germany	gr
Hungary	hu
Italy	it
Latin America	la
Netherlands	nl


Norway	no
Poland	pl
Portugal	po
Spain	sp
Sweden	sv
Switzerland (French)	sf
Switzerland (German)	sg
United Kingdom	uk
United States	us
Serbo-Croatian	yu

See Also

 [_Chcp](#)

More Information About Keyb

 [_Keyb--Notes](#)

 [_Keyb--Examples](#)



Keyb--Note

Displaying the current keyboard code and code page

If you use the **keyb** command with no parameters or switches, Windows NT lists the current keyboard code, the current keyboard's related code page, and the current code page used by your console (CON). The information is displayed in the following format:

Current keyboard layout: Language 12 Sublanguage 0

Current CON code page: 437

Implementing Keyb

The following list shows the three different ways that you can start the Keyb program:



Type **keyb** at the command prompt.



Include an **install** command for KEYB.COM in your CONFIG.NT file.



Include the appropriate **keyb** command in your AUTOEXEC.NT file.

Keyb exit codes

The following list shows each exit code and a brief description of its meaning:

- | | |
|---|---|
| 0 | Keyboard definition
file was loaded
successfully. |
| 1 | Invalid keyboard
code, code page, or
syntax was used. |

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by Keyb. For an example of a batch program that processes exit codes, see the [diskcomp](#) command.

More Information About Keyb



[Keyb--Examples](#)



[_Keyb](#)




Keyb--Example

If you want to use a German keyboard, type the following command:

keyb gr

More Information About Keyb

 [Keyb--Notes](#)

 [Keyb](#)



Label

Creates, changes, or deletes the volume label (name) of a disk.

Windows NT displays the volume label as part of the directory listing. If a volume serial number exists, Windows NT displays this number as well.

label [*drive:*][*label*]

Parameters

none

Type **label** without parameters to change the current volume label or delete the existing label.

drive:

Specifies the location of the disk you want to name.

label


Specifies the new volume label. You must include a colon (:) between drive and label.


See Also

 [Dir](#)

 [Vol](#)

More Information About Label

 [Label--Notes](#)

 [Label--Examples](#)



Label--Notes

Label command messages

If you do not specify a label when you use the **label** command, Windows NT displays a message in the following format:

```
Volume in drive A is xxxxxxxxxxxx  
Volume Serial Number is xxxx-xxxx  
Volume label (11 characters, ENTER for none)?
```

The "Volume Serial Number" part of the message is not displayed if the disk has no serial number.

You can type the volume label you want or press ENTER to delete the current label. If a disk has a label and you press ENTER for none, Windows NT prompts you with the following message:

```
Delete current volume label (Y/N)?
```

Press Y to delete the label; press N to keep the label.

Limitations on volume label names

A volume label can contain as many as 32 characters for NTFS volumes and as many as 11 characters for FAT volumes and can include spaces but no tabs.

Do not use any of the following characters in a volume label:

```
* ? / \ | . , ; : + = [ ] ( ) < > "
```

Windows NT displays FAT volume labels in uppercase letters. If you enter a volume label in lower-case letters, the **label** command converts the letters to uppercase. NTFS volume labels retain and display the case used when the label was created.

More Information About Label



[Label--Examples](#)



[Label](#)




Label--Example

To label a disk in drive A that contains sales information for July, you might type the following:

label a:sales-july

More Information About Label

 [Label--Notes](#)

 [Label](#)



Libpath

Specifies the directories the OS/2 subsystem is to search for dynamic-link libraries. To use this command, use an OS/2 editor to edit the C:\CONFIG.SYS file.

libpath=[*drive:*]*path*[:,*drive:*]*path*][...]

Parameters

drive:

Specifies the drive where dynamic-link libraries are located. If you do not specify a drive, the OS/2 subsystem searches the disk in the current drive.

path

Specifies the directory to search for dynamic-link libraries. You can specify more than one directory, separating the names with semicolons (;).

More Information About Libpath



[Libpath--Example](#)



Libpath--Example

To instruct the OS/2 subsystem to look for dynamic-link libraries in the current directory and in the directory DYNLIB on your hard disk (drive C), add the following line in your C:\CONFIG.SYS file using an OS/2 editor:

```
libpath=.;c:\dynlib
```

More Information About Libpath



[Libpath](#)



Lastdrive

Windows NT and the MS-DOS subsystem do not take action for this command. It is accepted only for compatibility with MS-DOS.

Drives are always set to the maximum (Z).



Loadfix

Ensures that a program is loaded above the first 64K of conventional memory, and runs the program.

loadfix [*drive:*][*path*] *filename* [*program-parameters*]

[*drive:*][*path*]

Specifies the drive and directory of the program.

filename

Specifies the name of the program.

program-parameters

Specifies any of the program's parameters that you want to use.

More Information About Loadfix



[Loadfix--Notes](#)



[Loadfix--Examples](#)



Loadfix--Note

Some programs will display the "Packed file corrupt" message when all or a portion of the program has been loaded in the first 64K of conventional memory and cannot run successfully. This error is most likely to occur when you load device drivers into the upper memory area, thereby freeing more of the first 64K of conventional memory. If Windows NT displays this message, use the **loadfix** command to ensure that a program is loaded above the first 64K of virtual conventional memory.

To use **loadfix**, include it at the beginning of the command that starts the program.

More Information About Loadfix



[Loadfix](#)



[Loadfix--Examples](#)



Loadfix--Example

To load a program named MYAPP.EXE (in the APPS directory of drive C) and use the /c parameter (for example, to load the program in character mode), type the following at the command prompt or include the line in a batch program:

```
loadfix c:\apps\myapp.exe /c
```

More Information About Loadfix



[Loadfix--Notes](#)



[Loadfix](#)



Loadhigh (Lh)

Loads a program into the upper memory area. Loading a program into the upper memory area leaves more room in conventional memory for other programs.

loadhigh [*drive:*][*path*] *filename* [*parameters*]

lh [*drive:*][*path*] *filename* [*parameters*]

Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of the program you want to load.

parameters

Specifies any command-line information required by the program.

More Information About Loadhigh



[Loadhigh--Notes](#)



[Loadhigh--Examples](#)



Loadhigh (lh)--Notes

DOS=UMB command required

To use the **loadhigh** command, you must include the **dos=umb** command in your CONFIG.NT or equivalent startup file. For more information about the **dos=umb** command, see the **dos** command.

Upper-memory-area manager must be installed

Before you can load a program into the upper memory area, you must install the HIMEM.SYS extended-memory manager. Use the **device** command in your CONFIG.NT or equivalent startup file to install HIMEM.SYS.


How loadhigh works


If you use the **loadhigh** command to load a program, Windows NT attempts to load it into the upper memory area. If there is insufficient space in the upper memory area, Windows NT loads the program into conventional memory. Windows NT does not indicate which memory area is used.

Using loadhigh in your AUTOEXEC.NT file

The most convenient way to use the **loadhigh** command is to include it in your AUTOEXEC.NT or equivalent startup file.

More Information About Loadhigh

 [Loadhigh--Examples](#)

 [Loadhigh](#)




Loadhigh (lh)--Example

To load a driver into high memory, include the following line in your AUTOEXEC.NT or equivalent startup file:

```
lh %systemroot%\system32\dosx
```

More Information About Loadhigh

 [Loadhigh--Notes](#)

 [Loadhigh](#)



Mem

Displays information about allocated memory areas, free memory areas, and programs that are currently loaded into memory in the MS-DOS subsystem.

mem [/program|/debug|/classify]

Parameters

none

Type **mem** without parameters to display the status of the MS-DOS subsystem's used and free memory

/program

Displays the status of programs that are currently loaded into memory. You cannot use the **/program** switch with the **/debug** switch or the **/classify** switch. You can abbreviate **/program** as **/p**.

/debug

Displays the status of currently loaded programs and of internal drivers, and displays other programming information. You cannot use the **/debug** switch with the **/program** switch or the **/classify** switch. You can abbreviate **/debug** as **/d**.

/classify

Displays the status of programs loaded into conventional memory and the upper memory area. This switch lists the size of each program in decimal and hexadecimal notation, provides a summary of memory use, and lists the largest memory blocks that are available. You cannot use the **/classify** switch with the **/program** switch or the **/debug** switch. You can abbreviate **/classify** as **/c**.

More Information about Mem



[Mem--Notes](#)



[Mem--Examples](#)



Mem--Notes

Displaying memory status

The MS-DOS subsystem displays the status of extended memory only if you have allocated memory above 1-megabyte (MB).

More Information about Mem



[Mem--Examples](#)



[Mem](#)



Mem--Example

Suppose your MS-DOS subsystem is configured with expanded memory. To display the status of your system's total memory--conventional, expanded, extended--and to display a list of programs currently loaded into memory, type the following command:

```
mem /program
```

The results might look similar to the following:

Address	Name	Size	Type
-----	-----	-----	-----
000000		000400	Interrupt Vector
000400		000100	ROM Communication Area
000500		000200	DOS Communication Area
000700	IO	000310	System Data
000A10	MSDOS	0014D0	System Data
001EE0	IO	0018D0	System Data
	KBD	000800	System Program
	HIMEM	000420	DEVICE=
		000340	FILES=
		000090	FCBS=
		000170	LASTDRIVE=
		000710	STACKS=
0037C0	COMMAND	000A40	Program
004210	MSDOS	000070	-- Free --
004290	COMMAND	0001F0	Environment
004490	MEM	0001D0	Environment
004670	MEM	017550	Program
01BBD0	MSDOS	084410	-- Free --
09FFF0	SYSTEM	028000	System Program
0C8000	IO	0083D0	System Data
	MOUSE	0083C0	System Program
0D03E0	MSDOS	000050	-- Free --
0D0440	REDIR	0009F0	Program
0D0E40	DOSX	007CA0	Program
0D8AF0	DOSX	001030	Data
0D9B30	MSDOS	0164C0	-- Free --

```
655360 bytes total conventional memory
655360 bytes available to MS-DOS
637296 largest executable program size


1048576 bytes total contiguous extended memory
0 bytes available contiguous extended memory
405504 bytes available XMS memory
MS-DOS resident in High Memory Area
```

"Total conventional memory" is the amount of virtual memory allocated to the MS-DOS subsystem up to the first 640K. "Available to MS-DOS subsystem" is the amount of conventional memory allocated, including the memory neededs for CMD.EXE. "Largest executable program size" is the largest contiguous block of conventional memory available for a program.

"Total EMS memory" (not shown in the preceding example) is the amount of expanded memory configured for the MS-DOS subsystem. "Free EMS memory" is the amount of expanded memory available for programs.

"Total contiguous extended memory" is the amount of memory beyond 1-megabyte (MB). "Available contiguous extended memory" is the extended memory available for the Interrupt 15h interface. This memory is not being managed by an extended-memory manager, such as HIMEM.SYS. Some older programs use this different extended-memory scheme. "Available XMS memory" is memory that is being managed by an extended-memory manager, such as HIMEM.SYS, and that is available to programs that can use it.

More Information about Mem

 [Mem--Notes](#)

 [Mem](#)



Mkdir (md)

Creates a directory or subdirectory.

mkdir [*drive:*]path

md [*drive:*]path

Parameters

drive:

Specifies the drive on which you want to create the new directory.

path

Specifies the name and location of the new directory. The maximum length of any single path is determined by the file system.

More Information About Mkdir



[Mkdir-Examples](#)



Mkdir (md)--Examples

Suppose you want to create a directory on the disk in the current drive and use the directory to store all your tax information. To create a directory named TAXES, type the following command:

```
mkdir taxes
```

Now suppose that the TAXES directory is the current directory and that you want to create a subdirectory of TAXES named PROPERTY. To create the PROPERTY directory, type the following command:

```
mkdir property
```

More Information About Mkdir

 [Mkdir](#)









Mode

Configures system devices. The **mode** command performs many different tasks, such as displaying system status, changing system settings, or reconfiguring ports or devices.

Using the mode command

Because the **mode** command can perform many different tasks, the syntax necessary to carry out each task is different. Therefore, this reference discusses the tasks separately. The following is a list of tasks for which you can use the **mode** command.

-  Reconfiguring a printer attached to a parallel port (PRN, LPT1, LPT2, or LPT3) for printing at 80 or 132 characters per line, 6 or 8 lines per inch, or both (if the printer supports these features). See [MODE \(configure printer\)](#) for more information.
-  Configuring the baud rate, parity, and number of data bits and stop bits of a serial communications port (COM1, COM2, COM3, and COM4) for use with a specific printer, modem, or other serial device. See [MODE \(configure serial port\)](#) for more information.
-  Displaying the status of all devices or of a single device. See [MODE \(display device status\)](#) for more information.
-  Redirecting printer output from a parallel port to a serial port so that the serial port becomes the system's default printer port. See [MODE \(redirect printing\)](#) for more information.
-  Changing the size of the command prompt window. See [MODE \(set display mode\)](#).
-  Setting the keyboard's typematic rate. See [MODE \(set typematic rate\)](#).



Mode (Configure Printer)

Configures a printer connected to a parallel printer port.

This version of the **mode** command sets the characteristics for an IBM-compatible or Epson-compatible printer connected to a parallel printer port (PRN, LPT1, LPT2, or LPT3).

mode *lptn*[:] [*c*][,][*l*][,*r*]

mode *lptn*[:] [**cols**=*c*] [**lines**=*l*]

Parameters

lptn

Specifies the parallel port to which the device is attached.

cols=c

Specifies the number of characters (columns) per line: 80 or 132. The default value is 80. You can abbreviate this parameter by simply omitting **cols=** and specifying a value for *c*.

lines=l

Specifies the vertical spacing and the number of lines per inch: 6 or 8. The default value is 6. You can abbreviate this parameter by simply omitting **lines=** and specifying a value for *l*.

More Information About Mode (Configure Printer)



[_Mkdir \(Configure Printer\)--Examples](#)



Mode (Configure Printer)--Examples

Suppose you want to be able to print 80 characters per line and 8 lines per inch on a parallel printer that is connected to the second parallel printer port (LPT2). To do this, type the following command:

```
mode lpt2:80,8
```

Because 80 characters per line is the default setting, however, you could achieve the same result typing the following command:

```
mode lpt2:,8
```

More Information About Mode (Configure Printer)



[Mode \(Configure Printer\)](#)



Mode (Configure Serial Port)

Configures a serial communications port.

This version of the **mode** command sets the parameters for a serial port (COM1, COM2, COM3, etc).

mode *comm*[:] [**baud**=*b*] [**parity**=*p*] [**data**=*d*] [**stop**=*s*] [**to**=**on|off**] [**xon**=**on|off**] [**odsr**=**on|off**] [**octs**=**on|off**] [**dtr**=**on|off|hs**] [**rts**=**on|off|hs|tg**] [**idsr**=**on|off**]

Parameters

comm

Specifies the number of the asynchronous-communications (COM) port.

baud=*b*

Specifies the transmission rate in bits per second. The following list shows valid abbreviations for *b* and its related rate:

11	110 baud
15	150 baud
30	300 baud
60	600 baud
12	1200 baud
24	2400 baud
48	4800 baud
96	9600 baud
19	19,200 baud

parity=*p*

Specifies how the system uses the parity bit to check for transmission errors. The *p* value can be one of the following: **n** (none), **e** (even), **o** (odd), **m** (mark), or **s** (space). The default value is **e**. Not all computers support the values **m** and **s**.

data=*d*

Specifies the number of data bits in a character. Valid values for **d** are in the range 5 through 8. The default value is 7. Not all computers support the values 5 and 6.

stop=*s*

Specifies the number of stop bits that define the end of a character: 1, 1.5, or 2. If the baud rate is 110, the default value is 2; otherwise, the default value is 1. Not all computers support the value 1.5.

to=**on|off**

Specifies whether infinite time-out processing is on or off. The default is off.

xon=**on|off**

Specifies whether the xon or xoff protocol for data-flow control is on or off.

odsr=**on|off**

Specifies whether output handshaking that uses the Data Set Ready (DSR) circuit is on or off.

octs=**on|off**

Specifies whether output handshaking that uses the Clear To Send (CTS) circuit is on or off.

dtr=**on|off**

Specifies whether the DTR circuit is on or off.

rts=**on|off|hs|tg**

Specifies whether the RTS circuit is set to on, off, handshake, or toggle.

idsr=**on|off**

Specifies whether the DSR circuit sensitivity is on or off.



Mode (Display Device Status)

Displays the status of one or all of the devices installed on your system.

mode [*device*] [**/status**]

Parameters

none

Used without parameters, **mode** displays the status of all devices installed on your system.

device

Specifies the name of the device for which you want to display the status.

/status

Requests the status of any redirected parallel printers. The **mode** command, when used without this switch, displays the status of all installed devices except redirected parallel printers. You can abbreviate the **/status** switch as **/sta**.



Mode (Redirect Printing)

Redirects output from a parallel port to a serial communications port.

You must be a member of the Administrators group to redirect printing.

mode **lpt***n*[:]=**com***m*[:]

Parameters

lpt*n*

Specifies the parallel port. Valid values for *n* are in the range 1 through 3.

com*m*

Specifies the serial port. Valid values for *m* are in the range 1 through 4.

More Information About Mode (Redirect Printing)



[_Mkdir \(Redirect Printing\)--Examples](#)



Mode (Redirect Printing)--Examples

Suppose you want to set up your system so that it sends parallel-printer output to a serial printer. To do this, you must use the **mode** command twice. The first time, you use **mode** to configure the serial port; the second time, you use **mode** to redirect parallel-printer output to the serial port you specified in the first **mode** command.

For example, if your serial printer operates at 4800 baud with even parity and is connected to the COM1 port (the first serial connection on your computer), you would type the following two commands:

```
mode com1 48,e,,b  
mode lpt1=com1
```

If you redirect parallel-printer output from LPT1 to COM1 but then decide that you want to print a file by using LPT1, use the following command before you print the file. This command prevents Windows NT from redirecting the file from LPT1 to COM1.

```
mode lpt1
```

More Information About Mode (Redirect Printing)



[_Mkdir \(Redirect Printing\)](#)



Mode (Set Device Code Pages)

Selects, refreshes, or displays the numbers of the code pages the console.

mode *device* **codepage select**=*yyy*

mode *device* **codepage** [**/status**]

Parameters

device

Specifies the device for which you want select a code page. CON is the only valid name for a device.

yyy

Specifies the number of the code page to select. The following list shows each code page that Windows NT supports and its country or language:

437	United States
850	Multilingual (Latin I)
852	Slavic (Latin II)
855	Cyrillic (Russian)
857	Turkish
860	Portuguese
861	Icelandic
863	Canadian-French
865	Nordic
866	Russian
869	Modern Greek

codepage select

Specifies (selects) which code page to use with the specified device. You can abbreviate **codepage** and **select** as **cp** and **sel**, respectively.

codepage

Displays the numbers of the code pages, if any, that are selected for the specified device.

/status

Displays the numbers of the current code pages selected for the specified device. You can abbreviate this switch as **/sta**. Whether or not you specify the **/status** switch, typing the **mode** command with a device name and the **codepage** parameter displays the numbers of the code pages that are selected for the specified device.



Mode (Set Display Mode)

Changes the size of the command prompt screen buffer (characters wide and lines deep).

mode con[:] [cols=c] [lines=n]

Parameters

con[:]

Indicates change is to the command prompt window.

cols=c

Specifies the number of characters (columns) wide in the command prompt screen buffer.

lines=n

Specifies the number of lines deep in the command prompt screen buffer



Mode (Set Typematic Rate)

Sets the keyboard typematic rate, the rate at which Windows NT repeats a character when you hold down the key for that character.

The typematic rate has two components, the rate and the delay. Some keyboards do not recognize this command.

mode con[:] [**rate=r** **delay=d**]

Parameters

con[:]

Refers to the keyboard.

rate=r

Specifies the rate at which a character is repeated on the screen when you hold down a key. Valid values are in the range 1 through 32. These values are equal to approximately 2 to 30 characters per second, respectively. The default value is 20 for IBM AT-compatible keyboards, and 21 for IBM PS/2-compatible keyboards. If you set the rate, you must also set the delay.

delay=d

Specifies the amount of time that must elapse—after you press and hold down a key—before Windows NT starts to repeat the character. Valid values for *d* are 1, 2, 3, and 4 (representing 0.25, 0.50, 0.75, and 1 second, respectively). The default value is 2. If you set the delay, you must also set the rate.



More

Displays one screen of output at a time. This command is commonly used to view long files. Enabling extended features activates commands that control display.

command name | **more** [/e] [/c] [/p] [/s] [/tn] [+n]

more [/e] [/c] [/p] [/s] [/tn] [+n] < [drive:] [path] filename

more [/e] [/c] [/p] [/s] [/tn] [+n] files

Parameters

[drive:] [path] filename

Specifies the file to display.

command name

Specifies a command whose output will be displayed.

/e

Enable extended features.

/c

Clears screen before displaying page.

/p

Expand form-feed characters

/s

Change multiple blank lines to one blank line.

/tn

Changes tabs to *n* spaces.

+n

Displays first file beginning at the line specified by *n*.

files

Specifies list of files to display. Separate filenames with a space.

If extended features are enabled, the following commands are accepted at the --More-- prompt.

Key	Action
space	Display next page.
Enter	Display next line.
F	Display next file
q	Quit.
?	Show available commands.
=	Show line number.
P n	Display next <i>n</i> lines.
S n	Skip next <i>n</i> lines.

See Also



[Filter commands](#)



[Redirection symbols](#)



[Dir](#)

Type

More Information About More

More--Notes

More--Examples



More--Notes

Sources of data

When using the redirection character (<), you must specify a filename as the source. When using the pipe (|), you can use such commands as **dir**, **sort**, and **type**.

More Information About More



[More--Examples](#)



[More](#)



More--Examples

To view the file named CLIENTS.NEW that you want to view on your screen, type either of the following two commands

more < clients.new

type clients.new | more

The **more** command displays the first screen of information from CLIENTS.NEW and then prompts you with the following message:

-- More --

You can then press any key to see the next screen of information.

To clear the screen and remove all extra blank lines before displaying the file CLIENTS.NEW, type either of the following two commands

more /c /s < clients.new

type clients.new | more /c /s

To use extended attributes while viewing EMPLOYEES.NORTH, type

more /e < employees.north

The **more** command displays the first screen of information from EMPLOYEES.NORTH and then prompts you with the following message:

-- More --

You can now use the extended attribute commands to control file display.

To display the file one line at a time press **ENTER**.

To display the next page press the **SPACEBAR**.

To display the next file listed on the command line press **f**

To quit the **more** program, press **q**.

To display the available extended attribute commands, press **?**. The available commands are added to the **more** prompt as in this example

-- More [Options: psfq=<space><ret>] --

To display the current line number press **=**. The current line number is added to the **more** prompt as in this example

-- More [Line: 24] --

To display a specific number of lines press **p**. **More** prompts for the number of lines to display

-- More -- Lines:

Type the number of lines to display and press **ENTER**. **More** displays the specified number of lines.

To skip a specific number of lines press **s**. **More** prompts for the number of lines to skip

-- More -- Lines:

Type the number of lines to skip and press **ENTER**. **More** skips the specified number of lines and displays the next screen of information.

More Information About More



[More--Notes](#)



[More](#)



Move

Moves one or more files from one directory to the specified directory.

move [*source*] [*target*]

Parameters

source

Specifies the path and name of the file or files to move.

target

Specifies the path and name to move files to.

More Information About Move



[Move--Examples](#)



Move--Example

To move all files with the .XLS extension from the \DATA directory to the \SECOND_Q\REPORTS directory and change the extension to .2ND, type

```
move \data\*.xls \second_q\reports\*.2nd
```

More Information About Move



Move

**Nlfunc**

Windows NT and the MS-DOS subsystem do not use this command. It is accepted only for compatibility with MS-DOS files.



Ntbooks

Accesses online Windows NT manuals.

ntbooks [/s] [/w] [/n:path]

Parameters

/s

Used from a Windows NT workstation to access Windows NT server documentation.

/w

Used from a Windows NT server to access Windows NT workstation documentation.

/n

Specifies the path to the online books. Normally, the last used path is remembered and used.

The **ntbooks** command searches for Windows NT manuals at the last used location, but does not display the path to the location. If the location was a CD-ROM drive and the CD with the books is not in the drive, you are prompted to insert the CD.



Ntcmdprompt

Runs the Windows NT command interpreter, CMD.EXE, rather than COMMAND.COM after running a TSR or after starting the command prompt from within an MS-DOS application.

ntcmdprompt

More Information About Ntcmdprompt



[Ntcmdprompt--Notes](#)



[Ntcmdprompt--Examples](#)



Ntcmdprompt--Notes

When COMMAND.COM is running, some features of the Windows NT command prompt, such as the Doskey display of command history, are not available. If you would prefer to run the Windows NT command interpreter after you have started a TSR or started the command prompt from within an MS-DOS-based application, you can use the **ntcmdprompt** command. However, keep in mind that the TSR may not be available for use when you are running CMD.EXE. You can include the **ntcmdprompt** command in your CONFIG.NT file or the equivalent custom startup file in an application's PIF.

More Information About Ntcmdprompt



[Ntcmdprompt--Examples](#)



[Ntcmdprompt](#)




Ntcmdprompt--Examples

Include ntcmdprompt in your CONFIG.NT file, or the Config startup file specified in the PIF.

ntcmdprompt

More Information About Ntcmdprompt

 [Ntcmdprompt--Notes](#)

 [Ntcmdprompt](#)



Path

Sets a search path for executable files.

Windows NT uses the **path** command to search for executable files in the directories you specify. By default, the search path is the current directory only.

path [[*drive:*]path[;...]] [%**path**%]

Parameters

none

Used without parameters, **path** displays the current search path.

[*drive:*]path

Specifies a drive, directory, and any subdirectories to search.

;

When used as the only parameter, clears all search-path settings and specifies that Windows NT is to search only the current directory.

%path%

Appends current path to the new setting.

More Information About Path



[Path--Notes](#)



[Path--Examples](#)



Path--Notes

Current directory

Windows NT always searches in the current directory first, before it searches directories in the search path.

Files with the same name, different extensions

You might have some files in the same directory that share the same filename but have different extensions. For example, you might have a file named ACCNT.COM that starts an accounting program and another file named ACCNT.BAT that connects your system to the accounting system network.

Windows NT searches for a file by using default filename extensions in the following order of precedence: .EXE, .COM, .BAT, and .CMD. To run ACCNT.BAT when ACCNT.COM exists in the same directory, you must include the .BAT extension on the command line.

Two or more identical filenames in the path

You might have two or more files in the search path that have the same filename and extension. Windows NT searches for the specified filename first in the current directory. Then it searches directories in the order in which they are listed in the **path** command.

Specifying multiple paths

To specify more than one path for Windows NT to search, separate entries with a semicolon (;).

Using path in your AUTOEXEC.NT file

If you place the **path** command in your AUTOEXEC.NT file, Windows NT automatically appends the specified MS-DOS subsystem search path to the Windows NT search path every time you log on to your computer.

More Information About Path



[Path--Examples](#)



[Path](#)



Path--Example

The following command specifies that Windows NT is to search three directories to find external commands (the three paths for these directories are C:\USER\TAXES, B:\USER\INVEST, and B:\BIN):

```
path c:\user\taxes;b:\user\invest;b:\bin
```

More Information About Path

 [Path--Notes](#)

 [Path](#)



Pause

Suspends processing of a batch program and displays a message prompting the user to press any key to continue.

pause

More Information About Pause



[Pause--Notes](#)



[Pause--Examples](#)



Pause--Notes

Prompting the user to continue the program

Windows NT displays the following message in response to the **pause** command:

Press any key to continue . . .

Dividing a batch file into sections

If you press CTRL+C to stop a batch program, Windows NT displays the following message:

Terminate batch job (Y/N)?

If you press Y (for yes) in response to this message, the batch program ends and control returns to the operating system. Therefore, you can insert the **pause** command before a section of the batch file you may not want to process. While **pause** suspends processing of the batch program, you can press CTRL+C and then Y to stop the batch program.

More Information About Pause



[Pause--Examples](#)



[Pause](#)



Pause--Example

Suppose you want a batch program to prompt the user to change disks in one of the drives. To do this, you might create the following file:

```
@echo off  
:begin  
copy a:*.*  
echo Please put a new disk into drive A  
pause  
goto begin
```

In this example, all the files on the disk in drive A are copied to the current directory. After the displayed comment prompts you to place another disk in drive A, the **pause** command suspends processing so that you can change disks and then press any key to resume processing. This particular batch program runs in an endless loop. The **goto BEGIN** command sends the command interpreter to the begin label of the batch file. To stop this batch program, press CTRL+C and then Y.

More Information About Pause



[Pause--Notes](#)



[Pause](#)



Pax

Starts the Portable Archive Interchange (Pax) utility.

pax [-**cimopuvy**] [-**f** *archive*] [-**s** *replstr*] [-**t** *device*] [*pattern*...]

pax -r [-**cimnopuvy**] [-**f** *archive*] [-**s** *replstr*] [-**t** *device*] [*pattern*...]

pax -w [-**adimuvy**] [-**b** *blocking*] [-**f** *archive*] [-**s** *replstr*] [-**t** *device*] [-**x** *format*] [*pathname*...]

pax -rw [-**ilmopuvy**] [-**s** *replstr*] [*pathname*...] *directory*

Pax is a POSIX program and pathnames used as arguments must be specified in POSIX format. Use **"//C/USERS/DEFAULT"** instead of **"C:\USERS\DEFAULT."**

Combinations of the **-r** and **-w** command line arguments specify whether **pax** will read, write or list the contents of the specified archive, or move the specified files to another directory.

Parameters

-r

Reads an archive file from the standard input. Only files with names that match any of the *pattern* operands are selected for extraction. The selected files are conditionally created and copied relative to the current directory tree, subject to the options described below. By default, the owner and group of selected files will be that of the invoking process, and the permissions and modification times will be the same as those in the archive. The supported archive formats are automatically detected on input. The default output format is *ustar*, but may be overridden by the **-x** *format* option described below.

-w

Writes the files and directories specified by the *pathname* operands to the standard output together with the pathname and status information prescribed by the archive format used. A directory *pathname* operand refers to the files and (recursively) subdirectories of that directory. If no *pathname* operands are given, then the standard input is read to get a list of pathnames to copy, one pathname per line. In this case, only those pathnames appearing on the standard input are copied.

-rw

Reads the files and directories named in the *pathname* operands and copies them to the destination directory. A directory *pathname* operand refers to the files and (recursively) subdirectories of that directory. If no *pathname* operands are given, the standard input is read to get a list of pathnames to copy, one pathname per line. In this case, only those pathnames appearing on the standard input are copied. The directory named by the *directory* operand must exist and have the proper permissions before the copy can occur.

-a

The files specified by *pathname* are appended to the specified archive.

-b *blocking*

Block the output at blocking bytes per write to the archive file. A *k* suffix multiplies blocking by 1024, a *b* suffix multiplies blocking by 512 and a *m* suffix multiplies blocking by 1048576 (1 mega-byte). If not specified, blocking is automatically determined on input and is ignored for **-rw**.

-c

Complement the match sense of the *pattern* operands.

-d

Intermediate directories not explicitly listed in the archive are not created. This option is ignored unless the **-r** option is specified.

-f *archive*

The archive option specifies the pathname of the input or output archive, overriding the default of standard input for **-r** or standard output for **-w**.

-i

Interactively rename files. Substitutions specified by **-s** options (described below) are performed before requesting the new file name from the user. A file is skipped if an empty line is entered and **pax** exits with an exit status of 0 if EOF is encountered.

-l

Files are linked rather than copied when possible.

-m

File modification times are not retained.

-n

When **-r** is specified, but **-w** is not, the *pattern* arguments are treated as ordinary file names. Only the first occurrence of each of these files in the input archive is read. **Pax** exits with a zero exit status after all files in the list have been read. If one or more files in the list is not found, **pax** writes a diagnostic to standard error for each of the files and exits with a non-zero exit status. The file names are compared before any of the **-i**, **-s**, or **-y** options are applied.

-o

Restore file ownership as specified in the archive. The invoking process must have appropriate privileges to accomplish this.

-p

Preserve the access time of the input files after they have been copied.

-s replstr

File names are modified according to the substitution expression using the syntax of `ed(1)` as shown:

`-s /old/new/[gp]`

Any non null character may be used as a delimiter (a / is used here as an example). Multiple **-s** expressions may be specified; the expressions are applied in the order specified terminating with the first successful substitution. The optional trailing `p` causes successful mappings to be listed on standard error. The optional trailing `g` causes the old expression to be replaced each time it occurs in the source string. Files that substitute to an empty string are ignored both on input and output.

-t device

The device option argument is an implementation defined identifier that names the input or output archive device, overriding the default of standard input for **-r** and standard output for **-w**.

-u

Copy each file only if it is newer than a pre-existing file with the same name. This implies **-a**.

-v

List file names as they are encountered. Produces a verbose table of contents listing on the standard output when both **-r** and **-w** are omitted, otherwise the file names are printed to standard error as they are encountered in the archive.

-x format

Specifies the output archive format. The input format, which must be one of the following, is automatically determined when the **-r** option is used. The supported formats are:

cpio The extended CPIO interchange format specified in Extended CPIO Format in IEEE Std. 1003.1-1988.

Usta The extended TAR interchange format specified in Extended TAR Format in IEEE
r Std. 1003.1-1988. This is the default archive format.

-y

Prompt for the disposition of each file. Substitutions specified by **-s** options (described above) are performed before prompting the user for disposition. EOF or an input line starting with the character `q` caused **pax** to exit.

Otherwise, an input line starting with anything other than `y` causes the file to be ignored. This option cannot be used in conjunction with the `-i` option.

Only the last of multiple `-f` or `-t` options take effect.

directory

The destination directory pathname for copies when both the `-r` and `-w` options are specified. The directory must exist and be writable before the copy or an error results.


pathname


A file whose contents are used instead of the files named on the standard input. When a directory is named, all of its files and subdirectories are copied as well.

pattern

A pattern is given using wildcards. The default is all files.

More Information About Pax

 [Pax--Notes](#)

 [Pax--Example](#)



Pax--Notes

Like other POSIX programs, **pax** is unable to access tape drives. It can only be used to read and write archive files on disk drives. **Pax** reads and writes archive files which conform to the Archive/Interchange File Format specified in IEEE Std. 1003.1-1988. Pax can also read, but not write, a number of other file formats in addition to those specified in the Archive/Interchange File Format description. Support for these traditional file formats, such as V7 tar and System V binary **cpio** format archives, is provided for backward compatibility and to maximize portability.

If neither the **-r** or **-w** options are given, then **pax** will list the contents of the specified archive. In this mode, **pax** lists normal files one per line, hard link pathnames as

```
pathname == linkname
```

and symbolic link pathnames (if supported by the implementation) as

```
pathname -> linkname
```

where pathname is the name of the file being extracted, and linkname is the name of a file which appeared earlier in the archive.

If the **-v** option is specified, then **pax** list normal path names in the same format used by the **ls** utility with the **-l** option. Hard links are shown as

```
<ls -l listing> == linkname
```

and symbolic links (if supported) are shown as

```
<ls -l listing> -> linkname
```

Upon detecting an end of medium on an archive which is not yet completed, **pax** will prompt the user for the next volume of the archive and will allow the user to specify the location of the next volume.

When writing to an archive, the standard input is used as a list of pathnames if no pathname operands are specified. The format is one pathname per line. Otherwise, the standard input is the archive file, which is formatted according to one of the specifications in Archive/Interchange File format in IEEE Std. 1003.1-1988, or some other implementation-defined format.

The user ID and group ID of the process, together with the appropriate privileges, affect the ability of **pax** to restore ownership and permissions attributes of the archived files. (See format-reading utility in Archive/Interchange File Format in IEEE Std. 1003.1-1988.)

The options **-a**, **-c**, **-d**, **-i**, **-l**, **-p**, **-t**, **-u**, and **-y** are provided for functional compatibility with the historical **cpio** and **tar** utilities. The option defaults were chosen based on the most common usage of these options, therefore, some of the options have meanings different than those of the historical commands.

More Information About Pax



[Pax--Example](#)



[Pax](#)



Pax--Example

The commands:

```
mkdir newdir
```

```
cd olddir
```

```
pax -rw . newdir
```

copy the contents of OLDDIR to NEWDIR .

The command:

```
pax -r -s '/*usr/*,' -f pax.out
```

reads the archive PAX.OUT with all files rooted in "/usr" in the archive extracted relative to the current directory.

More Information About Pax



[Pax--Notes](#)



[Pax](#)



Pentnt

Detects floating point division error, when present, in the Pentium chip, disables floating point hardware, and turns on floating point emulation.

pentnt [-c] [-f] [-o] [-?|-h]

Parameters

-c

Enables conditional emulation. Floating-point emulation will be forced on if and only if the system detects the Pentium processor floating-point division error at start time. If you select this parameter, you must restart the computer for the changes to take effect.

-f

Enables forced emulation. Floating-point hardware is disabled and floating-point emulation will always be forced on, regardless of whether the system exhibits the Pentium processor floating-point division error. This parameter is useful for testing software emulators and for working around floating-point hardware defects known to the operating system. If you select this parameter, you must restart the computer for the changes to take effect.

-o

Disables forced emulation and re-enables floating-point hardware if it is present. If you select this parameter, you must restart the computer for the changes to take effect.

-?|-h

Displays Help for the command.



Popd

Changes to the directory stored by the **pushd** command.

Popd can be used only once to change directories; the buffer is cleared after the first use.

popd

More Information About Popd



[Popd--Example](#)



Popd--Example

You can use **pushd** and **popd** in a batch program to return to the directory where the batch program was started:

```
@echo off  
rem This batch file deletes all .TXT files in a specified directory  
pushd %1  
del *.txt  
popd  
cls  
echo All text files deleted in the %1 directory
```

More Information About Popd





Portuas

Merges a LAN Manager 2.x user accounts database into an existing Windows NT user accounts database.

portuas -f filename [-u username] [-v] [/codepage codepage] [/log filename]

Parameters

-f filename

Specifies the LAN Manager 2.x NET.ACC file.

-u username

Specifies a single user or group to restore.

-v

Displays all messages (verbose).

/codepage codepage

Specifies the OEM codepage the LAN Manager 2.x NET.ACC file is in.

/log filename

Specifies a file to log results.

More Information About Portuas



[Portuas--Notes](#)



[Portuas--Examples](#)



Portuas--Notes

Portuas allows you to restore the user accounts and groups of a LAN Manager 2.x server onto a Windows NT computer.

You must be a member of an administrator group to use **portuas**.

The Server and Net Logon services must be running on the Windows NT computer.

For complete information about restoring user account databases from LAN Manager OS/2 servers, see your Windows NT documentation.

More Information About Portuas



[Portuas--Examples](#)



[Portuas](#)



Portuas--Example

To merge a LAN Manager 2.x user account database into the Windows NT user accounts database, type

```
portuas -f \\lanman\accounts\net.acc -v
```

To add the user account CRISTALW from a LAN Manager 2.x user account database to a Windows NT computer, type

```
portuas -f a:\net.acc -u cristalw
```

More Information About Portuas



[Portuas--Notes](#)



[Portuas](#)



Print

Prints a text file while you are using other Windows NT commands.

This command can print in the background if you have an output device connected to one of your system's serial or parallel ports.

print [/d:device] [drive:][path] filename[...]

Parameters

none

Type **print** without parameters to display the contents of the print queue.

/d:device

Specifies the name of the print device. Valid values for parallel ports are LPT1, LPT2, and LPT3. Valid values for serial ports are COM1, COM2, COM3, and COM4. You can also specify a network printer by its sharename (\\servername\print_share). The default value is PRN. The values PRN and LPT1 refer to the same parallel port.

[drive:][path] filename

Specifies the location and name of a file or set of files you want to print. You can include multiple files on one command line.

See Also

For information about configuring a printer connected to a parallel port, see the [MODE \(configure printer\)](#) command.

For information about displaying the status of a printer, see the [MODE \(display device status\)](#) command.

For information about configuring a printer connected to a serial port, see the [MODE \(redirect printing\)](#) command.

For information about preparing printers for code-page switching, see the [MODE \(set device code pages\)](#) command.

More Information About Print



[Print--Notes](#)



[Print--Examples](#)



Print--Notes

Printing files generated by programs

Many programs have their own print commands. You should use a program's print command to print files that you create with the program.

More Information About Print



[Print--Examples](#)



[Print](#)



Print--Examples

The following command prints the file REPORT.TXT on LPT1:

```
print /d:LPT1 report.txt
```

More Information About Print

 [Print--Notes](#)

 [Print](#)



Prompt

Changes the Windows NT command prompt.

You can customize the command prompt to display any text you want, including such information as the name of the current directory, the time and date, and the Windows NT version number.

prompt [*text*]

Parameters

text

Specifies any text and information you want included in your system prompt. The following list shows the character combinations you can include instead of, or in addition to, any character string(s) in the text parameter. The list includes a brief description of the text or information that each character combination adds to your command prompt.

Characters	Displays
\$q	= (equal sign)
\$\$	\$ (dollar sign)
\$t	Current time
\$d	Current date
\$p	Current drive and path
\$v	Windows NT version number
\$n	Current drive
\$g	> (greater-than sign)
\$l	< (less-than sign)
\$b	(pipe)
\$_	ENTER-LINEFEED
\$e	ANSI escape code (code 27)
\$h	Backspace (to delete a character that has been written to the prompt command line)
\$a	& (ampersand)
\$c	((left parenthesis)
\$f) (right parenthesis)
\$s	space

More Information About Prompt



[Prompt--Notes](#)



[Prompt--Examples](#)



Prompt--Notes

Using the prompt command without the text parameter

When you use the **prompt** command without specifying a value for text, **prompt** resets the command prompt to the default setting, the current drive letter followed by the current directory and a greater-than sign (>).

Using the \$p value for text

If you include the **\$p** character in the text parameter, Windows NT reads your disk after you enter each command to determine the current drive and path. This can take extra time, especially for floppy disk drives.

More Information About Prompt



[Prompt--Examples](#)



[Prompt](#)



Prompt--Examples

The following example sets the command prompt to display the current drive and path followed by the greater-than sign (>):

```
prompt $p$g
```

The following command displays a two-line prompt in which the current time appears on the first line and the current date appears on the second line:

```
prompt time is: %t%_date is: %d
```

More Information About Prompt



[Prompt--Notes](#)



[Prompt](#)

**Protshell**

Windows NT and the OS/2 subsystem do not use this command. It is accepted only for compatibility with files from MS OS/2 version 1.3 or earlier.



Pushd

Stores the current directory for use by the **popd** command, then changes to the specified directory.

pushd [*path* | ..]

Parameter

path

Specifies the directory to make the current directory.

More Information About Pushd



[Pushd--Example](#)



Pushd--Example

You can use **pushd** and **popd** in a batch program to return to the directory where the batch program was started:

```
@echo off  
rem This batch file deletes all .TXT files in a specified directory  
pushd %1  
del *.txt  
popd  
cls  
echo All text files deleted in the %1 directory
```

More Information About Pushd



[Pushd](#)



QBasic

Starts Windows NT QBasic, a program that reads instructions written in the Basic computer language and interprets them into executable computer code.

qbasic [/b] [/editor] [/g] [/h] [/mbf] [/nohi] [[/run][drive:][path] filename]

Parameters

[drive:][path] filename

Specifies the location and name of the file to load when QBasic starts.

/b

Displays QBasic in black and white if you have a color monitor.

/editor

Invokes MS-DOS Editor, a full-screen text editor.

/g

Provides the fastest update of a CGA monitor.

/h

Displays the maximum number of display lines possible on your screen.

/mbf

Converts the built-in functions MKS\$, MKD\$, CVS, and CVD to MKSMBF\$, MKDMBF\$, CVSMBF, and CVDMBF, respectively.


/nohi

Allows the use of a monitor that does not support high-intensity video. Do not use this switch with COMPAQ laptop computers.

/run

Runs the specified Basic program before displaying it. You must specify a filename.

More Information About Qbasic

 [qbasic--Notes](#)



QBasic--Notes

Cannot use MS-DOS Editor if Windows NT QBasic is not present

To use MS-DOS Editor, you must have the QBASIC.EXE file in the current directory or in your search path or in the same directory as the EDIT.COM file. If you delete QBASIC.EXE to save space on your hard disk, you cannot use MS-DOS Editor.

Running consecutive Basic programs

You can run consecutive Basic programs from a batch file by using the Basic system statement and the **qbasic** command with the **/run** switch. A **system** statement returns control to Windows NT after a Basic program has run, instead of returning to QBasic. This allows you to run more than one Basic program from a batch file without having to intervene.


Converting a GW-BASIC program to run in QBasic

For a Basic program that helps convert GW-BASIC programs to QBasic, see the file REMLINE.BAS (included with QBasic).

Display of shortcut keys

Some monitors may not support the display of shortcut keys by default. If your monitor does not display shortcut keys, use the **/b** switch (for CGA monitors) and the **/nohi** switch (for systems that do not support bold characters).

More Information About QBasic

 [_qbasic](#)



Recover

Recovers readable information from a bad or defective disk.

The **recover** command reads a file sector by sector and recovers data from the good sectors. Data in bad sectors is lost.

recover [*drive:*][*path*] *filename*


Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of the file you want to recover.

More Information About Recover

 [Recover--Notes](#)

 [Recover--Examples](#)



Recover--Notes

Limitation on [drive:][path]filename

You cannot use wildcards (* and ?) with the **recover** command. You must specify a file.

Reentering lost data

Since all data in bad sectors is lost when you recover a file, you should recover files one at a time. This method allows you to edit each file and reenter missing information after you recover the file.

Recovering bad sectors

Bad sectors reported by **chkdsk** were marked as "bad" when your disk was first prepared for operation. They pose no danger, and **recover** will not affect them.

Recover vs. backup and restore

The **recover** command does not work with the MS-DOS **backup** or **restore** command. You must use **restore** with backup files you created by using the MS-DOS **backup** command.

More Information About Recover



[Recover--Examples](#)



[Recover](#)



Recover--Examples

To recover the file STORY.TXT in the \FICTION directory in drive d:, type

```
recover d:\fiction\story.txt
```

More Information About Recover

 [Recover--Notes](#)

 [Recover](#)



Rem

Enables you to include comments (remarks) in a batch file or in your configuration files.

rem [*comment*]

Parameters

comment

Specifies any string of characters you want to include as a comment.

See Also



[Echo](#)

More Information About Rem



[Rem--Notes](#)



[Rem--Examples](#)



Rem--Notes

Using the echo command to display comments

The **rem** command does not display comments on the screen. You must use the **echo on** command in your batch or CONFIG.NT file in order to display comments on the screen.

Restrictions on batch-file comments

You cannot use a redirection character (or) or pipe (|) in a batch-file comment.

Using rem to add vertical spacing

Although you can use **rem** without a comment to add vertical spacing to a batch file, you can also use blank lines. Windows NT ignores the blank lines when processing the batch program.

More Information About Rem



[Rem--Examples](#)



[Rem](#)



Rem--Examples

The following example shows a batch file that uses remarks for both explanations and vertical spacing:

```
@echo off  
rem This batch program formats and checks new disks.  
rem It is named CHECKNEW.BAT.  
rem  
echo Insert new disk in drive B.  
pause  
format b: /v  
chkdsk b:
```

Suppose you want to include in your CONFIG.NT file an explanatory comment before the **prompt** command. To do this, add the following lines to CONFIG.NT:

```
rem Set prompt to indicate current directory  
prompt $p$g
```

More Information About Rem



[_Rem--Notes](#)



[_Rem](#)



Rename (Ren)

Changes the name of a file or files.

You can rename all files matching the specified filename. You cannot use the **rename** command to rename files across drives or to move files to a different directory location.

rename [*drive:*][*path*] *filename1 filename2*

ren [*drive:*][*path*] *filename1 filename2*

Parameters

[*drive:*][*path*] *filename1*

Specifies the location and name of the file or set of files you want to rename.


filename2

Specifies the new name for the file or, if you use wildcards, the new names for the files. (You cannot specify a new drive or path.)

See Also


 [Move](#)

 [Copy](#)

 [Xcopy](#)

More Information About Rename

 [Rename--Notes](#)

 [Rename--Examples](#)



Rename--Notes

Using wildcards with rename

You can use wildcards (* and ?) in either filename parameter. If you use wildcards in *filename2*, the characters represented by the wildcards will be identical to the corresponding characters in *filename1*.

Rename will not work if filename2 already exists

If, for *filename2*, you specify a filename that already exists, **rename** displays the following message:

Duplicate file name or file not found

More Information About Rename



[Rename--Examples](#)



[Rename](#)



Rename--Examples

Suppose you want to change the extensions of all the filenames in the current directory that have the extension .TXT; for example, suppose you want to change the .TXT extensions to .DOC extensions. To make this change, type the following command:

```
ren *.txt *.doc
```

To rename a file or directory named CHAP10 to PART10, type the following command:

```
REN CHAP10 PART10
```

More Information About Rename



[Rename--Notes](#)



[Rename](#)



Replace

Replaces files in the destination directory with files in the source directory that have the same name. You can also use **replace** to add unique filenames to the destination directory.

replace [*drive1:*][*path1*] *filename* [*drive2:*][*path2*] [**/a**] [**/p**] [**/r**] [**/w**]

replace [*drive1:*][*path1*] *filename* [*drive2:*][*path2*] [**/p**] [**/r**] [**/s**] [**/w**] [**/u**]

Parameters

[*drive1:*][*path1*] *filename*

Specifies the location and name of the source file or set of files.

[*drive2:*][*path2*]

Specifies the location of the destination file. You cannot specify a filename for files you replace. If you specify neither a drive nor a directory, **replace** uses the current drive and directory as the destination.

/a

Adds new files to the destination directory instead of replacing existing files. You cannot use this switch with the **/s** or **/u** switch.

/p

Prompts you for confirmation before replacing a destination file or adding a source file.

/r

Replaces read-only files as well as unprotected files. If you do not specify this switch but attempt to replace a read-only file, an error results and stops the replacement operation.

/s

Searches all subdirectories of the destination directory and replaces matching files. You cannot use the **/s** switch with the **/a** switch. The replace command does not search subdirectories specified in *path1*.

/w

Waits for you to insert a disk before replace begins to search for source files. If you do not specify **/w**, replace begins replacing or adding files immediately after you press ENTER.

/u


Replaces (updates) only those files on the destination directory that are older than those in the source directory. You cannot use the **/u** switch with the **/a** switch.

See Also

 [_Attrib](#)

More Information About Replace

 [_Replace--Notes](#)

 [_Replace--Examples](#)



Replace--Notes

Replace messages

As **replace** adds or replaces files, Windows NT displays their filenames on the screen. After the **replace** command is finished, Windows NT displays a summary line in one of the following formats:

nnn files added
nnn files replaced
no file added
no file replaced

Replacing files on floppy disks

If you are using floppy disks and need to switch disks during the **replace** operation, you can specify the **/w** switch so that **replace** will wait for you to switch disks, as necessary.

Limitations on replace

You cannot use the **replace** command to update hidden files or system files. For information about changing hidden and system attributes, see the [attrib](#) command.

Replace exit codes

The following list shows each exit code and a brief description of its meaning:

- | | |
|----|---|
| 0 | Replace successfully replaced or added the files. |
| 1 | Replace encountered an incorrect version of MS-DOS. |
| 2 | Replace could not find the source files. |
| 3 | Replace could not find the source or destination path. |
| 5 | The user does not have access to the files you want to replace. |
| 8 | There is insufficient system memory to carry out the command. |
| 11 | The user used the wrong syntax on the command line. |

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by **replace**. For an example of a batch program that processes exit codes, see the [if](#) command.

More Information About Replace



[Replace--Examples](#)

 Replace



Replace--Examples

Suppose that several directories on drive C contain different versions of a file named PHONES.CLI, which contains client names and phone numbers. To replace all of these files with the latest version of the PHONES.CLI file from the disk in drive A, type the following command:

replace a:\phones.cli c:\ /s

Suppose you want to add new printer device drivers to a directory on drive C named TOOLS, which already contains several printer device-driver files for a word processor. To do this, type the following command:

replace a:*.prd c:\tools /a

This command searches the current directory on drive A for any files that have the extension .PRD and then adds these files to the TOOLS directory on drive C. Because the **/a** switch is included, **replace** adds only those files from drive A that do not exist on drive C.

More Information About Replace



[Replace--Notes](#)



[Replace](#)



Restore

Restores files that were backed up by using the MS-DOS **backup** command.

restore *drive1:* *drive2:[path[filename]]* [/s] [/p] [/b:date] [/a:date] [/e:time] [/l:time] [/m] [/n] [/d]

Parameters

drive1:

Specifies the drive on which the backed-up files are stored.

drive2:

Specifies the drive to which the backed-up files will be restored.

path

Specifies the directory to which the backed-up files will be restored. You must specify the same directory from which the files were backed up.

filename

Specifies the names of the backed-up files you want to restore.

/s

Restores all subdirectories.

/p

Prompts you for permission to restore files that are read-only (that have the read-only attribute set) or that have changed since the last backup (that have the archive attribute set).

/b:date

Restores only those files last modified on or before the specified *date*. For information about specifying *date*, see the **date** command.

/a:date

Restores only those files last modified on or after the specified date. For information about specifying *date*, see the date command.

/e:time

Restores only those files last modified at or earlier than the specified time. For information about specifying *time*, see the time command.

/l:time

Restores only those files last modified at or later than the specified *time*. For information about specifying *time*, see the time command.

/m

Restores only those files modified since the last backup.

/n

Restores only those files that no longer exist on the destination disk.

/d

Displays a list of the files on the backup disk that match the names specified in *filename* without restoring any files. Even though no files are being restored, you must specify *drive2* when you use **/d**.

More Information About Restore



[Restore--Notes](#)



[Restore--Examples](#)



Restore--Notes

Checking restored files

Once a file has been restored, you can use the **dir** or **type** command to make sure the file was restored properly.

Compatibility with previous versions of backup

The Windows NT **restore** command can restore files that were backed up by using any previous version of the MS-DOS **backup** command.

Restore exit codes

The following list shows each exit code and a brief description of its meaning:

- 0 Restore **successfully restored the file or files.**
- 1 Restore **could not find the files to restore.**
- 3 The user pressed CTRL+C to stop the restoring operation.
- 4 Restore **stopped because of an error.**

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by **restore**.

Listing the names of backed-up files

Use the **/d** switch to see a list of the backed up files. If you specify *filename* with the **/d** switch, **restore** displays a list of the backed up files that match the name you specify. If you use the **/d** switch, **restore** does not restore any files.

More Information About Restore

 [Restore--Examples](#)

 [Restore](#)



Restore--Examples

To restore the file INVEST.MNT from the backup disk in drive A to the IRS directory on drive C, type the following command:

restore A: C:\IRS\INVEST.MNT

Windows NT prompts you to insert the backup disk into drive A. Once the backup disk is in drive A, press **ENTER** to continue.

Suppose you backed up all of the files in the directory \USER\ADAMS on drive C. To restore these files, insert the backup disk in drive A and type the following command:

restore A: C:\USER\ADAMS*.*

It is important that you specify *.* for filename. Otherwise, the **restore** command attempts to restore a file named ADAMS in the USER directory.

To restore a complete hard disk from a backup disk (or disks) in drive A, type the following command:

restore A: C:*.* /s

The **/s** switch and the wildcards (*.*) specify that **restore** is to restore all backed-up files to their original directories and subdirectories on drive C.

More Information About Restore



[Restore--Notes](#)



[Restore](#)



Rmdir (Rd)

Deletes (removes) a directory.

rmdir [*drive:*]*path* [**/s**]

rd [*drive:*]*path* [**/s**]

Parameters


[*drive:*]*path*

Specifies the location and name of the directory you want to delete.

/s

Removes the specified directory and all subdirectories including any files. Used to remove a directory tree.


See Also


 [Mkdir](#)

 [Dir](#)

 [Attrib](#)

More Information About Rmdir

 [Rmdir--Notes](#)

 [Rmdir--Examples](#)



Rmdir--Notes

Cannot delete directory with hidden or system files

You cannot delete a directory that contains files, including hidden or system files. If you attempt to do so, Windows NT displays the following message:

The directory not empty

Use the **dir** command to list hidden and system files and the **attrib** command to remove hidden and system attributes from files. For more information, see those commands.

Using the backslash character with the path parameter

If you insert a backslash (\) before the first directory name in path, Windows NT treats the directory as a subdirectory of the root directory--regardless of your current directory. If you do not insert a backslash before the first directory name in path, Windows NT treats the directory as a subdirectory of the current directory.

Deleting the current directory

You cannot use **rmdir** to delete the current directory. You must first change to a different directory (not a subdirectory of the current directory) and then use **rmdir** with a path. If you attempt to delete the current directory, Windows NT displays a message in the following format:

The process cannot access the file because it is being used by another process.

More Information About Rmdir



[Rmdir-Examples](#)



[Rmdir](#)



Rmdir--Example

To delete a directory named \USER\SMITH, first ensure that the directory is empty, as in following example:

dir \user\smith /a

Windows NT should display only the "." and ".." symbols.

Then, from any directory except \USER\SMITH, type the following command:

rmdir \user\smith

To delete the directory \USER and all files and subdirectories therein, type

rmdir /s \user

More Information About Rmdir



[Rmdir--Notes](#)



[Rmdir](#)



Set

Displays, sets, or removes Windows NT environment variables.

You use environment variables to control the behavior of some batch files and programs and to control the way Windows NT and the MS-DOS subsystem appears and works. The **set** command is often used in the AUTOEXEC.NT file to set environment variables.

set [*variable*=[*string*]]

Parameters

none

Used without parameters, **set** displays the current environment settings.

variable

Specifies the variable you want to set or modify.

string

Specifies the string you want to associate with the specified variable.

More Information About Set



[Set--Notes](#)



[Set--Examples](#)



Set--Notes

Displaying the current environment settings

When you type the **set** command alone, Windows NT displays the current environment settings. These settings usually include the **comspec** and **path** environment variables that Windows NT uses to help find programs on disk. **Prompt** and **dircmd** are two other environment variables that Windows NT uses.

Using parameters

When you use a set command and specify values for both variable and string, Windows NT adds the specified variable value to the environment and associates the string with that variable. If variable already exists in the environment, the new string value replaces the old string value.

If you specify only a variable and an equal sign (without a string) for the **set** command, Windows NT clears the string value associated with the variable (as if the variable is not there at all).

Using set in batch files

When creating batch files, you can use the **set** command to create variables and use them in the same way as you would the numbered variables **%0** through **%9**. You can also use the variables **%0** through **%9** as input for the **set** command.

Calling a set variable from a batch file

When you call a variable value from a batch file, you must enclose the value with percent signs (%). For example, if your batch program creates an environment variable named **baud**, you can use the string associated with **baud** as a replaceable parameter by inserting **%baud%** on the command line.

More Information About Set



[Set--Examples](#)



[Set](#)



Set--Examples

To set an environment variable named `include` so that the string `C:\INC` (the `INC` directory on drive `C`) is associated with it, type the following command:

```
set include=c:\inc
```

You can then use the string `C:\INC` in batch files by enclosing the name **include** with percent signs (%). For example, you might include the following command in a batch file in order to display the contents of the directory associated with the **include** environment variable:

```
dir %include%
```

When Windows NT processes this command, the string `C:\INC` replaces **%include%**.

Another possible use for the **set** command is in a batch program that adds a new directory to the **path** environment variable, as the following example shows:

```
@echo off  
rem ADDPATH.BAT adds a new directory  
rem to the path environment variable.  
set path=%1;%path%  
set
```

More Information About Set



[Set--Notes](#)



[Set](#)




Setlocal

Begins localization of environment variables in a batch file. Each **setlocal** command must have an **endlocal** command to restore environment variables.


setlocal

See Also

 [Endlocal](#)

More Information About Setlocal

 [Setlocal--Notes](#)

 [Setlocal--Examples](#)



Setlocal--Notes

Setlocal permits you to change environment variables during the execution of a batch file; the environment variables are restored to their original values after an **endlocal** command is issued. Environment variables are not restored to their original values when a batch file ends without the **endlocal** command.

You can have more than one **setlocal/endlocal** command in a batch program (recursion).

More Information About Setlocal



[Setlocal](#)



[Setlocal--Examples](#)



Setlocal--Example

You can localize environment variables in a batch file.

```
@echo off  
rem This program starts the superapp batch program on the network,  
rem directs the output to a file, and displays the file  
rem in Notepad.  
setlocal  
path=g:\programs\superapp;%path%  
call superapp>c:\superapp.out  
endlocal  
start notepad c:\superapp.out
```

More Information About Setlocal



[Setlocal--Notes](#)



[Setlocal](#)



Setver

Sets the MS-DOS version number that the MS-DOS subsystem reports to a program.

setver [*drive:path*] [*filename n.nn*]

setver [*drive:path*] [*filename*] [**/delete**] [**/quiet**]

To display the current version table, use the following syntax:

setver [*drive:path*]

Parameters

[*drive:path*]

Specifies the location of the **setver**.EXE file.

filename

Specifies the name of the program file (.EXE or .COM) that you want to add to the version table. You cannot use a wildcard (* or ?).

n.nn

Specifies the MS-DOS version (for example, 3.3 or 4.01) that the MS-DOS subsystem reports to the specified program file.

/delete

Deletes the version-table entry for the specified program file. You can abbreviate this switch as **/d**.

/quiet

Hides the message typically displayed during deletion of an entry from the version table.

More Information About Setver



[Setver--Notes](#)



[Setver--Examples](#)



Setver--Notes

Using the version table

Many programs designed to run with a previous version of MS-DOS will run correctly with Windows NT. In some cases, however, a program might not run correctly unless its name is included in the version table. The table indicates to the program that it is running with the MS-DOS version for which it was designed, even though it is running in the MS-DOS subsystem. By interpreting MS-DOS version 5.0 as the earlier version, the program will probably run correctly; however, using **setver** will not solve the problem if the program is not compatible with Windows NT.

Loading the version table into memory

Before you can use the **setver** command, the version table must be loaded into memory by a **device** command in your CONFIG.NT file.

Restarting after updating the version table

When you update the version table by adding or deleting entries, you must start a new command prompt session to reread the version table.

Updating existing entries

If you specify a filename that is already in the version table, the new entry replaces the existing entry.

Setver exit codes


The following list shows each exit code and a brief description of its meaning:

- | | |
|---|--|
| 0 | Setver successfully completed its task. |
| 1 | The user specified an invalid command switch. |
| 2 | The user specified an invalid filename. |
| 3 | There is insufficient system memory to carry out the command. |
| 4 | The user specified an invalid version-number format. |
| 5 | Setver could not find the specified entry in the version table. |
| 6 | Setver could not find the SETVER.EXE file. |

- 7 The user specified an invalid drive.
- 8 The user specified too many command-line parameters.
- 9 Setver **detected missing command-line parameters.**
- 10 Setver **detected an error while reading the SETVER.EXE file.**
- 11 The SETVER.EXE file is corrupt.
- 12 The specified SETVER.EXE file does not support a version table.
- 13 There is insufficient space in the version table for a new entry.
- 14 Setver **detected an error while writing to the SETVER.EXE file.**

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by **setver**. For an example of a batch program that processes exit codes, see the [if](#) command.

More Information About Setver

 [Setver--Examples](#)

 [Setver](#)



Setver--Examples

Suppose you have a program file named MYPROG.EXE that runs with MS-DOS version 3.30. To run MYPROG.EXE, you must first use the setver command to create an entry in the version table that will cause MYPROG.EXE to interpret the MS-DOS subsystem as version 3.30:

```
setver myprog.exe 3.30
```

To delete the MYPROG.EXE entry from the version table (without otherwise affecting the MYPROG.EXE file), type the following command:

```
setver myprog.exe /delete
```

To list the contents of the version table on drive C, type the following command:

```
setver c:
```

More Information About Setver



[Setver--Notes](#)



[Setver](#)

**Share**

Windows NT and the MS-DOS subsystem do not use this command. It is accepted only for compatibility with MS-DOS files.



Shell

Specifies the name and location of an alternate command interpreter you want Windows NT to use for the MS-DOS subsystem.

shell=[[drive:]path] filename [parameters]

Parameters

[[drive:]path] filename

Specifies the location and name of the command interpreter you want Windows NT to use.

parameters

Specifies any command-line parameters or switches that can be used with the specified command interpreter.

More Information About Shell



[Shell--Notes](#)



[Shell--Examples](#)



Shell--Notes

Default setting

By default, the MS-DOS subsystem uses a special version of COMMAND.COM that works seamlessly with the other Windows NT subsystems (including piping and redirection between subsystems); the **shell** command is unnecessary.

Although it is not recommended, you can use the **shell** command to specify your own 16-bit command interpreter.

Using switches with a command interpreter

The **shell** command itself does not accept any switches, but if the command interpreter does, you can include them on the **shell** command line.

More Information About Shell



[Shell--Examples](#)



[Shell](#)



Shell--Example

Suppose the file NEWSHELL.COM is in a directory named BIN on your startup drive, and suppose you want to use NEWSHELL.COM as your command interpreter. To do this, add the following command to your CONFIG.NT file:

shell=\bin\newshell.com

More Information About Shell



Shell--Notes



Shell




Shift

Changes the position of replaceable parameters in a batch file.

shift

More Information About Shift

 [Shift--Notes](#)

 [Shift--Examples](#)



Shift--Notes

How the shift command works

The **shift** command changes the values of the replaceable parameters **%0** through **%9**, by copying each parameter into the previous one. In other words, the value of **%1** is copied to **%0**, the value of **%2** is copied to **%1**, and so on. This is useful for writing a batch file that performs the same operation on any number of parameters.

Working with more than 10 command-line parameters

You can also use the **shift** command to create a batch file that can accept more than 10 parameters. If you specify more than 10 parameters on the command line, those that appear after the tenth (**%9**) will be shifted one at a time into **%9**.

Shifting parameters back

There is no backward SHIFT command. Once you carry out the SHIFT command, you cannot recover the first parameter (**%0**) that existed before the shift.

More Information About Shift



[Shift--Examples](#)



[Shift](#)



Shift--Example

The following batch file, MYCOPY.BAT, shows how to use the **shift** command with any number of parameters. It copies a list of files to a specific directory. The parameters are the directory name followed by any number of filenames.

```
@echo off  
rem MYCOPY.BAT copies any number of files  
rem to a directory.  
rem The command uses the following syntax:  
rem mycopy dir file1 file2 ...  
set todir=%1  
:getfile  
shift  
if "%1"==" " goto end  
copy %1 %todir%  
goto getfile  
:end  
set todir=  
echo All done
```

More Information About Shift



[Shift--Notes](#)



[Shift](#)



Sort

Reads input, sorts data, and writes the results to the screen, to a file, or to another device.

sort [/r] [/+n] [<] [drive1:][path1] filename1 [> [drive2:][path2] filename2]

[command |] **sort** [/r] [/+n] [> [drive2:][path2] filename2]

Parameters

[drive1:][path1] filename1

Specifies the location and name of the file whose data you want to sort.

[drive2:][path2] filename2

Specifies the location and name of a file in which the sorted output is to be stored.

command

Specifies a command whose output is the data you want to sort.

/r

Reverses the order of the sorting operation; that is, sorts from Z to A, and then from 9 to 0.

/+n

Sorts the file according to the character in column *n*. If you do not use this switch, the **sort** command sorts data according to the characters in column 1.

See Also



[filter commands](#)



[redirection symbols](#)

More Information About Sort



[Sort--Notes](#)



[Sort--Examples](#)



Sort--Notes

Specifying a source

Unless you specify the *command* or filename parameter, **sort** acts as a filter and takes input from the Windows NT standard input (usually from the keyboard, from a pipe, or from a file).

Using redirection symbols with sort

You can use the pipe (|) or the less-than sign (<) to direct data through the **sort** *command* from *command* or *filename*. If you want to display the information one screen at a time or direct the information to a file, you can also specify the **more** command or a filename. You can use the greater-than sign (>) to direct the sorted output to a file.


Uppercase vs. lowercase

Sort **does not distinguish between uppercase and lowercase letters.**

Limits on file size

The **sort** command has no limit on file size.

More Information About Sort

 [Sort--Examples](#)

 [Sort](#)

Collating sequence

The sort program uses the collating-sequence table corresponding to the country code and code-page settings. Characters greater than ASCII code 127 are sorted based on information in the COUNTRY.SYS file or in an alternate file specified by the COUNTRY command in your CONFIG.NT file.



Sort--Examples

The following command reads the file EXPENSES.TXT, sorts it in reverse order, and displays it on your screen:

sort /r < expenses.txt

Suppose you want to search a large file named MAILLST.TXT for the text "Jones", and suppose you want to sort the results of the search. To do this, use the pipe (|) to direct the output of a **find** command to the SORT command, as shown in the following example:

find "Jones" maillist.txt | sort

The command produces a sorted list of lines that contain the specified text.

To sort keyboard input and display the results alphabetically on the screen, you can first use the **sort** command with no parameters, as the following example shows:

sort

Then type the text you want sorted, pressing ENTER at the end of each line. When you have finished typing text, press CTRL+Z, and then press ENTER. The **sort** command displays the text you typed, sorted alphabetically. You could also redirect sorted keyboard input to a file.

More Information About Sort



[Sort--Notes](#)



[Sort](#)



Stacks

Supports the dynamic use of data stacks to handle hardware interrupts.

stacks= n,s

Parameters

n

Specifies the number of stacks. Valid values for n are 0 and numbers in the range 8 through 64.

s

Specifies the size (in bytes) of each stack. Valid values for s are 0 and numbers in the range 32 through 512.

More Information About Stacks



[Stacks--Notes](#)



[Stacks--Examples](#)



Stacks--Notes

Default settings

The default settings for the STACK command are as follows:

Computer	Stacks
IBM PC, IBM PC/XT, IBM PC-Portable	0,0
Other	9,128

Special cases for stack allocation

Upon receiving a hardware interrupt, Windows NT allocates one stack from the specified number of stacks. When you specify 0 for the *n* and *s* values, Windows NT allocates no stacks. If the values are 0, each running program must have enough stack space to accommodate the computer's hardware interrupt drivers. Many computers operate correctly, saving some memory for programs, with *n* and *s* values of 0. If, however, your computer becomes unstable when you set these values to 0, return to the default values.

More Information About Stacks



[Stacks--Examples](#)



[Stacks](#)



Stacks--Example

To allocate 8 stacks of 512 bytes each for hardware-interrupt handling, add the following command to your CONFIG.NT file:

stacks=8,512

More Information About Stacks



[Stacks--Notes](#)



[Stacks](#)



Start

Starts a separate window to run a specified program or command.

start ["*title*"] [/d*path*] [/i] [/min] [/max] [/separate] [/low] [/normal] [/high] [/realtime] [/wait] [/b]
[*filename*] [*parameters*]

Parameters

none

Used without parameters, **start** opens a second command prompt window.

"title"

Title to display in window title bar

/dpath

Startup directory

/i

Passes CMD.EXE startup environment to the new window.

/min

Starts window minimized.

/max

Starts window maximized.

/separate

Starts 16-bit Windows programs in separate memory space.

/low

Starts application in the idle priority class.

/normal

Starts application in the normal priority class.

/high

Starts application in the high priority class.

/realtime

Starts application in the realtime priority class.

/wait

Starts application and waits for it to terminate.

/b

Does not create a new window. **CTRL+C** handling is ignored unless the application enables **CTRL+C** processing.

Use **CTRL+BREAK** to interrupt the application.

filename

Specifies the command or program to start.

parameters

Specifies parameters to pass to the command or program.

More Information About Start



[Start--Example](#)



Start--Example

To start the MYAPP program at the command prompt and retain use of the current window, type

start myapp

More Information About Start

 [Start](#)



Subst

Associates a path with a drive letter.

subst [*drive1*: [*drive2*:]*path*]

subst *drive1*: /**d**

Parameters

none

Used without parameters, **subst** displays the names of the virtual drives in effect.

drive1:

Specifies the virtual drive to which you want to assign a path.

drive2:

Specifies the physical drive that contains the specified path (if different from the current drive).

path

Specifies the path that you want to assign to a virtual drive.

/d

Deletes a virtual drive.

More Information About Subst



[Subst--Notes](#)



[Subst--Examples](#)



Subst--Notes

Using other commands with subst

The following commands do not work, or should not be used, on drives used in the subst command:

chkdsk **format** **restore**
 label
diskcomp
diskcopy **recover**

Valid *drive1* values

The *drive1* parameter must be within the range specified by the lastdrive command. If not, subst displays the following error message:

Invalid parameter - drive1:

More Information About Subst



[Subst--Examples](#)



[Subst](#)



Subst--Example

The following command creates a virtual drive Z for the path B:\USER\BETTY\FORMS:

```
subst z: b:\user\betty\forms
```

Now, instead of typing the full path, you can reach this directory by typing the letter of the virtual drive, followed by a colon, as in the following example:

z:

More Information About Subst



[Subst--Notes](#)



[Subst](#)



Switches

Forces an enhanced keyboard to behave like a conventional keyboard.

You use this command in your CONFIG.NT file.

switches=/k

More Information About Switches



[Switches--Notes](#)



[Switches--Examples](#)



Switches--Notes

When to use the Switches command

If you have a program that does not correctly interpret input from an enhanced keyboard, add this command to your CONFIG.NT file so your enhanced keyboard will use conventional keyboard functions.

Using the /k switch with ANSI.SYS

If you use the switches=/k command and you install the ANSI.SYS device driver, use the /k switch on the **device** command line for ANSI.SYS.

More Information About Switches

 [Switches--Examples](#)

 [Switches](#)



Switches--Example

If you want Windows NT to use conventional keyboard functions even though you are using an enhanced keyboard, add the following command to your CONFIG.NT file:

switches=/k

More Information About Switches



Switches--Notes



Switches



Time

Displays the system time or sets your computer's internal clock.

time [*hours*:[*minutes*[:*seconds*[.*hundredths*]]][**A|P**]

Parameters

none

Used without parameters, **time** displays the computer's clock time and prompts for the new time. Press enter to leave the time unchanged or type the new time using the syntax above.

hours

Specifies the hour. Valid values are in the range 0 through 23.

minutes

Specifies minutes. Valid values are in the range 0 through 59.

seconds

Specifies seconds. Valid values are in the range 0 through 59.

hundredths

Specifies hundredths of a second. Valid values are in the range 0 through 99.


A|P


Specifies A.M or P.M. for the 12-hour time format. If you type a valid 12-hour time but do not type A or P, TIME uses A (for A.M.).

See Also

 [Date](#)

More Information About Time

 [Time--Notes](#)

 [Time--Examples](#)



Time--Notes

Specifying an invalid time format

If you specify the time in an invalid format, Windows NT displays the following message and then waits for you to specify the time:

Invalid time

Enter new time: _

Changing the time format

You can change the **time** format by changing the setting in the Date/Time option in Control Panel. For for just the MS-DOS subsystem, change the **country** setting in your CONFIG.NT file. Depending on the country selected, Windows NT will display the time in the 12-hour format or the 24-hour format. If you are setting the time in the 12-hour format, be sure to specify P for hours after noon.

More Information About Time



[Time--Examples](#)



[Time](#)



Time--Examples

To set your computer's clock to 1:36 P.M., use either of the following commands:

time 13:36

time 1:36P

More Information About Time

 [Time--Notes](#)

 [Time](#)

**Title**

Sets the title for the command prompt window.

title [*string*]

Parameter

string

Specifies the title for the command prompt window.

More Information About Title



[Title--Notes](#)



[Title--Examples](#)



Title--Notes

Title is useful to set the window title for batch programs. Include the command at the beginning of a batch program.

Once set, the window title can be reset with the **title** command only.

More Information About Title

 [Title](#)

 [Title--Examples](#)



Title--Examples

To set the window title for a batch program:

```
rem This batch program updates the employee data
cls
@echo off
title Updating Files
copy \\server\share\*.xls c:\users\common\*.xls
echo Files Updated.
title Command Prompt
```

More Information About Title



[Title--Notes](#)



[Title](#)



Tree

Graphically displays the directory structure of a path or of the disk in a drive.

tree [*drive:*][*path*] [**/f**] [**/a**]

Parameters

drive:

Specifies the drive that contains the disk for which you want to display the directory structure.

path

Specifies the directory for which you want to display the directory structure.

/f

Displays the names of the files in each directory.

/a

Specifies that **tree** is to use text characters instead of graphic characters to show the lines linking subdirectories.

See Also



[Dir](#)

More Information About Tree



[Tree--Notes](#)




[Tree--Examples](#)



Tree--Note

The structure that **tree** displays depends upon the parameters you specify on the command line. If you do not specify a drive or path, **tree** displays the tree structure beginning with the current directory of the current drive.

More Information About Tree

 [Tree--Examples](#)

 [Tree](#)



Tree--Examples

To display the names of all the subdirectories on the disk in your current drive, type the following command:

**tree **

To display, one screen at a time, the files in all the directories on drive C, type the following command:

tree c:\ /f | more

To print the same list that the previous example displayed, type the following command:

tree c:\ /f prn

More Information About Tree



[Tree--Notes](#)



[Tree](#)



Type

Displays the contents of a text file.

Use the **type** command to view a text file without modifying it.

type [*drive:*][*path*] *filename* [...]


Parameter


[*drive:*][*path*] *filename*

Specifies the location and name of the file or files that you want to view. Separate multiple filenames with spaces.

If using an NTFS drive, and the filename contains spaces, you must enclose the filename with quotation marks (").

See Also


 [Filter commands](#)

 [Redirection symbols](#)

 [Dir](#)

More Information About Type

 [Type--Notes](#)

 [Type--Examples](#)



Type--Notes

Displaying binary files

If you display a binary file or a file created by a program, you may see strange characters on the screen, including formfeed characters and escape-sequence symbols. These characters represent control codes used in the binary file. In general, you should avoid using the **type** command to display binary files.

More Information About Type



[Type--Examples](#)



[Type](#)



Type--Examples


If you want to display the contents of a file named HOLIDAY.MAR, type the following command:

type holiday.mar

If the file you want to display is long, you can use the **more** command along with **type**, as shown in the following command, to display the file's contents one screen at a time:

type holiday.mar | more

More Information About Type

 [Type--Notes](#)

 [Type](#)



Ver

Displays the Windows NT version number.

Syntax

ver



Verify

Windows NT does not use this command. It is accepted only for compatibility with MS-DOS files.



Vol

Displays the disk volume label and serial number, if they exist.

A serial number is displayed for a disk formatted with MS-DOS version 4.0 or later.

vol [*drive:*]


Parameters

drive:

Specifies the drive that contains the disk for which you want to display the volume label and serial number.

See Also

 [Format](#)

 [Label](#)



Winnt

Performs an installation or upgrade of Windows NT 4.00.

winnt32 [/s:sourcepath] [/i:inf_file] [/t:drive_letter] [/x] [/b] [/ox] [/u[:script] [/r:directory]

Parameters

/s:sourcepath

Specifies the location of the Windows NT files.

/i:inf_file

Specifies the filename (no path) of the setup information file. The default is DOSNET.INF.

/t:drive_letter

Forces Setup to place temporary files on the specified drive.

/x

Prevents Setup from creating Setup boot floppies. Use this when you already have Setup boot floppies (from your administrator, for example).

/b

Causes the boot files to be loaded on the system's hard drive rather than on floppy disks, so that floppy disks do not need to be loaded or removed by the user.

/ox

Specifies that Setup create boot floppies for CD-ROM.

/u

Upgrades your previous version of Windows NT in unattended mode. All user settings are taken from the previous installation, requiring no user intervention during Setup.

/u:script

Similar to previous, but provides a script file for user settings rather than using the settings from the previous installation.

/r:directory

Installs an additional directory within the directory tree where the Windows NT files are installed. Use additional */r* switches to install additional directories.



Winnt32

Performs an installation or upgrade of Windows NT 4.00.

winnt32 [/s:sourcepath] [/i:inf_file] [/t:drive_letter] [/x] [/b] [/ox] [/u[:script] [/r:directory] [/e:command]

Parameters

/s:sourcepath

Specifies the location of the Windows NT files.

/i:inf_file

Specifies the filename (no path) of the setup information file. The default is DOSNET.INF.

/t:drive_letter

Forces Setup to place temporary files on the specified drive.

/x

Prevents Setup from creating Setup boot floppies. Use this when you already have Setup boot floppies (from your administrator, for example).

/b

Causes the boot files to be loaded on the system's hard drive rather than on floppy disks, so that floppy disks do not need to be loaded or removed by the user.

/ox

Specifies that Setup create boot floppies for CD-ROM.

/u

Upgrades your previous version of Windows NT in unattended mode. All user settings are taken from the previous installation, requiring no user intervention during Setup.

/u:script

Similar to previous, but provides a script file for user settings rather than using the settings from the previous installation.

/r:directory

Installs an additional directory within the directory tree where the Windows NT files are installed. Use additional */r* switches to install additional directories.

/e:command

Instructs Setup to execute a specific command after installation is complete.



Xcopy

Copies files and directories, including subdirectories.

xcopy *source* [*destination*] [/w] [/p] [/c] [/v] [/q] [/f] [/l] [/d[:*date*]] [/u] [/i] [/s [/e]] [/t] [/k] [/r] [/h] [/a|m] [/n] [/exclude:*filename*] [/z]

Parameters

source

Specifies the location and names of the files you want to copy. *Source* must include either a drive or a path.

destination

Specifies the destination of the files you want to copy. *Destination* can include a drive letter and colon, a directory name, a filename, or a combination.

/w

Displays the following message and waits for your response before starting to copy files:

Press any key to begin copying file(s)

/p

Prompts you to confirm whether you want to create each destination file.

/c

Ignores errors.

/v

Verifies each file as it is written to the destination file to make sure that the destination files are identical to the source files. This switch is ignored because the functionality is inherent to the Windows NT operating system. The switch is accepted only for compatibility with previous versions of MS-DOS.

/q

Suppresses display of **xcopy** messages.

/f

Displays source and destination filenames while copying.

/l

Does not copy files, only displays (lists) files that would be copied.

/d[:*date*]

Copies only source files changed on or after the specified date. If the *date* value is missing, **xcopy** copies all *source* files that are newer than the time of existing *destination* files. This option allows you to update only files that have changed. If you specify a date, use a "-" as the separator rather than a "/" so that the date is not interpreted as another parameter.

/u

Copies (updates) only files from *source* that exist on *destination*.

/i

If *source* is a directory or contains wildcards, and *destination* does not exist, **xcopy** assumes *destination* specifies a directory name and creates a new directory then copies all specified files into the new directory. By default, **xcopy** will prompt you to specify whether *destination* is a file or directory.

/s

Copies directories and subdirectories, unless they are empty. If you omit this switch, **xcopy** works within a single directory.

/e

Copies all subdirectories, even if they are empty. Used with the **/s** and **/t** switches.

/t

Copies only subdirectory structure (tree), not files. To copy empty directories, you must include the **/e** switch.

/k

Copies files and retains the read-only attribute on destination files if present on the source files. By default, the read-only attribute is removed.

/r

Copies over read-only files.

/h

Copies files with the hidden and system file attributes. **Xcopy** will not copy hidden or system files by default.

/a

Copies only source files that have their archive file attributes set. This switch does not modify the archive file attribute of the source file. For information about how to set the archive file attribute, see the **attrib** command.

/m

Copies source files that have their archive file attributes set. Unlike the **/a** switch, the **/m** switch turns off archive file attributes in the files specified in source. For information about how to set the archive file attribute, see the **attrib** command.

/n

Copies using NTFS short file or directory names. This switch is required when copying files or directories from an NTFS volume to a FAT volume or when the FAT file system naming convention (8.3) is required on the destination volume. The destination file system may be FAT or NTFS.

/exclude:filename

Excludes the files listed in the specified file from the copy operation. The exclusion file can have a list of exclusion patterns (one per line, no wild card characters are supported). If any exclusion pattern in the file matches any part of the path of a subject file, that file is not copied.

/z

Copies over a network in restartable mode.


See Also


 [Copy](#)

 [Move](#)

 [Dir](#)

More Information About Xcopy

 [Xcopy--Notes](#)

 [Xcopy--Examples](#)



Xcopy--Notes

Default value for destination

If you omit destination, the **xcopy** command copies the files to the current directory.

Specifying whether destination is a file or directory

If destination does not contain an existing directory and does not end with a backslash (\), **xcopy** prompts you with a message in the following format:

Does destination specify a file name
or directory name on the target
(F = file, D = directory)?

Press **F** if you want the file(s) to be copied to a file. Press **D** if you want the file(s) to be copied to a directory.

You can avoid this prompt by using the **/i** switch. If the **/i** switch is used **xcopy** assumes the destination is a directory if the source is more than one file or a directory.

Xcopy sets archive attribute for destination files

Xcopy creates files with the archive attribute set, whether or not this attribute was set in the source file. For more information about file attributes, see the [attrib](#) command.

Xcopy vs. diskcopy

If you have a disk that contains files in subdirectories and you want to copy it to a disk that has a different format, you should use the **xcopy** command instead of **diskcopy**. Since the **diskcopy** command copies disks track by track, it requires that your source and destination disks have the same format. **Xcopy** has no such requirement. In general, use **xcopy** unless you need a complete disk image copy.

Xcopy exit codes

The following list shows each exit code and a brief description of its meaning:

- 0 Files were copied without error.
- 1 No files were found to copy.
- 2 The user pressed CTRL+C to terminate **xcopy**.
- 4 Initialization error occurred. There is not enough memory or disk space, or you entered an invalid drive name or invalid syntax on the command line.
- 5 Disk write error occurred.

You can use the **errorlevel** parameter on the **if** command line in a batch program to process exit codes returned by **xcopy**. See [Examples](#).

More Information About Xcopy



[Xcopy--Examples](#)



[Xcopy](#)



Xcopy--Examples

To copy all the files and subdirectories (including any empty subdirectories) from the disk in drive A to the disk in drive B:

xcopy a: b: /s /e

To include any system or hidden files in the previous example, add the /h switch:

xcopy a: b: /s /e /h

To update files in the REPORTS directory with the files in the directory RAWDATA that have changed since December 29, 1993

xcopy \rawdata \reports /d:12-29-93

To update all the files that exist on \REPORTS in the previous example, regardless of date, type

xcopy \rawdata \reports /u

To obtain only a list of the files that would be copied for the previous command, without copying the files, type

xcopy \rawdata \reports /d:12-29-93 /l > xcopy.out

The file XCOPY.OUT lists every file that would be copied.

To copy the \CUSTOMER directory and all subdirectories, including empty directories, to the directory \PUBLIC\ADDRESS on network drive H: and retain the read-only attribute, while being prompted when a new file will be created on H:, type

xcopy \customer h:\public\address /s /e /k /p

To issue the previous command and ensure xcopy creates the directory \ADDRESS if it does not exist, without prompting, add the /i switch:

xcopy \customer h:\public\address /s /e /k /p /i

You can create a batch program to perform xcopy operations and use the batch if command to process the exit code in case an error occurs. For example, the following batch program uses replaceable parameters for the xcopy source and destination parameters:


```
@echo off  
rem COPYIT.BAT transfers all files in all subdirectories of  
rem the source drive or directory (%1) to the destination  
rem drive or directory (%2)  
  
xcopy %1 %2 /s /e  
  
if errorlevel 4 goto lowmemory  
if errorlevel 2 goto abort  
if errorlevel 0 goto exit  
  
:lowmemory  
echo Insufficient memory to copy files or  
echo invalid drive or command-line syntax.  
goto exit  
  
:abort  
echo You pressed CTRL+C to end the copy operation.  
goto exit  
  
:exit
```


To use this batch program to copy all files in the C:\PRGMCODE directory and its subdirectories to drive B, type the following command:

copyit c:\prgmscode b:

The command interpreter substitutes C:\PRGMSCODE for %1 and B: for %2, then uses **xcopy** with the /e and /s switches. If **xcopy** encounters an error, the batch program reads the exit code and goes to the label indicated in the appropriate **if errorlevel** statement. Windows NT displays the appropriate message and exits from the batch program.

More Information About Xcopy

 [Xcopy--Notes](#)

 [Xcopy](#)



What's New or Different from MS-DOS

Windows NT retains and enhances almost all the functionality of MS-DOS. These tables explain new Windows NT commands, changes to MS-DOS commands, and unavailable MS-DOS commands.

Windows NT Commands

This table explains Windows NT system commands not found in MS-DOS.

Command	Function
At	Schedules commands and programs to run on a computer at a specified time and date.
Cacls	Displays or modifies access control lists (ACLs) of files.
Convert	Converts file systems from FAT to NTFS.
Diskperf	Starts, stops, and displays system disk performance counter use.
Dosonly	Prevents starting applications other than MS-DOS-based applications from the COMMAND.COM prompt.
Echoconfig	Displays messages when reading the MS-DOS subsystem CONFIG.NT file.
Endlocal	Ends localization of environment variables.
Findstr	Searches for text in files using regular expressions.
Ntcmdprompt	Runs the Windows NT command interpreter, CMD.EXE, rather than COMMAND.COM after running a TSR or after starting the command prompt from within an MS-DOS application.
Popd	Changes to the directory last set with the pushd command.
Pushd	Saves the current directory for use by the popd command, and then changes to the specified directory.
Setlocal	Begins localization of environmental variables.
Start	Runs a specified program or command in a secondary window and in its own memory space.
Title	Sets the title of the command prompt window.
&&	Command following this symbol runs only if the command preceding the symbol succeeds.
 	Command following this symbol runs only if the command preceding the symbol fails.

&	Separates multiple commands on the command line.
()	Groups commands.
^	Escape character. Allows input of command symbols as text.
; or ,	Separates parameters.

Changes to MS-DOS Commands

This table lists changes and improvements to MS-DOS commands.

Command	Changed features
chcp	Changes code pages for full-screen mode only.
cmd	CMD.EXE replaces COMMAND.COM.
del	New switches provide many more functions.
dir	New switches provide many more functions.
diskcomp	Switches /1 and /8 are not supported.
diskcopy	Switch /1 is not supported.
doskey	Available for all character-based programs that accept buffered input. Several other enhancements improve doskey .
format	20.8 MB floptical drive supported. Switches /b , /s , and /u are not supported.
keyb	KEYBOARD.SYS is no longer used.
label	The symbols ^ and & can be used in a volume label.
mode	Extensive changes.
more	New switches provide many more functions.
path	The %PATH% environment variable appends the current path to a new setting at the command prompt.
print	Switches /b , /u , /m , /s , /q , /t , /c , and /p are not supported.
prompt	New character combinations allow you to add ampersands (\$a), parentheses (\$c and \$f), and spaces (\$s) to your prompt.
recover	Recovers files only.
rmdir	New /s switch deletes directories containing files and subdirectories.
sort	Does not require TEMP environment variable. File size is unlimited.

xcopy New switches provide many more functions.

Unavailable MS-DOS Commands

The following MS-DOS commands are not available at the command prompt.

Command	New procedure or reason for obsolescence
assign	Not supported in Windows NT.
choice	The choice command is not currently supported.
ctty	Not currently supported.
dblspace	The Dblspace program is not supported.
defrag	Windows NT automatically optimizes disk use.
deltree	The rmdir /s command deletes directories containing files and subdirectories.
dosshell	Unnecessary with Windows NT.
drvspace	The Drvspace program is not currently supported.
emm386	Unnecessary with Windows NT.
fasthelp	This MS-DOS 6.0 command is the same as the Windows NT command help . Windows NT also provides an online Command Reference in the Windows NT Help icon in the Main program group.
fdisk	Disk Administrator prepares hard disks for use with Windows NT.
include	Multi-configurations of the MS-DOS subsystem are not supported.
interlnk	The Interlnk program is not supported.
intersrv	The Intersrv program is not supported.
join	Increased partition size and an improved file system eliminate the need to join drives.
memmaker	Windows NT automatically optimizes the MS-DOS subsystem's memory use.
menucolor	Multi-configurations of the MS-DOS subsystem are not supported.
menudefault	Multi-configurations of the MS-DOS subsystem are not supported.
menuitem	Multi-configurations of the MS-DOS

	subsystem are not supported.
mirror	Not supported in Windows NT.
msav	The Msav program is not supported.
msbackup	Windows NT provides the Backup utility (in the Administrative Tool group) for computers with tape drives, or the backup and xcopy commands for computers without tape drives.
mscdex	It is unnecessary to configure the MS-DOS subsystem to use a CD-ROM drives. Windows NT provides access to CD-ROM drives for the MS-DOS subsystem.
msd	Use the Windows NT Diagnostics program in the Administrative Tools group.
numlock	The numlock command is not currently supported.
power	The Power utility is not supported.
scandisk	The Scandisk utility is not supported.
smartdrv	Windows NT automatically provides caching for the MS-DOS subsystem.
submenu	Multi-configurations of the MS-DOS subsystem are not supported.
sys	Windows NT will not fit on a standard 1.2 MB or 1.44 MB floppy disk.
undelete	Not supported in Windows NT.
unformat	Not supported in Windows NT.
vsafe	The Vsafe program is not supported.



What's New or Different from LAN Manager

The following table lists commands that have changed or been added since LAN Manager version 2.2. See [TCP/IP utilities](#) for a list of TCP/IP commands.

New or Changed Commands from LAN Manager

Command	Feature
at	In addition to local scheduling, you can remotely schedule events on a computer.
lpxroute	New command supports routing for the NWLink protocol on a token-ring network.
net accounts	Server roles cannot be set. Windows NT security controls lockout.

net computer	This new command allows you to add or delete computers from a domain database. This command is available only on computers running Windows NT Server.
net config	Peer functionality is inherent to Windows NT. Separate commands, such as net config peer are no longer required.
net config server	Most network services are self-configuring. The switches /autodisconnect , /srvcomment , and /hidden can be configured. The switch /srvhidden is now /hidden .
net config workstation	Most network services are self-configuring. The switches /charcount , /chartime , and /charwait can be configured.
net continue	Use printer property sheets to control printing. See net pause in this table for the list of pausable services that can be continued.
net group	Net group manages global groups and is only for computers that are members of a domain.
net localgroup	This new command manages local groups.
net pause	You can pause these network services: file server for macintosh and remoteboot (Windows NT Server only), ftp publishing service , lpdsvc , net logon , network dde , network dde dsdm , nt lm security support provider , remote access server , schedule , server , simple tcp/ip services , or workstation . Use printer property sheets to manage printers.
net print	Use printer property sheets to manage printers. (Right-click on a printer and click Properties .)
net send	Sending files is not supported. The /broadcast switch is not supported.

net share Remote administration is automatic. Use printer property sheets to share printers. Comm queues are not supported in this release.

net start You can start these network services: **alerter, client service for netware, clipbook server, computer browser, dhcp client, directory replicator, eventlog, ftp publishing service, lpdsvc, messenger, net logon, network dde, network dde dsdm, network monitor agent, nt lm security support provider, ole, remote access connection manager, remote access isnsap service, remote access server, remote procedure call (rpc) locator, remote procedure call (rpc) service, schedule, server, simple tcp/ip services, snmp, spooler, tcp/ip netbios helper, ups, and workstation.**

These services are available only on Windows NT Server: **file server for macintosh, gateway service for netware, microsoft dhcp server, print server for macintosh, remoteboot, windows internet name service.**

Note

that services can be configured to start automatically.

net start alerter Self-configuring.

net start eventlog This new service logs any significant system, security, and application occurrences that require users to be notified.

net start messenger Self-configuring.

net start net logon Self-configuring.

net start directory replicator Use Server Manager to configure the Replicator service.

net start server Use Server Manager to configure the Server service.

net start snmp	New options permit logging.
net start workstation	The Workstation is configured at setup and in various applications.
net statistics	Peer functionality is an inherent part of Windows NT and no longer requires separate commands. The statistics log cannot be cleared.
net stop	You can stop all network services except eventlog . See net start for a list of services that can be started and stopped.
net use	The /persistent switch has only yes and no values. Comm queues are not supported in this release. You can use NetWare volumes.
net user	Switches /logonserver , /maxstorage , /operator , and /privilege are not supported.
net view	New switch /domain permits viewing of domains and viewing of computers in a specified domain. You can view servers on NetWare networks.

Superseded or Obsolete Commands from LAN Manager

The LAN Manager commands below are no longer available at the command prompt. The following table provides alternatives, or explains why the command no longer exists.

Command	Action or explanation
net	Use the Windows NT administrative tools to administer the network.
net admin	Use the Windows NT administrative tools to administer the network.
net access	Use property sheets to set permissions on files on NTFS disk partitions.
net audit	Use the Event Viewer to keep track of network resource use.
net comm	Communication-device queues are not currently supported.
net config peer	Not needed with Windows NT.
net console	Use Windows NT security features to prevent unwanted access.

net copy	Use Windows NT Explorer or the system copy command to copy files.
net device	Use printer property sheets to display information about shared printers.
net error	Use the Event Viewer to keep track of network resource use.
net forward	You cannot forward network messages with Windows NT.
net log	Message logging is not supported.
net logoff	Logging off is an inherent part of Windows NT.
net logon	Logging on is an inherent part of Windows NT.
net move	Use Windows NT Explorer or the system copy and delete commands to move files.
net password	Press CTRL+ALT+DEL to change your password.
net run	Programs cannot be executed remotely in Windows NT.
net separator	Use printer property sheets to control separator pages.
net start netpopup	Windows NT processes network messages automatically.
net start netrun	This command is not supported in Windows NT.
net start nvalert	The Nvalert service is not currently supported.
net start peer	Peer functionality is an inherent part of Windows NT.
net start timesource	The functionality is included in the Server service.
net status	The information can be found using net config server and net share .
net version	In Control Panel, use the System application's General tab to determine the current version you are using.
net who	Use Server Manager to view users logged on to a server.



Batch Commands

Select a command to get more information.

call

echo

endlocal

for

goto

if

pause

rem

setlocal

shift

Batch programs (also called batch files) allow you to simplify routine or repetitive tasks. A *batch program* is an unformatted text file that contains one or more commands and has a .BAT or .CMD filename extension. When the filename is typed at the command prompt, the commands in the file are executed sequentially.

Any command can be included in a batch file. In addition, the **for**, **goto**, and **if** commands allow conditional processing of the commands in the batch file. For example, the **if** command carries out a command based on the results of a condition. Other commands allow you to control input and output and call other batch programs.



MS-DOS Configuration Commands

Select a command to get more information.

- | | |
|---|--|
| <input type="checkbox"/> buffers | <input type="checkbox"/> fcbs |
| <input type="checkbox"/> country | <input type="checkbox"/> files |
| <input type="checkbox"/> device | <input type="checkbox"/> install |
| <input type="checkbox"/> devicehigh | <input type="checkbox"/> lastdrive |
| <input type="checkbox"/> dos | <input type="checkbox"/> ntcmdprompt |
| <input type="checkbox"/> dosonly | <input type="checkbox"/> shell |
| <input type="checkbox"/> driveparm | <input type="checkbox"/> stacks |
| <input type="checkbox"/> echoconfig | <input type="checkbox"/> switches |

You configure the MS-DOS subsystem with configuration commands, such as **device** or **lastdrive**. Place these commands in the CONFIG.NT file in the `\systemroot\SYSTEM32` directory or the Config file specified by an application's PIF . These commands affect only the MS-DOS subsystem. Many of them, such as **buffers** and **break**, are ignored because the MS-DOS subsystem works without them. They are accepted for compatibility only.



MS-DOS Subsystem Commands

Select a command to get more information.

<input type="checkbox"/> append	<input type="checkbox"/> fastopen
<input type="checkbox"/> backup	<input type="checkbox"/> graphics
<input type="checkbox"/> debug	<input type="checkbox"/> loadfix
<input type="checkbox"/> debug_commands	<input type="checkbox"/> loadhigh (lh)
<input type="checkbox"/> edit	<input type="checkbox"/> mem
<input type="checkbox"/> edlin	<input type="checkbox"/> nlsfunc
<input type="checkbox"/> edlin_commands	<input type="checkbox"/> qbasic
<input type="checkbox"/> exe2bin	<input type="checkbox"/> setver
<input type="checkbox"/> expand	<input type="checkbox"/> share

Windows NT includes 16-bit commands (non-native) for the MS-DOS and other subsystems. These are older commands, such as **edlin** or **graphics**, and MS-DOS-specific commands, such as **debug** or **exe2bin**. These 16-bit commands are included to maintain MS-DOS and MS OS/2 version 1.x compatibility.

Other MS-DOS subsystem commands, such as **share**, perform functions that are now inherent to Windows NT or the MS-DOS subsystem. The commands are accepted to preserve compatibility with existing files, but the commands have no effect because the functionality is automatic in Windows NT.




OS/2 Configuration Commands

Select a command to get more information.

 [Codepage](#)

 [Libpath](#)

 [Devinfo](#)


 [Protshell](#)


You configure the OS/2 subsystem with CONFIG.SYS commands, such as **devicename** or **libpath**. Use an OS/2 editor to edit C:\CONFIG.SYS. These commands affect only the OS/2 subsystem.




Command Symbols and Filter Commands

Select a topic to get more information.

 [Redirection \(> <>>\)](#)

 [Filter Commands \(| **more**, **sort**, **find**\)](#)

 [Conditional Processing Symbols \(& && || \(\) ^\)](#)



Redirection

Redirection characters change where a command gets information from or sends information to.

Redirecting Command Input and Output

Unless you specify otherwise, Windows NT receives input from your keyboard and sends output to your screen. Sometimes it is useful to redirect the input or output to a file or a printer. For example, you might want to redirect a directory listing from the screen to a file.

To redirect the input or output of a command, you use one of the following redirection characters:



The greater-than sign (>) sends the output of a command to a file or a device, such as a printer.



The less-than sign (<) takes the input needed for a command from a file rather than from the keyboard.



The double greater-than sign (>>) adds output from a command to the end of a file without deleting the information already in the file.

Redirecting the Output of a Command

Almost all commands send output to your screen. Even commands that send output to a drive or printer also display messages and prompts on your screen.

To redirect the output from the screen to a file or printer, use the greater-than sign (>). You can use the greater-than sign with most Windows NT commands. For example, in the following command, the directory listing produced by the **dir** command is redirected to the DIRLIST.TXT file:

```
dir > dirlist.txt
```

If the DIRLIST.TXT file doesn't exist, Windows NT creates it. If DIRLIST.TXT exists, Windows NT replaces the information in the file with the output from the **dir** command.

To add the output from a command to the end of a file without losing any of the information already in the file, use a double greater-than sign (>>). For example, in the following command, the directory listing produced by the **dir** command is appended to the DIRLIST.TXT file:

```
dir >> dirlist.txt
```

Note



Some command output, such as error messages, may not be redirected when using the greater-than sign (>).

Redirecting the Input to a Command

Just as you can send the output of a command to a file or printer rather than to your screen, you can take the input for a command from a file rather than from the keyboard. To take input from a file, use the less-than sign (<). For example, the following command takes the input for the **sort** command from the LIST.TXT file:

```
sort < list.txt
```

Windows NT alphabetizes the lines of the LIST.TXT file and displays the result on your screen.



Filter Commands

Filter commands help you sort, view, and select parts of the output of a command.

Passing Information Through Filter Commands

Filter commands divide, rearrange, or extract portions of the information that passes through them. Windows NT has three filter commands:



The more command displays the contents of a file or the output of a command one screen at a time.



The find command searches through files and command output for the characters you specify.



The sort command alphabetizes files and command output.

To send input from a file to a filter command, use the less-than sign (<). If you want the filter command to get its input from another command, use the pipe (|).

Controlling the Screen Display by Using the More Command

The **more** command displays the contents of a file or the output of a command one screen at a time. For example, the following **more** command displays the contents of the LIST.TXT file one screen at a time:

```
more < list.txt
```

After a screen of information is displayed, the word "More" appears. To continue to the next screen, press any key. To stop the command without viewing more information, press CTRL+C.

The **more** command is helpful if you are working with a command that produces more than one screen of output. For example, suppose you want to view a directory tree for your hard disk. If you have more directories than Windows NT can display on the screen, you can use the **tree** command with a pipe (|) and a **more** command, as in the following example:

```
tree c:\ | more
```

The first screen of output from the **tree** command is displayed, followed by the word "More." Windows NT pauses until you press any key (except the PAUSE key).

Searching for Text by Using the Find Command

The **find** command searches one or more files for the text you specify. Windows NT displays every line containing that text. The **find** command can be used as a filter command or as a standard Windows NT command. For information about using **find** as a standard Windows NT command, see [find](#).

To use **find** as a filter command, include a less-than sign (<) and a filename to search through. (The search is case-sensitive.) For example, the following command finds occurrences of the string "Pacific Rim" in the file TRADE.TXT:

```
find "Pacific Rim" < trade.txt
```

To save the output of the **find** command rather than display it, use a greater-than sign (>) and the name of the file that is to store the output. For example, the following command finds occurrences of "Pacific Rim" in the TRADE.TXT file and saves them in the NWTRADE.TXT file:

```
find "Pacific Rim" < trade.txt > nwtrade.txt
```

Sorting Text Files

The **sort** command alphabetizes a text file or the output of a command. For example, you would use the following command to sort the contents of a file named LIST.TXT and display the results on your screen:

```
sort < list.txt
```

In this example, the **sort** command sorts the lines of the LIST.TXT file and displays the results without changing the file. To save the output of the **sort** command rather than display it, include a greater-than sign (>) and a filename in the command. For example, you would use the following command to alphabetize the lines of the LIST.TXT file and store the results in the ALPHLIST.TXT file:

```
sort < list.txt > alphlist.txt
```

To sort the output of a command, type the command followed by a pipe (|) and the **sort** command. For example,

the following command sorts the output of the **find** command:

```
find "Jones" mailst.txt | sort
```

When you type this command, Windows NT lists in alphabetic order the lines in which the string "Jones" appears.

Combining Commands with Redirection Characters

You can combine filter commands, other commands, and filenames to make custom commands. For example, you could use the following command to store the names of files that contain the string "LOG":

```
dir /b | find "LOG" > loglist.txt
```

Windows NT sends the output of the **dir** command through the **find** filter command and stores the filenames that contain the string "LOG" in the LOGLIST.TXT file. The results are stored as a list of filenames (for example, A.LOG, LOGDAT.SVD, and MYLOG.BAT).

To use more than one filter in the same command, separate the filters with a pipe (|). For example, the following command searches every directory on drive C, finds the filenames that include the string "LOG", and displays them one screen at a time:

```
dir c:\ /s /b | find "LOG" | more
```

Because you use a pipe (|), Windows NT sends the output of the DIR command through the **find** command. The **find** command selects only filenames that contain the string "LOG". The **more** command displays the filenames that are selected by the **find** command, one screen at a time.



Conditional Processing Symbols

Conditional processing symbols allow you to control the execution of commands.

Processing Commands Conditionally

Conditional processing symbols allow you to issue multiple commands from the same prompt and to act based on the results of a command.



The ampersand (&) separates multiple commands on one command line.



The parentheses groups multiple commands.



The semicolon or comma (; ,) separate command parameters.



The caret (^) allows you to use a command symbol as text (ignores the symbols special meaning).



The double ampersand (&&) causes the command following this symbol to run only if command preceding symbol is successful.



The double pipe (||) causes the command following this symbol to run only if command preceding symbol fails.



Edlin

Starts Edlin, a line-oriented text editor with which you can create and change ASCII files.

Edlin numbers each line of the text file that is located in memory. You can use Edlin to insert, modify, copy, move, and delete lines of the file. If you want to use a full-screen editor, use the **edit** command.

edlin [*drive:*][*path*] *filename* [**/b**]

Parameters

[*drive:*][*path*] *filename*

Specifies the location and name of an ASCII file on a disk. If the file exists, Edlin opens it. If the file does not exist, Edlin creates a file in memory and uses the specified location and filename to create the file on a disk when you use the Edlin **e** command.

/b

Specifies that Edlin is to ignore the end-of-file character (CTRL+Z).

More Information About Edlin



[Edlin--Note](#)



[Edlin Commands](#)



Edlin--Note

Maximum line length

Edlin accepts a maximum of 253 characters per line.

Edlin commands

The following is a list of Edlin commands with a brief description of each command:

Command	Description
<i>line</i>	Displays the line you specify.
?	Displays a list of Edlin commands.
A	Loads a portion of a file into memory when insufficient memory prohibits loading the entire file.
C	Copies a block of consecutive lines to the line number you specify.
D	Deletes a block of consecutive lines.
E	Writes the edited file from memory to a disk (saves the file), and stops the Edlin session.
I	Inserts one or more lines.
L	Displays a block of consecutive lines.
M	Moves a block of consecutive lines.
P	Displays a file one page at a time.
Q	Stops the Edlin session without writing the edited file from memory to a disk.
R	Searches for a string of one or more characters, and replaces it.
S	Searches for a string of one or more characters.

- T** Merges the contents of another file on a disk with the contents of the file that is in memory.
- w** Writes the first portion of the file in memory to a disk.

Meaning of the asterisk character in Edlin

The asterisk (*) is used for two purposes in Edlin. When an asterisk appears as the only character on the display line, it is the Edlin prompt after which you type Edlin commands. When an asterisk appears after a line number on the display line, it indicates that the line is the current line (where the cursor is located).

Meaning of a page of text

A page of text is one full screen of information. With a 25-line screen mode, Edlin displays 24 lines of text per page. The number of lines per page depends on the screen mode you are using.


Starting and stopping insert mode

To insert lines into the file in memory, use the Edlin **I** (Insert) command. Once you have finished inserting lines, press ENTER and then CTRL+C to stop insert mode. For more information about inserting lines, see the [Edlin I \(Insert\)](#) command.

Editing keys

Windows NT provides several editing keys that you can use to edit the file in memory.

More Information About Edlin

 [Edlin Commands](#)

 [Edlin](#)

**Edlin: [line]**

Displays the line of text you specify.

When you type a line number as a command, Edlin displays two lines. The first line contains the number you specified and its associated text. The second line contains the number again, followed by the cursor. The text on the first line of the display serves as a template for the second line. On the second line of the display, you can press ENTER to cancel the command without changing the text, type replacement text, or edit the line of text.

[line]

Parameters

line

Specifies the number of the line you want Edlin to display. To see the number and text of the current line, press ENTER.

More Information About Edlin: [line]



Edlin: [line]--Note



Edlin: [line]--Example




Edlin: [line]--Note

Entering changes into memory

After you edit a line, press ENTER to enter the changes into memory.


Caution

 If you press ENTER while the cursor is in the middle of a line, Edlin deletes the portion of the line that is between the cursor and the end of the line.

For information about saving the edited file from memory to a disk, see the [Edlin E \(End\)](#) and [Edlin W \(Write\)](#) commands.

More Information About Edlin: [line]

 [Edlin: \[line\]--Example](#)

 [Edlin: \[line\]](#)



Edlin: [line]--Example

Suppose that the following file is in memory and ready to edit. When you use the Edlin **I** (list) command at the Edlin prompt, Edlin displays the contents of the file.

```
1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6: impressed with your work.
```

To edit line 6, type **6**. Edlin displays the following two lines:

```
6:*impressed with your work.
6:*_
```

The first line contains the specified line number and its associated text. The second line contains the same line number and the cursor.

Now suppose you want to insert the word "fine" before the word "work" in the previous example. You can specify that Edlin is to redisplay a portion of the first line, beginning at the cursor position on the second line. First, press **F2** and type **w**. Edlin displays up to, but not including, the first "w" in line 6, as follows:

```
6:*impressed _
```

Then, press **F2** and type **w** again. Edlin displays up to, but not including, the next "w" in line 6, as follows:

```
6:*impressed with your _
```

Now press the **INSERT** key and type **fine** and then **a space**. Then press the **F3** key. Edlin displays the edited line, as follows:

```
6:*impressed with your fine work._
```

Press **ENTER** to accept the change.

At the Edlin prompt, use the Edlin **I** (list) command to see a display of the edited file now in memory. Edlin displays the following:

```
1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6:*impressed with your fine work.
```

More Information About Edlin: [line]



[Edlin: \[line\]--Note](#)



[Edlin: \[line\]](#)



Edlin: A (Append)

Loads a portion of a file into memory when insufficient memory prevents Edlin from loading the entire file.

When you start Edlin, it reads as many lines as possible from your disk file into memory. If the size of your file exceeds available memory, you must edit your file in stages. That is, you edit the first part of the file, write that part of the file to your disk by using the **w** (write) command, and then load more unedited lines from your disk into memory.

[*n*]a

Parameters

n

Specifies the number of lines you want Edlin to read into memory from the disk.

More Information About Edlin: A



[Edlin: A--Notes](#)



[Edlin: A--Example](#)



Edlin: A (Append)--Notes

Default setting

If you do not specify a value for n , Edlin loads lines from the disk file until available memory is 75-percent full. If available memory is already 75-percent full, Edlin loads no lines.

End-of-file message

After the **a** command reads the last line of the file into memory, Edlin displays the following message:

End of input file

More Information About Edlin: A



[Edlin: A--Example](#)



[Edlin: A](#)




Edlin: A (Append)--Example

Suppose the last 100 lines of your disk file do not fit into memory. After you edit the first part of the file and write a portion of it back to a disk, you can type the following command to load the remaining 100 lines into memory:

100a

More Information About Edlin: A

 [Edlin: A--Notes](#)

 [Edlin: A](#)



Edlin: C (Copy)

Copies a block of consecutive lines to one or more locations within the file in memory.

The **c** command copies the block of consecutive lines you specify to a line number you specify. This block can be copied as many times as necessary.

*[line1],[line2],line3,[count]***c**

Parameters

line1

Specifies the first line you want Edlin to copy.

line2

Specifies the last line you want Edlin to copy.

line3

Specifies the line before which Edlin is to insert the specified block of lines.

count

Specifies the number of times you want Edlin to copy the block of lines.

More Information About Edlin: C



[Edlin: C--Notes](#)



[Edlin: C--Examples](#)



Edlin: C (Copy)--Notes

Default settings

If you omit *line1* or *line2*, Edlin copies only the current line. You must include the commas on the command line even if you omit one or both of these parameters.

If you omit the *count* parameter, Edlin copies the lines one time.

Line renumbering

After Edlin copies lines, you can use the Edlin **l** (list) command at the Edlin prompt to see the correctly renumbered lines.

Overlapping line numbers

The line you specify for the *line3* parameter cannot be part of the block of lines to be copied. If you overlap line numbers in this way, Edlin cannot complete the copy operation and displays the following message:

Entry error

For example, the following command results in an error message:

```
3,20,15c
```

More Information About Edlin: C



[Edlin: C--Examples](#)



[Edlin: C](#)



Edlin: C (Copy)--Examples

If you type the following command, Edlin copies lines 1 through 5 one time, beginning on line 6:

```
1,5,6c
```

Lines 6 through 10 become identical to lines 1 through 5.

To copy the current line to line 5, use the following command:

```
„5c
```

More Information About Edlin: C



[Edlin: C--Notes](#)



[Edlin: C](#)



Edlin: D (Delete)

Deletes the block of consecutive lines you specify.

*[line1][,line2]***d**

Parameters

line1

Specifies the first line you want Edlin to delete.

line2

Specifies the last line you want Edlin to delete.

More Information About Edlin: D



[Edlin: D--Notes](#)



[Edlin: D--Examples](#)



Edlin: D (Delete)--Notes

Default parameter values

If you omit both parameters or only the *line2* parameter, Edlin deletes the current line. However, if you omit only the *line1* parameter, Edlin deletes the block of text that includes the current line through the line whose number is specified for *line2*. In the latter case, you cannot specify a line number for *line2* that precedes the current line number. In general, the number you specify for *line2* cannot be smaller than the number you specify for *line1*. If you omit only the *line1* parameter, you need to insert a comma as a placeholder preceding *line2*, as shown in the syntax line.

Line renumbering

After Edlin deletes lines, you can use the Edlin **l** (list) command at the Edlin prompt to see the correctly renumbered lines that remain.

More Information About Edlin: D



[Edlin: D--Examples](#)



[Edlin: D](#)



Edlin: D (Delete)--Examples

If you want Edlin to delete line 7, type:

7d

If you want Edlin to delete the block of text on lines 22 through 32, type the following command:

22,32d

Finally, suppose that the number of the current line is 7. To specify that Edlin is to delete the block of text that includes the current line through line 11, type the following command:

,11d

More Information About Edlin: D



[Edlin: D--Notes](#)



[Edlin: D](#)



Edlin: E (End)

Writes the current file from memory to a disk and stops the Edlin session.

The **e** command renames the original input file on the disk with the .BAK extension, writes the edited file from memory to the original input file on the disk, and then stops the Edlin session. However, if the file in memory is one that you created during this session rather than one that Edlin loaded from a disk, Edlin does not create a backup (.BAK) file on the disk.

e

More Information About Edlin: E



[Edlin: E--Notes](#)



Edlin: E (End)--Notes

Default drive and directory

Edlin writes the edited file from memory to the drive, directory, and filename on a disk that you specified when you started the current Edlin session. If you omitted a drive name at that time, Edlin writes to the current drive. If you omitted a directory name at that time, Edlin writes to the current directory.

Checking for disk space

Before using the **e** command, you should be sure your disk contains enough free space for the entire edited file that is in memory. If it does not, Edlin loses part or all of the file.

Read-only .BAK file

Suppose you want Edlin to save an edited file from memory to a disk, but the .BAK version of the file is a read-only file. In this case, Edlin displays a message in the following format to inform you that Edlin cannot replace the .BAK file:

```
Access denied - [drive:][path] filename.BAK
```

Both the original and backup versions of your file on the disk remain unchanged.

More Information About Edlin: E



Edlin: E



Edlin: I (Insert)

Inserts lines before the line number you specify in the edited file in memory.

If you are creating a new file, you must type the **i** command before you can insert a new line. Edlin displays the next line number each time you press ENTER. Edlin remains in insert mode until you press CTRL+C.

*[line]*i

Parameters

line

Specifies the line number before which you want Edlin to insert lines. The default value of *line* is the number of the current line.

More Information About Edlin: I



[Edlin: I--Notes](#)



[Edlin: I--Example](#)



Edlin: I (Insert)--Notes

Line renumbering

When you quit insert mode, the line immediately following the inserted lines becomes the current line. You can use the Edlin I (list) command at the Edlin prompt to see the correctly renumbered lines.

Inserting control characters

To insert a control character in text, type ^V followed by the ASCII symbol that represents the control character. For example, to insert an escape character (CTRL+[), type the following:

^V[

To insert a character that produces a tone (CTRL+G), type the following:

^VG

Appending text

If the value for *line* exceeds the number of lines in the file you are editing or if you specify a number sign (#) for *line*, Edlin appends the inserted line(s) to the end of the file. In either case, the last line you insert becomes the current line. If only a portion of the file is in memory, the line is appended at the end of the portion in memory.

More Information About Edlin: I



[Edlin: I--Example](#)



[Edlin: I](#)



Edlin: I (Insert)--Example

Suppose you have used the Edlin I (list) command at the Edlin prompt to display the following text on your screen:

1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6: impressed with your work.
7:
8: Sincerely,
9:
10: S.L. Martin, President

Suppose you want to add another paragraph to the letter. To insert text before line 8, type **8I**. Edlin displays the following:

8:*_

Now type the following line at the cursor on line 8:

8:***I think you will enjoy working with**

Press **ENTER** at the completion of each new line and continue by typing the following lines:

9:***Mr. Lang on the new project. Please**
10:***let me know if there is anything I**
11:***can do to assist you.**


Edlin displays the following:

12:*_

Insert a blank line by pressing ENTER and complete the insertion by pressing CTRL+C on the next line. You can type **1I** to see the following correctly renumbered lines:

1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6: impressed with your work.
7:
8: I think you will enjoy working with
9: Mr. Lang on the new project. Please
10: let me know if there is anything I
11: can do to assist you.
12:
13:*Sincerely,
14:
15: S.L. Martin, President

More Information About Edlin: I

 [Edlin: I--Notes](#)

 [Edlin: I](#)

**Edlin: L (List)**

Displays the block of consecutive lines you specify.

Syntax

[line1][,line2] |

Parameters

line1

Specifies the first line you want Edlin to display.

line2

Specifies the last line you want Edlin to display.

More Information About Edlin: L



[Edlin: L--Notes](#)




[Edlin: L--Example](#)





Edlin: L (List)--Notes

Default values

You can omit the *line1* parameter, the *line2* parameter, or both. The following list describes the default value(s) for each of these cases:

 If you omit only the *line1* parameter, Edlin displays up to one page (full screen of text) at a time, beginning 11 lines before the current line and ending with the line whose number is specified in *line2*. When you omit only *line1*, you must insert a comma as a placeholder.

 If you omit only the *line2* parameter, Edlin displays up to one page, beginning with the line whose number is specified in *line1*.

 If you use the Edlin I (List) command with no parameters, Edlin displays up to one page, beginning 11 lines before the current line. If you install the ANSI.SYS device driver, the number of lines Edlin displays per page depends on the type of monitor you have. This number might be greater than 24.


Blocks of more than one page

When the block of lines you specify contains more than one page, Edlin displays the first page and then prompts you with the following message:

Continue (Y/N)?

More Information About Edlin: L

 [Edlin: L--Example](#)

 [Edlin: L](#)




Edlin: L (List)--Example

To see lines 5 through 10, type the following:

5,10I

More Information About Edlin: L

 [Edlin: L--Notes](#)

 [Edlin: L](#)



Edlin: M (Move)

Moves the block of consecutive lines you specify to another location in the file in memory.

[line1],[line2],line3m

[line1],+n,line3m

Parameters

line1

Specifies the first line you want Edlin to move.

line2

Specifies the last line you want Edlin to move.

line3

Specifies the line before which you want Edlin to move the block of lines.

+n

Specifies that you want Edlin to move the block of lines that begins with the line whose number is specified in *line1* and includes the next *n* lines. If you omit the *line1* parameter, the block of lines to be moved begins with the current line.

More Information About Edlin: M



[Edlin: M--Notes](#)



[Edlin: M--Examples](#)



Edlin: M (Move)--Notes

Line renumbering

After Edlin moves lines, you can use the Edlin **l** (list) command at the Edlin prompt to see the correctly renumbered lines.

Overlapping line numbers

The line you specify for the *line3* parameter cannot be part of the block of lines to be moved. If you overlap line numbers in this way, Edlin cannot complete the move operation and displays the following message:

Entry error

For example, the following command results in an error message:

5,10,8m

More Information About Edlin: M



[Edlin: M--Examples](#)



[Edlin: M](#)



Edlin: M (Move)--Examples

Suppose that the following file is in memory and ready to edit. You can type **1L** at the Edlin prompt to see the contents of the file.

1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6: impressed with your hard work.
7:
8: I think you will enjoy working with
9: Mr. Lang on the new project. Please
10: let me know if there is anything I
11: can do to assist you.
12:
13: Sincerely,
14:
15: S.L. Martin, President
16: Rockdale Corporation
17: "A World Leader in Technology"

What if you prefer to have the motto at the beginning of the memo? You can move lines 16 and 17 before the existing line 1 by typing the following command:

16,17,1m

Type the Edlin **I** (list) command at the Edlin prompt to see the following correctly renumbered lines:

1: Rockdale Corporation
2: "A World Leader in Technology"
3: Dear Mr. Muster:
4:
5: Congratulations on your promotion
6: to the position of Senior Chemical
7: Engineer. I continue to be most
8: impressed with your hard work.
9:
10: I think you will enjoy working with
11: Mr. Lang on the new project. Please
12: let me know if there is anything I
13: can do to assist you.
14:
15: Sincerely,
16:
17: S.L. Martin, President

The following command specifies that Edlin is to move the block of lines including the current line through the next 25 lines to immediately before line 100:

,+25,100m

More Information About Edlin: M



[Edlin: M--Notes](#)



[Edlin: M](#)

**Edlin: P (Page)**

Displays all or part of a file, one page (full screen of text) at a time.

The last line displayed per screen becomes the current line.

*[line1][, line2]***p**

Parameters

line1

Specifies the first line you want Edlin to display.

line2

Specifies the last line you want Edlin to display.

More Information About Edlin: P



[Edlin: P--Notes](#)



[Edlin: P--Example](#)



Edlin: P (Page)--Notes

Omitting only the *line1* parameter

When you omit the *line1* parameter, Edlin displays a page of text that begins with the current line through *line2*.

Omitting only the *line2* parameter

When you omit the *line2* parameter, Edlin displays a page of text that begins with the line whose number you specify for *line1*.

Omitting both parameters

When you omit both parameters, Edlin displays a page of text that begins with the line after the current line.

More Information About Edlin: P



[Edlin: P--Example](#)



[Edlin: P](#)





Edlin: P (Page)--Example

To see lines 100 through 200, one page at a time, type the following command:

100,200p

More Information About Edlin: P

 [Edlin: P--Notes](#)

 [Edlin: P](#)

**Edlin: Q (Quit)**

Stops the current Edlin session without writing the edited file from memory to a disk.

When you use the **q** command, the Edlin session stops and the Windows NT prompt appears.

To specify that Edlin is to write the edited file from memory to a disk before ending the current session, you must use the **e** (end) command.

q

More Information About Edlin: Q



[Edlin: Q--Notes](#)



Edlin: Q (Quit)--Notes

A difference between the q and e commands

Suppose that the file you are editing is one that Edlin loaded into memory from a disk at the beginning of this session rather than one that you created in memory during the session. If you use the **q** command to quit the session, the contents of both the original input disk file and the .BAK version of the disk file (if one exists) remain unchanged. However, if you use the **e** command to quit the session and the file you are editing has changed during the session, the edits are saved and the original input disk file becomes the .BAK version.

Quitting Edlin without writing the edited file from memory to a disk

Use the following procedure to quit the Edlin session without writing the edited file from memory to a disk:

1 At the Edlin prompt, type **q**. Edlin displays the following message:

Abort edit (Y/N)? _

2 Press **Y** (for yes).

More Information About Edlin: Q

 Edlin: Q



Edlin: R (Replace)

Searches a block of consecutive lines for a string of one or more characters you specify, and replaces each occurrence of that string with another string you specify.

The last line in which the replacement occurs becomes the new current line.

`[line1][,line2][?]r[string1][separator string2]`

Parameters

line1

Specifies the first line in which you want Edlin to replace the string specified in *string1*.

line2

Specifies the last line in which you want Edlin to replace the string specified in *string1*.

? (question mark)

Specifies that Edlin is to prompt you by displaying a confirmation message before replacing an occurrence of the string specified in *string1*.

string1

Specifies the string that you want Edlin to replace.

separator

Separates the *string1* and *string2* values. The only valid value for this parameter is the end-of-file character (CTRL+Z).

string2

Specifies the new string that is to replace each occurrence of the string specified for *string1*.

More Information About Edlin: R



[Edlin: R--Notes](#)



[Edlin: R--Examples](#)



Edlin: R (Replace)--Notes

Command-line spacing

You must not insert a space between the **r** and any subsequent parameter on the command line.

Default settings

If you omit the *line1* parameter, Edlin begins the search on the line after the current line. If you omit the *line2* parameter, Edlin stops the search at the end of the file or at the end of the portion of text in memory.

If you omit the *string1* parameter, Edlin uses the more recently used of the following two values: the value that you specified for *string1* the last time you used the **r** command or the value that you specified for *string* the last time you used the **s** command during this session. If you omit *string1* and you have not used the **r** or **s** command yet during the editing session, the command stops.

If you omit the *string2* parameter, Edlin uses the value you specified the last time you used the **r** command during this session. If you omit the *string2* parameter and you have not used the **r** command yet during this session, Edlin deletes all occurrences of the string that is specified for *string1*.

Using the separator parameter

You must separate the *string1* and *string2* values by using the CTRL+Z key combination. Even if you omit *string1*, you need to press CTRL+Z to mark the beginning of *string2*. When you press the CTRL+Z key combination, the characters displayed are not "CTRL+Z". Instead, you see the following:

^Z

Using the question mark (?)

If you include the **?** parameter in your command, Edlin displays the line containing the first occurrence of the string specified for *string1* and prompts you by displaying the following confirmation message:


O.K.? _


If you press **Y** (for yes) or press ENTER, Edlin replaces this occurrence of the value for *string1* with the value for *string2* and searches for the next occurrence. If you press **N** (for no), Edlin does not replace this occurrence of the value for *string1* and searches for the next occurrence.

If you do not use the question mark (?)

If you do not use the **?** parameter to confirm replacements as they are made, Edlin makes all the replacements at once and then displays each line that contains a replacement. If a line contains two or more replacements, Edlin displays the line once for each replacement.

More Information About Edlin: R

 [Edlin: R--Examples](#)

 [Edlin: R](#)



Edlin: R (Replace)--Examples

Suppose you want Edlin to carry out only each confirmed replacement of the word "mine" with the word "ours" within the first 20 lines of the edited file in memory. Type the first part of the command as follows, but do not press ENTER:

1,20?rmine

To complete the command, press **CTRL+Z** (which appears on the screen as ^Z), type the word **ours**, and press **ENTER**. The complete command appears on the screen as follows:

```
1,20?rmine^Zours
```

Suppose that the following file is in memory and ready to edit. You can type **!l** at the Edlin prompt to see the contents of the file.

```
1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6: impressed with your hard work.
7:
8: I think you will enjoy working with
9: Mr. Lang on the new project. Please
10: let me know if there is anything I
11: can do to assist you.
12:
13: Sincerely,
14:
15: S.L. Martin, President
16: Rockdale Corporation
17: "A World Leader in Technology"
```

Now suppose that in lines 5 through 10 you want Edlin to replace all occurrences of the word "I" with the words "yours truly". Type the first part of the command as follows, but do not press ENTER:

5,10rI

To complete the command, press CTRL+Z (which appears on the screen as ^Z), type the words **yours truly**, and press ENTER. The complete command appears on the screen as follows:

5,10rI^Zyours truly

Because the ? parameter is omitted, Edlin replaces the three occurrences of "I" without prompting you by displaying the confirmation message. When Edlin finishes carrying out the command, it displays the following lines, which are changed as a result of the three replacements:

```
5: Engineer. yours truly continue to be most
8: yours truly think you will enjoy working with
10: let me know if there is anything yours truly
```

In the previous example, two unintended replacements occurred--in lines 5 and 8. You can avoid such changes by adding the ? parameter to the command. The completed command should appear on screen as follows:

```
5,10?rI^Zyours truly
```

Now, Edlin prompts you by displaying the confirmation message for each occurrence of the string specified in *string1* and carries out only confirmed replacements, as the following example shows:


```
5: Engineer. yours truly continue to be most
O.K.? n
8: yours truly think you will enjoy working with
O.K.? n
```


10: let me know if there is anything yours truly
O.K.? y

When the ? parameter is used, Edlin does not automatically display the lines that are changed as a result of the confirmed replacements. If you type the Edlin I (list) command at the Edlin prompt, Edlin displays the edited file that is in memory, as follows:

1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6: impressed with your hard work.
7:
8: I think you will enjoy working with
9: Mr. Lang on the new project. Please
10: let me know if there is anything yours truly
11: can do to assist you.
12:
13: Sincerely,
14:
15: S.L. Martin, President
16: Rockdale Corporation
17: "A World Leader in Technology"

More Information About Edlin: R

 [Edlin: R--Notes](#)

 [Edlin: R](#)



Edlin: S (Search)

Searches for the string of one or more characters that you specify.

Edlin displays the first line that contains an occurrence of the string. The search then stops and that line becomes the current line.

`[line1][,line2][?]s[string]`

Parameters

line1

Specifies the first line you want Edlin to search.

line2

Specifies the last line you want Edlin to search.

? (question mark)

Specifies that Edlin is to prompt you by displaying a confirmation message when it finds the first occurrence of the value you specify for *string*.

string

Specifies the string for which you want Edlin to search. You must not insert a space before this parameter on the command line, unless the space is part of the search text.

More Information About Edlin: S



[Edlin: S--Notes](#)



[Edlin: S--Examples](#)



Edlin: S (Search)--Notes

Default settings

If you omit the *line1* parameter, Edlin starts the search on the line after the current line. If you omit the *line2* parameter, Edlin stops the search at the end of the file.

If you omit the *string* parameter, Edlin uses the more recently used of the following two values: the value that you specified for *string* the last time you used the **s** command, or the value that you specified for *string1* the last time you used the **r** (replace) command during this session. If you omit the *string* parameter and this is your first use of an **s** or **r** command during this session, the **s** command stops immediately.

Using the ? (question mark)

If you include the **?** parameter in your command, Edlin displays the line containing the first occurrence of the characters specified for *string* and prompts you with the following confirmation message:

O.K.? _

If you press **Y** (for yes) or press **ENTER**, the line displayed before the message becomes the current line and the search stops. If you press **N** (for no), the search continues until another occurrence is found or until Edlin displays the following message indicating that all lines have been searched:

Not found

More Information About Edlin: S



[Edlin: S--Examples](#)



[Edlin: S](#)



Edlin: S (Search)--Examples

Suppose that the following file is in memory and ready to edit. You can type **1l** at the Edlin prompt to see the contents of the file.

1: Dear Mr. Muster:
2:
3: Congratulations on your promotion
4: to the position of Senior Chemical
5: Engineer. I continue to be most
6: impressed with your hard work.
7:
8: I think you will enjoy working with
9: Mr. Lang on the new project. Please
10: let me know if there is anything I
11: can do to assist you.
12:
13: Sincerely,
14:
15: S.L. Martin, President

To specify that Edlin is to search lines 2 through 12 for the first occurrence of the word "to", type the following command:

2,12sto

Edlin displays the following line:

4: to the position of Senior Chemical

To specify that Edlin is to display the line containing the first occurrence of "to" and then prompt you with a confirmation message, type the following command:

1,?sto

Edlin displays the following lines:

4: to the position of Senior Chemical

O.K.? _

If you press any key other than Y or ENTER, the search continues. For this example, press **N** (for no), as follows:

O.K.? **n**

Edlin continues the search and displays the following lines:

5: Engineer. I continue to be most

O.K.? _

Press **Y** to stop the search.

More Information About Edlin: S

[Edlin: S--Notes](#)

[Edlin: S](#)



Edlin: T (Transfer)

Merges the contents of another file from a disk with the contents of the file that is in memory.

[line]t[drive:][path] filename

Parameters

line

Specifies the line number before which you want Edlin to insert the file it is transferring from a disk. The default value of this parameter is the number of the current line.

[drive:][path] filename

Specifies the location and name of the file you want Edlin to insert before the line whose number is specified in the *line* parameter. The default value for *drive* is the current drive; the default value for *path* is the current directory.

More Information About Edlin: T



[Edlin: T--Note](#)




[Edlin: T--Example](#)




Edlin: T (Transfer)--Note

After Edlin merges a file from a disk, you can use the Edlin **I** (list) command at the Edlin prompt to see the correctly renumbered lines.

More Information About Edlin: T

 [Edlin: T--Example](#)

 [Edlin: T](#)





Edlin: T (Transfer)--Example

To merge a file named TAXES.MEM to line 12 of the file you are editing, type the following command:

12t taxes.mem

More Information About Edlin: T

 [Edlin: T--Note](#)

 [Edlin: T](#)



Edlin: W (Write)

Writes the first portion of the edited file from memory to a disk.

When you start Edlin, it reads as many lines as possible from your disk file into memory. If the size of your file exceeds available memory, you must edit your file in stages. That is, you edit part of the file, write that part to your disk by using the **w** command, and then load the next part from disk by using the **a** (append) command.

[n]w

Parameters

n

Specifies the number of lines that you want Edlin to write to the disk, beginning with the first line of the edited file in memory.

More Information About Edlin: W



[Edlin: W--Notes](#)



[Edlin: W--Example](#)



Edlin: W (Write)--Notes

How the W command works

When you open a file, Edlin reads lines from disk until memory is more than 75-percent full. It reserves the other 25 percent for changes you might make to the text. If your entire file fits in memory, Edlin displays the following message:

End of input file

If you see this message, you do not need to use the **w** and **a** (append) commands.

If Edlin does not display this message when you open a file, the size of the file exceeds available memory. Therefore, you must edit your file in stages by using the **w** and **a** commands to write and read parts of the file, respectively.

The **w** command does not write to disk the changes you make unless it was actually necessary to use the **w** command. Therefore, if you use the **w** command even though the whole file fit into memory and then you use the **q** (quit) command to quit Edlin, none of the changes you made to the file are saved.

Line renumbering

After Edlin writes the first portion of the edited file from memory to a disk, you can use the Edlin **l** (list) command at the Edlin prompt to see the correctly renumbered lines that remain, beginning with line number 1.

Default setting

If you omit the *n* parameter, Edlin writes lines from the edited file in memory to a disk until memory is 25-percent full.

More Information About Edlin: W



[Edlin: W--Example](#)



[Edlin: W](#)



Edlin: W (Write)--Example

Suppose the final 100 lines of your disk file do not fit into memory. After you edit the first part of the file, you can free enough space to load the remainder of your disk file into memory and continue editing by typing the following command:

125w

More Information About Edlin: W



[Edlin: W--Notes](#)



[Edlin: W](#)



Edlin Commands

Select an Edlin command to get more information.

- | | |
|--|--|
| <input type="checkbox"/> Edlin: [line] | <input type="checkbox"/> Edlin: M (Move) |
| <input type="checkbox"/> Edlin: A (Append) | <input type="checkbox"/> Edlin: P (Page) |
| <input type="checkbox"/> Edlin: C (Copy) | <input type="checkbox"/> Edlin: Q (Quit) |
| <input type="checkbox"/> Edlin: D (Delete) | <input type="checkbox"/> Edlin: R (Replace) |
| <input type="checkbox"/> Edlin: E (End) | <input type="checkbox"/> Edlin: S (Search) |
| <input type="checkbox"/> Edlin: I (Insert) | <input type="checkbox"/> Edlin: T (Transfer) |
| <input type="checkbox"/> Edlin: L (List) | <input type="checkbox"/> Edlin: W (Write) |



Net (command options)

Many Windows NT networking commands begin with the word **net**. These **net** commands have some common properties:



You can see a list of all available **net** commands by typing **net /?**.



You can get syntax help at the command line for a **net** command by typing **net help command**. For example, for help with the **net accounts** command, type **net help accounts**.



All **net** commands accept **/yes** and **/no** options (can be abbreviated **/y** and **/n**). The **/y** option automatically answers `yes` to any interactive prompt that the command generates; **/n** answers `no.` For example, **net stop server** usually prompts you to confirm that you want to stop all services that depend on the Server service; **net stop server /y** automatically answers `yes` to the prompt and the Server service shuts down.



Net Accounts

Updates the user accounts database and modifies password and logon requirements for all accounts. The Net Logon service must be running on the computer for which you want to change account parameters.

net accounts [/forcelogoff:{*minutes* | **no**}] [/minpwlen:*length*] [/maxpwage:{*days* | **unlimited**}] [/minpwage:*days*] [/uniquepw:*number*] [/domain]

net accounts [/sync] [/domain]

Parameters

none

Type **net accounts** without parameters to display the current settings for password, logon limitations, and domain information.

/forcelogoff:{*minutes* | **no**}

Sets the number of minutes to wait before ending a user's session with a server when the user account or valid logon time expires. The **no** option prevents forced logoff. The default is **no**.

When the **/forcelogoff:minutes** option is specified, Windows NT sends a warning *minutes* minutes before it forces the user off from the network. If any files are open, Windows NT warns the user. If *minutes* is less than two, Windows NT warns the user to log off from the network immediately.

/minpwlen:*length*

Sets the minimum number of characters for a user-account password. The range is 0-14 characters; the default is 6 characters.

/maxpwage::{*days* | **unlimited**}

Sets the maximum number of days that a user account's password is valid. A value of **unlimited** sets no maximum time. The **/maxpwage** option must be greater than **/minpwage**. The range is 1-49,710 days (**unlimited**); the default is 90 days.

/minpwage:*days*

Sets the minimum number of days before a user can change a new password. A value of 0 sets no minimum time. The range is 0-49,710 days; the default is 0 days.

/uniquepw:*number*

Requires that a user not repeat the same password for *number* password changes. The range is 0-8 password changes; the default is 5 password changes.

/domain

Performs the operation on the primary domain controller of the current domain. Otherwise, the operation is performed on the local computer.

This parameter applies only to Windows NT Workstation computers that are members of a Windows NT Server domain. By default, Windows NT Server computers perform operations on the primary domain controller.

/sync

When used on the primary domain controller, causes all backup domain controllers in the domain to synchronize. When used on a backup domain controller, causes that backup domain controller only to synchronize with the primary domain controller. This command applies only to computers that are members of a Windows NT Server domain.

More Information About Net Accounts



[Net Accounts--Examples](#)



Net Accounts--Examples

To display the current settings for forced logoff, the password requirements, and the server role for a server, type
net accounts

To set a minimum of seven characters for user-account passwords, type
net accounts /minpwlen:7

To specify that no password can be used more than every fifth time a password is changed, type
net accounts /uniquepw:5

To prevent users from changing passwords more often than every 7 days, to force users to change passwords every 30 days, and to force logoff after logon time expires and provide a 5-minute warning before forcing the user off, type

```
net accounts /minpwage:7 /maxpwage:30 /forcelogoff:5
```

To perform the preceding task on a Windows NT Workstation computer and ensure that the settings take effect for the Windows NT Server domain which the computer is logged on to, type

```
net accounts /minpwage:7 /maxpwage:30 /domain
```

To update the user accounts database of all member servers, type

```
net accounts /sync
```

More Information About Net Accounts



[Net Accounts](#)



Net Computer

Adds or deletes computers from a domain database. This command is available only on computers running Windows NT Server.

net computer *\\computername* {/add | /del}

Parameters

\\computername

Specifies the computer to add or delete from the domain.

/add

Adds the specified computer to the domain.

/del

Removes the specified computer from the domain.

More Information About Net Computer



[Net Computer--Notes](#)



[Net Computer--Examples](#)



Net Computer--Notes

The **net computer** command is available only on computers running Windows NT Server.

All computer additions and deletions are forwarded to the primary domain controller.

More Information About Net Computer

 [Net Computer--Examples](#)

 [Net Computer](#)




Net Computer--Examples

To add the computer GRIZZLYBEAR to the logged on domain, type

```
net computer \\grizzlybear /add
```

More Information About Net Computer

 [Net Computer--Notes](#)

 [Net Computer](#)



Net Config

Displays the configurable services that are running, or displays and changes settings for a service.

net config [*service* [*options*]]

Parameters

none

Type **net config** without parameters to display a list of configurable services.

service

Is a service (**server** or **workstation**) that can be configured with the **net config** command.

options

Are specific to the service. See **net config server** or **net config workstation** for complete syntax.

More Information About Net Config



[Net Config--Notes](#)




[Net Config--Example](#)




Net Config--Notes

Use the **net config** *service* command to change configurable Server or Workstation service parameters. The changes take effect immediately and are permanent.

See Also

 [Net Config Server](#)

 [Net Config Workstation](#)

More Information About Net Config

 [Net Config--Example](#)


 [Net Config](#)



Net Config--Example

To view which configurable services are running, type
net config

More Information About Net Config

 [Net Config--Notes](#)

 [Net Config](#)



Net Config Server

Displays or changes settings for the Server service while the service is running.

net config server [/autodisconnect:*time*] [/srvcomment:"*text* "] [/hidden:{**yes** | **no**}]

Parameters

none

Type **net config server** to display the current configuration for the Server service.

/autodisconnect:*time*

Sets the maximum number of minutes a user's session can be inactive before it is disconnected. You can specify -1 to never disconnect. The range is -1-65535 minutes; the default is 15.

/srvcomment:"*text* "

Adds a comment for the server that is displayed in Windows NT screens and with the **net view** command. The comment can have as many as 48 characters. Enclose the text in quotation marks.

/hidden:{**yes** | **no**}

Specifies whether the server's computername appears on display listings of servers. Note that hiding a server does not alter the permissions on that server. The default is **no**.

More Information About Net Config Server



[_Net Config Server--Notes](#)









[_Net Config Server--Examples](#)



Net Config Server--Notes


Use the **net config server** command to change configurable Server service parameters. The changes take effect immediately and are permanent.

Not all Server service parameters can be changed using the **net config server** command; however, the **net config server** display includes additional information. The display includes the following information about the server:

-  The server's computername, a descriptive comment, and the software version.
-  The network description.
-  The server's hidden setting.
-  The maximum number of users who can use the server's shared resources.
-  The maximum number of server files that can be open.
-  The Idle session time setting.

More Information About Net Config Server

 [Net Config Server--Examples](#)

 [Net Config Server](#)



Net Config Server--Examples

To display information about the local server and prevent the display from scrolling, type

```
net config server | more
```

To hide a server's computername from the list of available servers, type

```
net config server /hidden:yes
```

To disconnect users after 15 minutes of inactivity, type

```
net config server /autodisconnect:15
```

More Information About Net Config Server



[Net Config Server--Notes](#)



[Net Config Server](#)



Net Config Workstation

Displays or changes settings for the Workstation service while the service is running.

net config workstation [/charcount:*bytes*] [/chartime:*msec*] [/charwait:*sec*]

Parameters

none

Type **net config workstation** to display the current configuration for the local computer.

/charcount:*bytes*

Specifies the amount of data Windows NT collects before sending the data to a communication device. If **/chartime:***msec* is also set, Windows NT acts on whichever option is satisfied first. The range is 0-65535 bytes; the default is 16 bytes.

/chartime:*msec*

Sets the number of milliseconds Windows NT collects data before sending the data to a communication device. If **/charcount:***bytes* is also set, Windows NT acts on whichever option is satisfied first. The range is 0-65535000 milliseconds; the default is 250 milliseconds.

/charwait:*sec*

Sets the number of seconds Windows NT waits for a communication device to become available. The range is 0-65535 seconds; the default is 3600 seconds.

More Information About Net Config Workstation



[Net Config Workstation--Notes](#)



[Net Config Workstation--Example](#)



Net Config Workstation--Notes

Use the **net config workstation** command to change configurable Workstation service parameters. The changes take effect immediately and are permanent.

Not all Workstation service parameters can be changed using the **net config workstation** command. Other parameters can be changed in the configuration registry.

More Information About Net Config Workstation



[Net Config Workstation--Example](#)



[Net Config Workstation](#)



Net Config Workstation--Example

To display the current configuration for the Workstation service, type

```
net config workstation
```

To set the number of milliseconds Windows NT waits before sending data to a communication device to 500 milliseconds, type

```
net config workstation /chartime:500
```

More Information About Net Config Workstation



[Net Config Workstation--Notes](#)



[Net Config Workstation](#)



Net Continue

Reactivates suspended services.

net continue *service*

Parameters

service

Services that can be continued are: **file server for macintosh** (Windows NT Server only), **ftp publishing service**, **lpdsvc**, **net logon**, **network dde**, **network dde dsdm**, **nt lm security support provider**, **remoteboot** (Windows NT Server only), **remote access server**, **schedule**, **server**, **simple tcp/ip services**, and **workstation**.

More Information About Net Continue



[Net Continue--Notes](#)



[Net Continue--Example](#)



Net Continue--Notes

At a server and a client:

Use the **net continue** command to reactivate a service that has been paused. Pause a service before stopping the service to let users complete jobs or disconnect from resources. To make a minor correction to a resource, pausing the service or printer might be sufficient. Then use the **net continue** command to reactivate the service or printer, without canceling users' connections.

At a client:

Use the **net pause** and **net continue** commands to switch between network printers and a printer attached to your computer.

See Also

[Net Pause](#)

[Net Start](#)

[Net Stop](#)

More Information About Net Continue

[Net Continue--Example](#)

[Net Continue](#)



Net Continue--Example

To continue the Workstation service, type
net continue workstation

More Information About Net Continue

 [Net Continue--Notes](#)

 [Net Continue](#)



Net File

Displays the names of all open shared files on a server and the number of file locks, if any, on each file. This command also closes individual shared files and removes file locks.

net file [*id* [/close]]

Parameters

none

Type **net file** without parameters to get a list of the open files on a server.

id

Is the identification number of the file.

/close

Closes an open file and releases locked records. Type this command from the server where the file is shared.

More Information About Net File



[Net File--Notes](#)



[Net File--Examples](#)



Net File--Notes

This command can also be typed **net files**.

Use the **net file** command to view and control files shared on the network. Shared files are sometimes left open and locked by mistake. When this happens, the locked portions of a file cannot be accessed by other computers on the network. Use the **net file** command's **/close** option to remove the lock and close the file.

The display for the **net file** command is similar to the following:

File	Path	Username	#locks
0	C:\A_FILE.TXT	MARYSL	0
1	C:\DATABASE	DEBBIET	2

More Information About Net File



[Net File--Examples](#)



[Net File](#)



Net File--Examples

To view a display of information about shared files, type
net file

To close a file with identification number 1, type
net file 1 /close

More Information About Net File

 [Net File--Notes](#)

 [Net File](#)



Net Group

Adds, displays, or modifies global groups on Windows NT Server domains. This command is available for use only on Windows NT Server domains.

net group [*groupname* [/comment:"text"]] [/domain]

net group *groupname* {/add [/comment:"text"] | /delete} [/domain]

net group *groupname* *username* [...] {/add | /delete} [/domain]

Parameters

none

Type **net group** without parameters to display the name of a server and the names of groups on the server.

groupname

Is the name of the group to add, expand, or delete. Supply only a groupname to view a list of users in a group.

/comment:"text"

Adds a comment for a new or existing group. The comment can have as many as 48 characters. Enclose the text in quotation marks.

/domain

Performs the operation on the primary domain controller of the current domain. Otherwise, the operation is performed on the local computer.

This parameter applies only to Windows NT Workstation computers that are members of a Windows NT Server domain. By default, Windows NT Server computers perform operations on the primary domain controller.

username [...]

Lists one or more usernames to add to or remove from a group. Separate multiple username entries with a space.

/add

Adds a group, or adds a username to a group. An account must be established for users added to a group with this command.

/delete

Removes a group, or removes a username from a group.

More Information About Net Group



[_Net Group--Notes](#)



[_Net Group--Examples](#)



Net Group--Notes

This command can also be typed **net groups**.

Use the **net group** command to group users who use the network in the same or similar ways. When you assign rights to a group, each member of the group automatically has those rights.

A display of groups on the server is similar to the following:

```
Group Accounts for \\PRODUCTION
```

```
-----
```

```
*Domain Admins          *Domain Users
```


Notice that each groupname is preceded by an asterisk (*). The asterisk distinguishes groups in displays that include both users and groups.

See Also

 [Net Localgroup](#)

 [Net User](#)

More Information About Net Group

 [Net Group--Examples](#)

 [Net Group](#)



Net Group--Examples

To display a list of all the groups on the local server, type

```
net group
```

To add a group called *exec* to the local user accounts database, type

```
net group exec /add
```

To add a group called *exec* to a Windows NT Server domain user accounts database from a computer with Windows NT Workstation software installed, type

```
net group exec /add /domain
```

To add the existing user accounts *stevev*, *ralphr*, and *jennynt* to the *exec* group on the local computer, type

```
net group exec stevev ralphr jennynt /add
```

To add the existing user accounts *stevev*, *ralphr*, and *jennynt* to the *exec* group of a Windows NT Server domain from a computer with Windows NT Workstation software installed, type

```
net group exec stevev ralphr jennynt /add /domain
```

To display users in the *exec* group, type

```
net group exec
```

To add a comment to the *exec* group record, type

```
net group exec /comment:"The executive staff."
```

More Information About Net Group



[Net Group--Notes](#)



[Net Group](#)



Net Help

Provides a list of network commands and topics you can get help with, or provides help with a specific command or topic. The available net commands are also listed in the Commands window of this Command Reference under N.

net help [*command*]

net *command* {/help | /?}

Parameters

none

Type **net help** without parameters to display a list of commands and topics for which you can get help.

command

Is the command you need help with. Don't type **net** as part of *command*.

/help

Provides an alternate way to display the help text.

/?

Displays the correct syntax for the command.

More Information About Net Help



[Net Help--Examples](#)



Net Help--Examples

To produce the same information about the **net use** command using two forms of the **net help** command, type

net help use

or

net use /help

To view the syntax for the **net use** command, type

net use /?

More Information About Net Help



[_Net Help](#)



Net Helpmsg

Provides help with a Windows NT error message.

net helpmsg *message#*

Parameters

message#

Is the four-digit number of the Windows NT message you need help with.

More Information About Net Helpmsg



[Net Helpmsg--Notes](#)



[Net Helpmsg--Example](#)



Net Helpmsg--Notes

When a network operation fails, you see a message similar to the following:


```
NET 2182: The requested service has already been started.
```

Windows NT error messages are preceded by the word **net**, but you don't have to type **net** with *message#*.

The **net helpmsg** command explains why an error occurred and tells you what action will solve the problem.

More Information About Net Helpmsg

 [Net Helpmsg--Example](#)

 [Net Helpmsg](#)





Net Helpmsg--Example

To get help with Windows NT error message NET 2182, type
net helpmsg 2182

The error message and help information appear.

More Information About Net Helpmsg

 [Net Helpmsg--Notes](#)

 [Net Helpmsg](#)



Net Localgroup

Adds, displays, or modifies local groups.

net localgroup [*groupname* [/comment:"text"]] [/domain]

net localgroup *groupname* {/add [/comment:"text"] | /delete} [/domain]

net localgroup *groupname name* [...] {/add | /delete} [/domain]

Parameters

none

Type **net localgroup** without parameters to display the name of the server and the names of local groups on the computer.

groupname

Is the name of the local group to add, expand, or delete. Supply only a *groupname* to view a list of users or global groups in a local group.

/comment:"text"

Adds a comment for a new or existing group. The comment can have as many as 48 characters. Enclose the text in quotation marks.

/domain

Performs the operation on the primary domain controller of the current domain. Otherwise, the operation is performed on the local computer.

This parameter applies only to Windows NT Workstation computers that are members of a Windows NT Server domain. By default, Windows NT Server computers perform operations on the primary domain controller.

name [...]

Lists one or more usernames or groupnames to add or to remove from a local group. Separate multiple entries with a space. Names may be local users, users on other domains, or global groups, but not other local groups. If a user is from another domain, preface the username with the domain name (for example, SALES\RALPHR).

/add

Adds a global groupname or username to a local group. An account must first be established for users or global groups before it is added to a local group with this command.

/delete

Removes a groupname or username from a local group.

More Information About Net Localgroup



[Net Localgroup--Notes](#)



[Net Localgroup--Examples](#)




Net Localgroup--Notes

Use the **net localgroup** command to group users who use the computer or network in the same or similar ways. When you assign rights to a local group, each member of the local group automatically has those same rights.

See Also

 [Net User](#)

 [Net Group](#) (Windows NT Server only)

More Information About Net Localgroup

 [Net Localgroup--Examples](#)

 [Net Localgroup](#)



Net Localgroup--Examples

To display a list of all the local groups on the local server, type

net localgroup

To add a local group called *exec* to the local user accounts database, type

net localgroup exec /add

To add a local group called *exec* to the Windows NT Server domain user accounts database, type

net localgroup exec /add /domain

To add the existing user accounts *stevev*, *ralphr* (from the SALES domain), and *jennynt* to the *exec* local group on the local computer, type

net localgroup exec stevev sales\ralphr jennynt /add

To add the existing user accounts *stevev*, *ralphr*, and *jennynt* to the *exec* group of a Windows NT Server domain, type

net localgroup exec stevev ralphr jennynt /add /domain

To display users in the *exec* local group, type

net localgroup exec

To add a comment to the *exec* local group record, type

net localgroup exec /comment:"The executive staff."

More Information About Net Localgroup



[Net Localgroup--Notes](#)



[Net Localgroup](#)



Net Name

Adds or deletes a messaging name (sometimes called an alias), or displays the list of names the computer will accept messages for. The Messenger service must be running to use **net name**.

net name [*name* [/add | /delete]]

Parameters

none

Type **net name** without parameters to display a list of names currently in use.

name

Specifies the name to receive messages. The name can have as many as 15 characters.

/add

Adds a name to a computer. Typing **/add** is optional; typing **net name name** works the same way as typing **net name name /add** .

/delete

Removes a name from a computer.

More Information About Net Name



[_Net Name--Notes](#)



[_Net Name--Examples](#)



Net Name--Notes

Use the **net name** command to specify a name for receiving messages. The Messenger service must be started to use the **net name** command. Each messaging name must be unique on the network. Names created with **net name** are strictly for messaging; names are not groups.

Windows NT uses three kinds of names:



Any name for messaging, which is added with **net name**.



The computer's computername, which is added when the Workstation service starts.



Your username, which is added when you log on, provided your user name is not in use as a message name elsewhere on the network.

The computername cannot be deleted. The username can be deleted.

See Also



[Net Send](#)

More Information About Net Name



[Net Name--Examples](#)



[Net Name](#)



Net Name--Examples

To view the list of names at your computer, type

net name

To add the name *rsvp* to your computer, type

net name rsvp

To remove the name *rsvp* from your computer, type

net name rsvp /delete

More Information About Net Name



[Net Name--Notes](#)



[Net Name](#)



Net Pause

Pauses running services.

net pause *service*

Parameters

service

Is **file server for macintosh** (Windows NT Server only), **ftp publishing service**, **lpdsvc**, **net logon**, **network dde**, **network dde dsdm**, **nt lm security support provider**, **remoteboot** (Windows NT Server only), **remote access server**, **schedule**, **server**, **simple tcp/ip services**, or **workstation**.

More Information About Net Pause



[Net Pause--Notes](#)



[Net Pause--Example](#)



Net Pause--Notes

At a server:

Use the **net pause** command before stopping a service to let users complete jobs or disconnect from resources. Pausing a service puts it on hold but doesn't remove the software from memory. Users who already have a connection to the resource are able to finish their tasks, but new connections to the resource are prevented.

If you plan to stop a service that affects shared resources, first pause the service. Then, using the **net send** command, broadcast a message that the service will be stopped. After sufficient time for people to finish using the resource, stop the service using the **net stop** command.

To reactivate a service that has been paused, use the **net continue** command.

At a client:

Use the **net pause** and **net continue** commands to switch between network printers and printers attached to your workstation.

At both a server and a client:

Note



Not all services can be paused.

Pausing affects the Windows NT services in the following ways:



Pausing the Net Logon service prevents the computer from processing logon requests. If the domain has other logon servers, users can still log on to the network.



Pausing the Server service prevents users from making new connections to the server's shared resources and, if there are no other logon servers on the network, from logging on to the network. An existing connection is unaffected. Administrators can make connections to the server even if it is paused.



Pausing the Workstation service keeps the username, password, and connections defined, but directs requests for print devices to printers attached to the computer rather than to printers connected to the network.

See Also



[Net Continue](#)



[Net Start](#)



[Net Stop](#)



[Net Use](#)

More Information About Net Pause



[Net Pause--Example](#)



[Net Pause](#)



Net Pause--Example

To pause the Server service, type

net pause server

To pause the Net Logon service, type

net pause "net logon"

More Information About Net Pause

 [Net Pause--Notes](#)

 [Net Pause](#)



Net Print

Displays or controls print jobs and printer queues.

net print *\\computername\sharename*

net print [*\\computername*] *job#* [**/hold** | **/release** | **/delete**]

Parameters

computername

Is the name of the computer sharing the printer queue(s).

sharename

Is the name of the printer queue. When including the *sharename* with the *computername*, use a backslash (\) to separate the names.

job#

Is the identification number assigned to a print job in a printer queue. A computer with one or more printer queues assigns each print job a unique number. If a job number is being used in one printer queue shared by a computer, that number is not assigned to any other job on that computer, not even to jobs in other printer queues on that computer.

/hold

When used with *job#*, holds a print job waiting in the printer queue. The print job stays in the printer queue, and other print jobs bypass it until it is released.

/release

Releases a print job that has been held.

/delete

Removes a print job from a printer queue.

More Information About Net Print



[Net Print--Notes](#)



[Net Print--Examples](#)



Net Print--Notes

The **net print** command displays information about printer queues in several ways.

You can display a particular queue by using **net print \\computername\sharename**. The following is a sample display of a print queue:

Printer queues at \\PRODUCTION

Name	Job #	Size	Status

LASER Queue	1 jobs		*Queue Active*
ERNESTA	84	0	Spooling

Use **net print job#** to display a single print job. A display similar to the following appears:

```
Job #           35
Status          Waiting
Size            3096
Remark
Submitting user MARYSL
Notify          MARYSL
Job data type
Job parameters
Additional info
```

More Information About Net Print



[Net Print--Examples](#)



[Net Print](#)



Net Print--Examples

To get information about job number 35 on the \\PRODUCTION computer, type

net print \\production 35

To hold job number 263 on the \\PRODUCTION computer, type

net print \\production 263 /hold

To release job number 263 on the \\PRODUCTION computer, type

net print \\production 263 /release

To list the contents of the DOTMATRIX printer queue on the \\PRODUCTION computer, type

net print \\production\dotmatrix

More Information About Net Print



[Net Print--Notes](#)



[Net Print](#)



Net Send

Sends messages to other users, computers, or messaging names on the network. The Messenger service must be running to receive messages.

net send {*name* | * | **/domain[:name]** | **/users**} *message*

Parameters

name

Is the username, computername, or messaging name to send the message to. If the name is a computername that contains blank characters, enclose the alias in quotation marks (" ").

*

Sends the message to all the names in your group.

/domain[:name]

Sends the message to all the names in the computer's domain. If *name* is specified, the message is sent to all the names in the specified domain or workgroup.

/users

Sends the message to all users connected to the server.

message

Is text to be sent as a message.

More Information About Net Send



[_Net_Send--Notes](#)



[_Net_Send--Examples](#)



Net Send--Notes

You can send a message only to a name that is active on the network. If the message is sent to a username, that user must be logged on and running the Messenger service to receive the message.

Sending messages to multiple users

Windows NT provides several ways to *broadcast* a message. You can broadcast a message to all the names in your computer's domain (use * or **/domain**), or a different domain (**/domain:domainname**). Broadcast messages can have as many as 128 characters.

The **/users** option lets you send a message to all users who have sessions with the server. Parameters that send to multiple users should be used with discretion.

See Also



[Net Name](#)



[Net Start Messenger](#)

More Information About Net Send



[Net Send--Examples](#)



[Net Send](#)



Net Send--Examples

To send the message "Meeting changed to 3 p.m. Same place." to the user robertf, type

net send robertf Meeting changed to 3 p.m. Same place.

To send a message to all users connected to the server, type

net send /users This server will shut down in 5 minutes.

To send a message that includes a slash mark, type

net send robertf "Format your disk with FORMAT /4"

More Information About Net Send



[Net Send--Notes](#)



[Net Send](#)



Net Session

Lists or disconnects the sessions between a local computer and the clients connected to it.

net session [*\\computername*] [**/delete**]

Parameters

none

Type **net session** without parameters to display information about all sessions with the local computer.

\\computername

Identifies the computer for which to list or disconnect sessions.

/delete

Ends the computer's session with *\\computername* and closes all open files on the computer for the session. If *\\computername* is omitted, all sessions on the local computer are canceled.

More Information About Net Session



[Net Session--Notes](#)



[Net Session--Examples](#)



Net Session--Notes

The **net session** command can also be typed **net sessions** or **net sess**.

Use the **net session** command to display the computernames and usernames of users accessing a server, whether they have files open, and how long each user's session has been idle.

The display is similar to the following:

Computer	User name	Client type	Opens	Idle time
\\BASSETT	CHRISDR	NT	1	00:00:13
\\SHARONCA	Administrator	DOS LM 2.1	0	01:05:13

To display one user's session, include `\\computername` with the command. A single user's display includes a list of shared resources to which the user has connections.

A session is recorded when a user at a client successfully contacts a server. A successful session occurs when the two systems are on the same network, and the user has a username and password that are accepted by the server. A user at a client has to have a session with a server before being able to use the server's resources, and a session is not established until a user at a client connects to a resource. A client and a server have only one session, but they can have many entry points, or connections, to resources.

To set how long a session can remain idle before being automatically disconnected, set the **autodisconnect** feature using the **/autodisconnect** option of the **net config server** command. An automatic disconnection is transparent to the user because Windows NT automatically reconnects the session when the user uses the resource again.

To end a session with the server, use the **/delete** option along with `\\computername`.

More Information About Net Session



[Net Session--Examples](#)



[Net Session](#)



Net Session--Examples

To display a listing of session information for the local server, type

net session

To display session information for the client with the computername SHEPHERD, type

net session \\shepherd

To end all sessions between the server and the clients connected to it, type

net session /delete

More Information About Net Session



[Net Session--Notes](#)



[Net Session](#)



Net Share

Creates, deletes, or displays shared resources.

net share *sharename*

net share *sharename=drive:path* [/users:*number* | /unlimited] [/remark:"*text*"]

net share *sharename* [/users:*number* | unlimited] [/remark:"*text*"]

net share {*sharename* | *drive:path*} /delete

Parameters

none

Type **net share** without parameters to display information about all resources being shared on the local computer.

sharename

Is the network name of the shared resource. Type **net share** with a *sharename* only to display information about that share.

drive:path

Specifies the absolute path of the directory to be shared.

/users:*number*

Sets the maximum number of users who can simultaneously access the shared resource.

/unlimited

Specifies an unlimited number of users who can simultaneously access the shared resource.

/remark:"*text*"

Adds a descriptive comment about the resource. Enclose the text in quotation marks.

/delete

Stops sharing the resource.

More Information About Net Share

 [Net Share--Notes](#)

 [Net Share--Examples](#)



Net Share--Notes

Use the **net share** command to share resources.

To share a directory with a path that contains a blank character, enclose the drive and the path of the directory in quotation marks (" ").

When you display all the shared resources on a computer, Windows NT reports the sharename of the resource, the devicename(s) or path associated with the resource, and a descriptive comment about the resource.

The display is similar to the following:

Sharename	Resource	Remark
ADMIN\$	C:\WINNT	Remote Admin
C\$	C:\	Default Share for Internal Use
print\$	C:\WINNT\SYSTEM\SPOOL	
IPC\$		Remote IPC
LASER	LPT1	Spooled Laser printer

As you create shares on a server, they are saved. When you stop the Server service, all shares are disconnected, but they are reconnected automatically the next time the Server service is started or the computer is restarted.

More Information About Net Share



[Net Share--Examples](#)



[Net Share](#)



Net Share--Examples

To display information about shared resources on the computer, type

net share

To share a computer's C:\LETTERS directory with the sharename SECRETARY and include a remark, type

net share secretary=c:\letters /remark:"For department 123."

To stop sharing the LETTERS directory, type

net share secretary /delete

To share a computer's C:\ART LST directory with the sharename LIST, type

net share list="c:\art lst"

More Information About Net Share



[Net Share--Notes](#)



[Net Share](#)



Net Start

Starts a service, or displays a list of started services. Service names of two or more words, such as Net Logon or Computer Browser, must be enclosed in quotation marks (").

net start [*service*]

Parameters

none

Type **net start** without parameters to display a list of running services.

service

Includes **alerter, client service for netware, clipbook server, computer browser, dhcp client, directory replicator, eventlog, ftp publishing service, lpdsvc, messenger, net logon, network dde, network dde dsdm, network monitor agent, nt lm security support provider, ole, remote access connection manager, remote access isnsap service, remote access server, remote procedure call (rpc) locator, remote procedure call (rpc) service, schedule, server, simple tcp/ip services, snmp, spooler, tcp/ip netbios helper, ups, and workstation.**

These services are available only on Windows NT Server: **file server for macintosh, gateway service for netware, microsoft dhcp server, print server for macintosh, remoteboot, windows internet name service.**

More Information About Net Start



[_Net Start--Notes](#)



[_Net Start--Example](#)



Net Start--Notes

Use the **net start** *service* command to start Windows NT services. Some services are dependent on other services.


You can also use the Services option in Control Panel to configure services to stop and start automatically. This Control Panel option also allows you to manually stop, start, pause, and continue network services.

Service names with two words, such as Net Logon or Computer Browser, must be enclosed in quotation marks ("").

This command will also start network services that are not provided with Windows NT.

More Information About Net Start

 [Net Start--Example](#)

 [Net Start](#)



Net Start--Example

To list the services that are currently running, type

```
net start
```

More Information About Net Start

 [Net Start--Notes](#)

 [Net Start](#)



Net Start Alerter

Starts the Alerter service. The Alerter service sends alert messages.

net start alerter

More Information About Net Start Alerter

 [Net Start Alerter--Notes](#)

 [Net Start Alerter--Example](#)



Net Start Alerter--Notes

Start the Alerter service to send alert messages to specified users and to users connected to the server. Alert messages warn about security and access problems and user session problems.

Use Server Manager to specify the administrators who should receive administrative alerts.

Alert messages are sent as Windows NT messages from the server to a user's computer. The Messenger service must be running on the user's computer for the user to receive the alert messages.

More Information About Net Start Alerter



[Net Start Alerter--Example](#)



[Net Start Alerter](#)



Net Start Alerter--Example

To start the Alerter service, type

net start alerter

More Information About Net Start Alerter

 [Net Start Alerter--Notes](#)

 [Net Start Alerter](#)



Net Start Client Service for NetWare

Starts the Client Service for NetWare service. This command is available only on Windows NT Workstation if Client Service for NetWare has been installed.

net start "client service for netware"



Net Start ClipBook Server

Starts the ClipBook Server service. Service names with two words, such as ClipBook Server, must be enclosed in quotation marks (").

net start "clipbook server"

More Information About Net Start ClipBook Server



[Net Start ClipBook Server--Notes](#)




[Net Start ClipBook Server--Example](#)




Net Start ClipBook Server--Notes

The ClipBook Server permits cutting and pasting over the network.

More Information About Net Start ClipBook Server

 [Net Start ClipBook Server--Example](#)

 [Net Start ClipBook Server](#)



Net Start ClipBook Server--Example

To start the ClipBook Server service, type

net start "clipbook server"

More Information About Net Start ClipBook Server



[Net Start ClipBook Server--Notes](#)



[Net Start ClipBook Server](#)



Net Start Computer Browser

Starts the Computer Browser service.

net start "computer browser"

More Information About Net Start Computer Browser



[Net Start Computer Browser--Notes](#)



[Net Start Computer Browser--Example](#)



Net Start Computer Browser--Notes

The Computer Browser service supports browsing computers on the network and being browsed by other computers on the network. Service names with two words, such as Computer Browser, must be enclosed in quotation marks ("). This service can also be started with the command **net start browser**.

More Information About Net Start Computer Browser



[Net Start Computer Browser--Example](#)



[Net Start Computer Browser](#)




Net Start Computer Browser--Example

To start the Computer Browser service, type

net start "computer browser"

More Information About Net Start Computer Browser

 [Net Start Computer Browser--Notes](#)

 [Net Start Computer Browser](#)



Net Start DHCP Client

Starts the DHCP Client service. This command is available only if the TCP/IP protocol has been installed.

net start "dhcp client"

More Information About Net Start DHCP Client

 [Net Start DHCP Client--Notes](#)

 [Net Start DHCP Client--Example](#)



Net Start DHCP Client--Notes

The DHCP Client service supports obtaining an IP address from a DHCP server.

Service names with two words must be enclosed in quotation marks ("). This service can also be started with the command **net start dhcp**

You cannot stop or pause the DHCP Client service.

More Information About Net Start DHCP Client



[Net Start DHCP Client--Example](#)



[Net Start DHCP Client](#)



Net Start DHCP Client--Example

To start the DHCP Client service, type

net start "dhcp client"

More Information About Net Start DHCP Client

 [Net Start DHCP Client--Notes](#)

 [Net Start DHCP Client](#)



Net Start Directory Replicator

Starts the Directory Replicator service. The Directory Replicator service copies designated files to specified servers. Service names with two words, such as Directory Replicator, must be enclosed in quotation marks (""). This service can also be started with the command **net start replicator**.

net start "directory replicator"

More Information About Net Start Directory Replicator



[Net Start Directory Replicator--Notes](#)



[Net Start Directory Replicator--Example](#)



Net Start Directory Replicator--Notes

Start the Directory Replicator service to ensure that designated files are exactly the same on all specified servers. The Directory Replicator must be configured prior to starting the Directory Replicator service.

Service names with two words, such as Directory Replicator or Computer Browser, must be enclosed in quotation marks (""). This service can also be started with the command **net start replicator**.

More Information About Net Start Directory Replicator



[Net Start Directory Replicator--Example](#)



[Net Start Directory Replicator](#)



Net Start Directory Replicator--Example

To start the Directory Replicator service, type

net start "directory replicator"

More Information About Net Start Directory Replicator

 [Net Start Directory Replicator--Notes](#)

 [Net Start Directory Replicator](#)



Net Start Eventlog

Starts the event logging service, which logs events on the local computer. This service must be started prior to using the Event Viewer to view the logged events.

net start eventlog

More Information About Net Start Eventlog



[Net Start Eventlog--Notes](#)



[Net Start Eventlog--Example](#)



Net Start Eventlog--Notes

The Eventlog service logs any significant system, security, and application occurrences that require users to be notified.

For more information on Event Viewer and configuring log maintenance information, see online Help for Event Viewer.

You cannot stop or pause the Eventlog service.

More Information About Net Start Eventlog



[Net Start Eventlog--Example](#)



[Net Start Eventlog](#)




Net Start Eventlog--Example

To start the Eventlog service, type

net start eventlog

More Information About Net Start Eventlog

 [Net Start Eventlog--Notes](#)

 [Net Start Eventlog](#)



Net Start File Server for Macintosh

Starts the File Server for Macintosh service, permitting the sharing of files with Macintosh computers. This command is available only on computers running Windows NT Server.

net start "file server for macintosh"



Net Start FTP Publishing Service

Starts the FTP publishing service. This command is available only if Internet Information Server has been installed.

net start "ftp publishing service"



Net Start Gateway Service for NetWare

Starts the Gateway Service for NetWare service. This command is available only on Windows NT Server if Gateway Service for NetWare has been installed.

net start "gateway service for netware"





Net Start Lpdsvc

Starts the Lpdsvc service. This command is available only if the TCP/IP protocol has been installed.

net start lpdsvc

More Information About Net Start Lpdsvc

 [Net Start Lpdsvc--Notes](#)

 [Net Start Lpdsvc--Example](#)



Net Start Lpdsvc--Notes

The LPDSVC service is the server side of TCP/IP printing for UNIX clients. If any UNIX clients on the network want to print to a printer connected to a Windows NT computer, this service needs to be running on the Windows NT computer so it can accept requests from the UNIX clients. The LPDSVC service supports any print format, including plain-text. It does not perform any additional processing.

The LPDSVC service is independent of the LPRMON service, which runs automatically to allow a Windows NT computer (and all clients who can access this computer) to print to a printer connected to a UNIX system.

More Information About Net Start Lpdsvc



[Net Start Lpdsvc--Example](#)



[Net Start Lpdsvc](#)



Net Start Lpdsvc--Example

To start the Lpdsvc service, type

net start lpdsvc

Or, in **Control Panel**, choose the **Services** option. Then select **LPDSVC** in the Service list and choose the **Start** button.

On the UNIX computer, you can use the Windows NT printer by creating a remote printer. Creating an NT remote printer is no different from creating a UNIX remote printer (such as creating a printcap entry on BSD systems, etc.)

For LPR to print to an NT network share with a UNC name (such as \\ntprintserver\hplaser), the share needs to be designated as a null share. Any share can be designated as a null share by adding it to the NullSessionShares list under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters key in the Registry. The computer must be restarted for this change to become effective.

Currently, LPR cannot print to network shares on an OS/2 LAN Manager server.

More Information About Net Start Lpdsvc



[Net Start Lpdsvc--Notes](#)



[Net Start Lpdsvc](#)




Net Start Messenger

Starts the Messenger service. The Messenger service enables a computer to receive messages.

net start messenger

See Also

 [Net Send](#)

 [Net Name](#)

More Information About Net Start Messenger

 [Net Start Messenger--Notes](#)

 [Net Start Messenger--Example](#)



Net Start Messenger--Notes

Start the Messenger service so that a computer can receive messages. A message is sent to a computer using the names assigned to the computer as identification.

See Also

 [Net Send](#)

 [Net Name](#)

More Information About Net Start Messenger

 [Net Start Messenger--Example](#)

 [Net Start Messenger](#)



Net Start Messenger--Example

To start the Messenger service, type

net start messenger

More Information About Net Start Messenger

 [Net Start Messenger--Notes](#)

 [Net Start Messenger](#)



Net Start Microsoft DHCP Server

Starts the Microsoft DHCP Server service. This command is available only on computers running Windows NT Server and if the TCP/IP protocol and DHCP server have been installed.

net start "microsoft dhcp server"

More Information About Net Start Microsoft DHCP Server



[Net Start Microsoft DHCP Server--Notes](#)



[Net Start Microsoft DHCP Server--Example](#)



Net Start Microsoft DHCP Server--Notes

The Microsoft DHCP Server service supplies network clients with IP addresses.

Service names with two words must be enclosed in quotation marks ("). This service can also be started with the command **net start dhcpserver**.

More Information About Net Start Microsoft DHCP Server



[Net Start Microsoft DHCP Server--Example](#)



[Net Start Microsoft DHCP Server](#)



Net Start Microsoft DHCP Server--Example

To start the Microsoft DHCP Server service, type

net start "microsoft dhcp server"

More Information About Net Start Microsoft DHCP Server



[Net Start Microsoft DHCP Server--Notes](#)



[Net Start Microsoft DHCP Server](#)



Net Start Net Logon

Starts the Net Logon service. The Net Logon service verifies logon requests and controls replication of the user accounts database domainwide. Service names with two words, such as Net Logon, must be enclosed in quotation marks ("). This service can also be started with the command **net start netlogon**.

net start "net logon"

More Information About Net Start Net Logon



[Net Start Netlogon--Notes](#)



[Net Start Netlogon--Example](#)



Net Start Net Logon--Notes

Start the Net Logon service on all the servers in a domain that use a copy of the domain's user accounts database.

When starting services at the command prompt, service names with two words, such as Net Logon, must be enclosed in quotation marks (""). This service can also be started by typing **net start netlogon**.

Primary Domain Controllers in Windows NT and LAN Manager 2.x Mixed Domains

If you are using Windows NT Server domains, the primary domain controller must be a Windows NT Server computer. Backup domain controllers in the domain can be either Windows NT Servers or LAN Manager 2.x computers.

All Windows NT computers must log on to a Windows NT Server primary domain controller. LAN Manager 2.x computers can log on to LAN Manager 2.x or Windows NT Server primary domain controllers.

More Information About Net Start Net Logon



[Net Start Netlogon--Example](#)



[Net Start Netlogon](#)



Net Start Net Logon--Example

To start the Net Logon service, type

net start "net logon"

More Information About Net Start Net Logon

 [Net Start Netlogon--Notes](#)

 [Net Start Netlogon](#)



Net Start Network DDE

Starts the Network DDE service.

net start "network dde"



Net Start Network DDE DSDM

Starts the Network DDE server service.

net start "network dde dsdm"



Net Start Network Monitor Agent

Starts the Network Monitor Agent service. This command is available only if the Network Monitor Agent has been installed.

net start "network monitor agent"

More Information About Net Start Network Monitor Agent



[Net Start Network Monitor Agent--Notes](#)



[Net Start Network Monitor Agent--Example](#)




Net Start Network Monitor Agent--Notes

The Network Monitor Agent service supports remote monitoring of a client's network communication.

Service names with two words must be enclosed in quotation marks (").

More Information About Net Start Network Monitor Agent

 [Net Start Network Monitor Agent--Example](#)

 [Net Start Network Monitor Agent](#)




Net Start Network Monitor Agent--Example

To start the Network Monitor Agent service, type

net start "network monitor agent"

More Information About Net Start Network Monitor Agent

 [Net Start Network Monitor Agent--Notes](#)

 [Net Start Network Monitor Agent](#)



Net Start NT LM Security Support Provider

Starts the NT LM Security Support Provider service. This command is available only if the NT LM Security Support Provider has been installed.

net start "nt lm security support provider"



Net Start Ole

Starts the object linking and embedding service.

net start ole



Net Start Print Server for Macintosh

Starts the Print Server for Macintosh service, permitting printing from Macintosh computers. This command is available only on computers running Windows NT Server.

net start "print server for macintosh"



Net Start Remoteboot

Starts the Remoteboot service, permitting computers on the network to load the operating system from the computer. This command is available only on computers running Windows NT Server.

net start remoteboot



Net Start Remote Access Connection Manager

Starts the Remote Access Connection Manager service. This command is available only if the Remote Access Service has been installed.

net start "remote access connection manager"



Net Start Remote Access ISNSAP Service

Starts the Remote Access ISNSAP Service service. This command is available only if the Remote Access Service has been installed.

net start "remote access isnsap service"



Net Start Remote Access Server

Starts the Remote Access Server service. This command is available only if the Remote Access Service has been installed.

net start "remote access server"





Net Start Remote Procedure Call (RPC) Locator

Starts the RPC Locator service. The Locator service is the RPC name service for Microsoft Windows NT.

net start "remote procedure call (RPC) locator"

More Information About Net Start Remote Procedure Call (RPC) Locator

 [Net Start Remote Procedure Call \(RPC\) Locator--Notes](#)

 [Net Start Remote Procedure Call \(RPC\) Locator--Examples](#)




Net Start Remote Procedure Call (RPC) Locator--Notes

Start the RPC Locator service which allow distributed applications to use the Microsoft RPC name service. The RPC Locator manages the RPC name service database.

The server side of the distributed application registers its availability with the RPC Locator service. The client side of the distributed application queries the Locator service to find available compatible server applications.

Service names with two words must be enclosed in quotation marks ("). This service can also be started with the command **net start rpclocator**.

More Information About Net Start Locator

 [Net Start Remote Procedure Call \(RPC\) Locator--Examples](#)

 [Net Start Remote Procedure Call \(RPC\) Locator](#)





Net Start Remote Procedure Call (RPC) Locator--Example

To start the RPC Locator service, type

net start "remote procedure call (rpc) locator"

More Information About Net Start Remote Procedure Call (RPC) Locator

 [Net Start Remote Procedure Call \(RPC\) Locator--Notes](#)

 [Net Start Remote Procedure Call \(RPC\) Locator](#)



Net Start Remote Procedure Call (RPC) Service

Starts the Remote Procedure Call (RPC) Service service. The Remote Procedure Call (RPC) Service service is the RPC subsystem for Microsoft Windows NT. The RPC subsystem includes the endpoint mapper and other miscellaneous RPC services.

net start "remote procedure call (rpc) service"

More Information About Net Start Remote Procedure Call (RPC) Service



[Net Start Remote Procedure Call \(RPC\) Service--Notes](#)



[Net Start Remote Procedure Call \(RPC\) Service--Examples](#)



Net Start Remote Procedure Call (RPC) Service--Notes

Start the Remote Procedure Call (RPC) Service service to allow distributed applications to use dynamic endpoints. The Remote Procedure Call (RPC) Service service manages the endpoint map database.

The server side of the distributed application registers its endpoints with the Remote Procedure Call (RPC) Service service. The client run-time library, on behalf of the client side of the distributed application, queries the Remote Procedure Call (RPC) Service service to obtain the endpoint information.

To determine whether a distributed application uses the endpoint mapper service, see the documentation for that distributed application.

Service names with two words must be enclosed in quotation marks ("). This service can also be started with the command **net start rpcss**.

More Information About Net Start Remote Procedure Call (RPC) Service



[Net Start Remote Procedure Call \(RPC\) Service--Examples](#)




[Net Start Remote Procedure Call \(RPC\) Service](#)




Net Start Remote Procedure Call (RPC) Service--Example

To start the Remote Procedure Call (RPC) Service service, type
net start "remote procedure call (rpc) service"

More Information About Net Start Remote Procedure Call (RPC) Service

 [Net Start Remote Procedure Call \(RPC\) Service--Notes](#)

 [Net Start Remote Procedure Call \(RPC\) Service](#)



Net Start Schedule

Starts the Schedule service. The Schedule service enables a computer to start programs at a specified time with the **at** command.

net start schedule

More Information About Net Start Schedule



[Net Start Schedule--Notes](#)



[Net Start Schedule--Example](#)



Net Start Schedule--Notes

The Schedule service must be started to use the **at** command. Scheduled commands may require other services be running for the command to successfully run.

The Schedule service is initially configured to run in the System account on the local computer. When the Schedule service runs using this account, there are no restrictions on the jobs that can be executed by Schedule service. However, these jobs will have limited network access, because the System account on a local computer is not recognized by other computers.

To overcome network access limitations, you can configure the Schedule service to run in a user's account. Then, jobs executed by the Schedule service will be governed by network access of the user's account. However, since in this case Schedule service is not using the local System account, only jobs that do not require presence of a window can be run successfully.

To configure the Schedule service to run in a user's account, use User Manager to give this user's account the "Log on as a service" right on a local computer, then in the **Control Panel Services** option, configure the Schedule service to **startup using this user's account**.

See Also



[At](#)

More Information About Net Start Schedule



[Net Start Schedule--Example](#)



[Net Start Schedule](#)



Net Start Schedule--Example

To start the Schedule service, type

net start schedule

More Information About Net Start Schedule



[Net Start Schedule--Notes](#)



[Net Start Schedule](#)



Net Start Server

Starts the Server service. The Server service enables a computer to share resources on the network.

net start server

More Information About Net Start Server

 [Net Start Server--Notes](#)

 [Net Start Server--Example](#)



Net Start Server--Notes

Start the Server service to share a server's resources with users on the network.

More Information About Net Start Server

 [Net Start Server--Example](#)

 [Net Start Server](#)



Net Start Server--Example

To start the Server service, type

net start server

More Information About Net Start Server

 [Net Start Server--Notes](#)

 [Net Start Server](#)



Net Start Simple TCP/IP Services

Starts the Simple TCP/IP Services service. This command is available only if TCP/IP and the Simple TCP/IP Services have been installed.

net start "simple tcp/ip services"

More Information About Net Start Simple TCP/IP Services



[Net Start Simple TCP/IP Services--Notes](#)



[Net Start Simple TCP/IP Services--Example](#)



Net Start Simple TCP/IP Services--Notes

The Simple TCP/IP Services service supports the TCP/IP services Character Generator, Daytime, Discard, Echo, and Quote of the Day. When started, the computer can respond to requests from other computers that support these protocols.

Service names with two words must be enclosed in quotation marks (")..

More Information About Net Start Simple TCP/IP Services



[Net Start Simple TCP/IP Services--Example](#)



[Net Start Simple TCP/IP Services](#)



Net Start Simple TCP/IP Services--Example

To start the Simple TCP/IP Services service, type

net start "simple tcp/ip services"

More Information About Net Start Simple TCP/IP Services



[Net Start Simple TCP/IP Services--Notes](#)



[Net Start Simple TCP/IP Services](#)



Net Start SNMP

Starts the SNMP service. The SNMP service allows a server to report its current status to a SNMP management system on a TCP/IP network. This command is available only if TCP/IP and SNMP have been installed.

net start snmp

More Information About Net Start SNMP



[Net Start SNMP--Example](#)



Net Start SNMP--Example

To start the SNMP service, type

net start snmp

More Information About Net Start SNMP



[Net Start SNMP](#)



Net Start Spooler

Starts the Spooler service.

net start spooler



Net Start TCP/IP NetBIOS Helper

Enables the Netbios over TCP service. This command is available only if TCP/IP has been installed.

net start "tcp/ip netbios helper"




Net Start UPS

Starts the Uninterruptible Power Supply service.

net start ups

More Information About Net Start UPS

 [Net Start UPS--Notes](#)

 [Net Start UPS--Example](#)



Net Start UPS--Notes

Use the UPS option in Control Panel to configure the UPS service.

UPS service can be configured by using UPS option in Control Panel. Note that if UPS service is configured to execute a command at shutdown time, the chosen command ought to complete in 30 seconds. Otherwise, safe shutdown of a computer may be jeopardized.

More Information About Net Start UPS



[Net Start UPS](#)



[Net Start UPS--Example](#)



Net Start UPS--Example

To start the UPS service, type

net start ups

More Information About Net Start UPS

 [Net Start UPS--Notes](#)

 [Net Start UPS](#)



Net Start Windows Internet Name Service

Starts the Windows Internet Name Service service. This command is available only on Windows NT Servers if TCP/IP and the Windows Internet Name Service have been installed.

net start "windows internet name service"

More Information About Net Start Windows Internet Name Service



[Net Start Windows Internet Name Service--Notes](#)



[Net Start Windows Internet Name Service--Example](#)



Net Start Windows Internet Name Service--Notes

The Windows Internet Name Service service maps computer names to TCP/IP addresses for client computers on the network.

Service names with two words must be enclosed in quotation marks ("). This service can also be started with the command **net start wins**.

More Information About Net Start Windows Internet Name Service



[Net Start Windows Internet Name Service--Example](#)



[Net Start Windows Internet Name Service](#)




Net Start Windows Internet Name Service--Example

To start the Windows Internet Name Service service, type
net start "windows internet name service"

More Information About Net Start Windows Internet Name Service

 [Net Start Windows Internet Name Service--Notes](#)

 [Net Start Windows Internet Name Service](#)



Net Start Workstation

Starts the Workstation service. The Workstation service enables a computer to connect to and use network resources.

net start workstation

More Information About Net Start Workstation



[Net Start Workstation--Notes](#)



[Net Start Workstation--Example](#)



Net Start Workstation--Notes

Start the Workstation service so that your computer can use information and equipment on the network.

More Information About Net Start Workstation

 [Net Start Workstation--Example](#)

 [Net Start Workstation](#)



Net Start Workstation--Example

To start the Workstation service, type

net start workstation

More Information About Net Start Workstation

 [Net Start Workstation--Notes](#)

 [Net Start Workstation](#)



Net Statistics

Displays the statistics log for the local Workstation or Server service.

net statistics [**workstation** | **server**]

Parameters

none

Type **net statistics** without parameters to list the running services for which statistics are available.

workstation

Displays statistics for the local Workstation service.

server

Displays statistics for the local Server service.

More Information About Net Statistics



[Net Statistics--Notes](#)



[Net Statistics--Examples](#)








Net Statistics--Notes

This command can also be typed **net stats**.

Use the **net statistics** command to display performance information for the specified service.






The Server service:

Windows NT reports the computer's computername, the date and time when the statistics were last updated, and the following information:

-  The number of sessions that were started, disconnected automatically, and disconnected because of an error.
-  The number of kilobytes sent and received, and the average server-response time.
-  The number of errors and violations of password and permission limits.
-  The number of times the shared files, printers, and communication devices were used.
-  The number of times the size of the memory buffer was exceeded.


The Workstation service:

Windows NT reports the computer's computername, the date and time when the statistics were last updated, and the following information:

-  The number of bytes and SMBs received and transmitted.
-  The number of read and write operations that succeeded or failed.
-  The number of network errors.
-  The number of sessions that failed, disconnected, or were reconnected.
-  The number of connections to shared resources that succeeded or failed.

More Information About Net Statistics

 [Net Statistics--Examples](#)

 [Net Statistics](#)



Net Statistics--Examples

To display running services for which statistics are available, type

net stats

To display statistics for the Server service and prevent the display from scrolling, type

net statistics server | more

More Information About Net Statistics

 [Net Statistics--Notes](#)

 [Net Statistics](#)



Net Stop

Stops a Windows NT network service.

net stop *service*

Parameters

service

Includes **alerter, client service for netware, clipbook server, computer browser, directory replicator, ftp publishing service, lpdsvc, messenger, net logon, network dde, network dde dsdm, network monitor agent, nt lm security support provider, ole, remote access connection manager, remote access isnsap service, remote access server, remote procedure call (rpc) locator, remote procedure call (rpc) service, schedule, server, simple tcp/ip services, snmp, spooler, tcp/ip netbios helper, ups,** and **workstation.**

These services are available only on Windows NT Server: **file server for macintosh, gateway service for netware, microsoft dhcp server, print server for macintosh, windows internet name service.**

More Information About Net Stop



[Net Stop--Notes](#)



[Net Stop--Example](#)



Net Stop--Notes

Stop a service to halt the function it performs and to remove the software from memory.

Stopping the Server service prevents users from accessing the computer's shared resources. If you stop the Server service when users are accessing resources, Windows NT displays a warning message, requesting confirmation that you want to cancel the connections. A **y** response cancels all connections to the computer.

Before stopping the Server service, you can:



Pause the service (to disallow new connections).



Send a message advising users to disconnect from the server's resources.

Net stop can also stop network services not provided with Windows NT.

See Also

For information about

Using **/yes** and **/no**

Starting, pausing, and continuing services

See

[Net \(command options\)](#)

[Net Continue](#), [Net Pause](#), and [Net Start](#)

More Information About Net Stop



[Net Stop--Example](#)



[Net Stop](#)




Net Stop--Example

To stop the Server service, type

net stop server

More Information About Net Stop

 [Net Stop--Notes](#)

 [Net Stop](#)



Net Time

Synchronizes the computer's clock with that of another computer or domain. Used without the **/set** option, displays the time for another computer or domain.

net time [*computername* | **/domain[:name]**] [**/set**]

Parameters

computername

Is the name of a server you want to check or synchronize with.

/domain[:name]

Specifies the domain with which to synchronize time.

/set

Synchronizes the computer's clock with the time on the specified computer or domain.



Net Use

Connects a computer to or disconnects a computer from a shared resource, or displays information about computer connections. The command also controls persistent net connections.

net use [*devicename* | *] [*computername**sharename*[*\volume*]] [*password* | *] [/**user:** *domainname**username*] [/**delete**] [/**persistent**:{**yes** | **no**}]

net use *devicename* [/**home**[*password* | *]] [/**delete**:{**yes** | **no**}]

net use [/**persistent**:{**yes** | **no**}]

Parameters

none

Type **net use** without parameters to get a list of network connections.

devicename

Assigns a name to connect to the resource or specifies the device to be disconnected. There are two kinds of devicenames: disk drives (D: through Z:) and printers (LPT1: through LPT3). Type an asterisk instead of a specific devicename to assign the next available devicename.

*computername**sharename*

Is the name of the server and the shared resource. If the *computername* contains blank characters, enclose the double backslash (\\) and the *computername* in quotation marks (" "). The *computername* may be from 1 to 15 characters long.

\volume

Specifies a NetWare volume on the server. You must have Client Service for NetWare (Windows NT Workstation) or Gateway Service for NetWare (Windows NT Server) installed and running to connect to NetWare servers,

password

Is the password needed to access the shared resource.

*

Produces a prompt for the password. The password is not displayed when you type it at the password prompt.

/user

Specifies a different username with which the connection is made.

domainname

Specifies another domain. For example, **net use d: \\server\share /user:admin\mariel** connects the user mariel as if the connection were made from the admin domain. If domain is omitted, the current logged on domain is used.

username

Specifies the username with which to log on.

/home

Connects a user to their home directory.

/delete

Cancels the specified network connection. If the user specifies the connection with an asterisk, all network connections are cancelled.

/persistent

Controls the use of persistent network connections. The default is the setting used last. Deviceless connections are not persistent.

yes


Saves all connections as they are made, and restores them at next logon.

no

Does not save the connection being made or subsequent connections; existing connections will be restored at next logon. Use the **/delete** switch to remove persistent connections.

More Information About Net Use

 [_Net Use--Notes](#)

 [_Net Use--Examples](#)



Net Use--Notes

Use the **net use** command to connect to and disconnect from a network resource, and to view your current connections to network resources. You cannot disconnect from a shared directory if it's used as your current drive or is used by an active process

There are two ways to get information about a connection:



Type **net use *devicename*** to get information about a specific connection.



Type **net use** to get a list of all the computer's connections.

Deviceless connections

Deviceless connections are not persistent.

Connecting to NetWare servers

Once the Client or Gateway Service for NetWare software is installed and running, you can connect to a NetWare server on a Novell network. You use the same syntax as when connecting to a Windows Networking server, except you must include the volume you want to connect to.

More Information About Net Use



[Net Use--Examples](#)



[Net Use](#)



Net Use--Examples

To assign the disk-drive devicename E: to the LETTERS shared directory on the \\FINANCIAL server, type

net use e: \\financial\letters

To assign (map) the disk-drive devicename M: to the directory MARIA within the LETTERS volume on the FINANCIAL NetWare server, type

net use m: \\financial\letters\maria

To assign the devicename LPT1: to the LASER2 shared-printer queue on the \\ACCOUNTING server, type

net use lpt1: \\accounting\laser2

To disconnect from the LPT1: printer queue, type

net use lpt1: /delete

To assign the disk-drive devicename H: to a home directory as the user mariel, type

net use h: /home /user:mariel

To assign the disk-drive devicename F: to \\FINANCIAL server's SCRATCH shared directory, which requires the password *hctarcs*, but not make the connection persistent, type

net use f: \\financial\scratch hctarcs /persistent:no

To disconnect from the \\FINANCIAL\SCRATCH directory, type

net use f: \\financial\scratch /delete

To connect to a resource shared on the FINANCIAL 2 server type

net use k: "\\financial 2"\memos

You must use quotation marks around a servername that includes a space. If you omit the quotation marks, Windows NT displays an error message.

To restore the current connections at each logon, regardless of future changes, type

net use /persistent:yes

More Information About Net Use



[Net Use--Notes](#)



[Net Use](#)



Net User

Adds or modifies user accounts or displays user-account information.

net user [*username* [*password* | *] [*options*]] [/domain]

net user *username* {*password* | *} /add [*options*] [/domain]

net user *username* [/delete] [/domain]

Parameters

none

Type **net user** without parameters to view a list of the user accounts on the computer.

username

Is the name of the user account to add, delete, modify, or view. The name of the user account can have as many as 20 characters.

password

Assigns or changes a password for the user's account. A password must satisfy the minimum length set with the **/minpwlen** option of the **net accounts** command. It can have as many as 14 characters.

*

Produces a prompt for the password. The password is not displayed when you type it at a password prompt.

/domain

Performs the operation on the primary domain controller of the computer's primary domain.

This parameter applies only to Windows NT Workstation computers that are members of a Windows NT Server domain. By default, Windows NT Server computers perform operations on the primary domain controller.

NOTE: This action is taken on the primary domain controller of the computer's primary domain. This may not be the logged on domain.

/add

Adds a user account to the user accounts database.

/delete

Removes a user account from the user accounts database.

options are as follows:

/active:{no | yes}

Enables or disables the user account. If the user account is not active, the user cannot access resources on the computer. The default is yes (active).

/comment:"text"

Provides a descriptive comment about the user's account. This comment can have as many as 48 characters. Enclose the text in quotation marks.

/countrycode:nnn

Uses the operating-system country codes to implement the specified language files for a user's help and error messages. A value of 0 signifies the default country code.

/expires:{date | never}

Causes the user account to expire if *date* is set; never sets no time limit on the user account. Expiration dates can be in *mm/dd/yy*, *dd/mm/yy*, or *mmm,dd,yy* format, depending on the **/countrycode**. Note that the account expires at the beginning of the date specified. Months can be a number, spelled out, or abbreviated with three letters. Years can be two or four numbers. Use commas or slashes to separate parts of the date (no spaces). If *yy* is omitted, the next occurrence of the date (according to your computer's date and time) is assumed. For example, the following *date* entries are equivalent if entered between Jan. 10, 1994, and Jan. 8, 1995:

jan,9
1/9/95
january,9,1995
1/9

/fullname:"name"

Specifies a user's full name rather than a username. Enclose the name in quotation marks.

/homedir:path

Sets the path for the user's home directory. The path must exist.

/homedirreq:{yes | no}

Sets whether a home directory is required.

/passwordchg:{yes | no}

Specifies whether users can change their own password. The default is yes.

/passwordreq:{yes | no}

Specifies whether a user account must have a password. The default is yes.

/profilepath:[path]

Sets a path for the user's logon profile. This pathname points to a registry profile.

/scriptpath:path

Sets a path for the user's logon script. *Path* cannot be an absolute path; *path* is relative to %systemroot%\SYSTEM32\REPL\IMPORT\SCRIPTS.

/times:{times | all}

Specifies the times the user is allowed to use the computer. The *times* value is expressed as *day[-day][,day[-day]]*, *time[-time][,time[-time]]*, limited to 1-hour time increments. Days can be spelled out or abbreviated (M,T,W,Th,F,Sa,Su). Hours can be 12- or 24-hour notation. For 12-hour notation, use AM, PM, or A.M., P.M. The value all means a user can always log on. A null value (blank) means a user can never log on. Separate day and time with commas, and units of day and time with semicolons (for example, M,4AM-5PM;T,1PM-3PM). Do not use spaces when designating /times.


/usercomment:"text"


Lets an administrator add or change the "User comment" for the account. Enclose the text in quotation marks.

/workstations:{computername[,...] | *}

Lists as many as eight workstations from which a user can log on to the network. Separate multiple entries in the list with commas. If /workstations has no list, or if the list is *, the user can log on from any computer.

More Information About Net User

 [Net User--Notes](#)

 [Net User--Examples](#)




Net User--Notes

This command can also be typed **net users**.

Use the **net user** command to create and control user accounts for users in a domain. User-account information is stored in the user accounts database.

Note

 When you type the **net user** command at a computer running Windows NT Server, changes to the user-accounts database automatically occur on the primary domain controller and then are replicated to backup domain controllers. This applies only to Windows NT Server domains.


See Also

 [_Net Accounts](#)

 [_Net Localgroup](#)

 [_Net Group](#)

More Information About Net User

 [_Net User--Examples](#)

 [_Net User](#)



Net User--Examples

To display a list of all user accounts for the local computer, type

net user

To view information about the user account *jimmyh*, type

net user jimmyh

To add a user account for Henry James, with logon rights from 8 A.M. to 5 P.M. Monday through Friday (no spaces in time designations), a mandatory password, and the user's full name, type

net user henryj henryj /add /passwordreq:yes /times:monday-friday,8am-5pm/fullname:"Henry James"

The username (*henryj*) is entered the second time as the password.

To set *johnsw*'s logon time (8 A.M. to 5 P.M.) using 24-hour notation, type

net user johnsw /time:M-F,08:00-17:00

To set *johnsw*'s logon time (8 A.M. to 5 P.M.) using 12-hour notation, type

net user johnsw /time:M-F,8am-5pm

To specify logon hours of 4 A.M. until 5 P.M. on Monday, 1 P.M. until 3 P.M. on Tuesday, and 8 A.M. until 5 P.M. Wednesday through Friday for *marysl*, type

net user marysl /time:M,4am-5pm;T,1pm-3pm;W-F,8:00-17:00

To set **/homedirreq** to **yes** for *henryj* and assign \\SERVER\USERS\HENRYJ as the home directory, type

net user henryj /homedirreq:yes

/homedir \\SERVER\USERS\HENRYJ

More Information About Net User



[Net User--Notes](#)



[Net User](#)



Net View

Displays a list of domains, a list of computers, or the resources being shared by the specified computer.

net view [*\\computername* | **/domain**:*domainname*]

net view /network:nw [*\\computername*]

Parameters

none

Type **net view** without parameters to display a list of computers in your current domain.

\\computername

Specifies the computer whose shared resources you want to view.

/domain:*domainname*

Specifies the domain for which you want to view the available computers. If *domainname* is omitted, displays all domains in the network.

/network:nw

Displays all available servers on a NetWare network. If a *computername* is specified, the resources available on that computer in the NetWare network will be displayed. Other networks that are added to the system may also be specified with this switch.

More Information About Net View



[Net View--Notes](#)



[Net View--Examples](#)



Net View--Notes

Use the **net view** command to display a list of computers. The display is similar to the following:


Server Name	Remark

\\PRODUCTION	Production file server
\\PRINT1	Printer room, first floor
\\PRINT2	Printer room, second floor

See Also

 [Net Use](#)

More Information About Net View

 [Net View--Examples](#)

 [Net View](#)



Net View--Examples

To see a list of the resources shared by the \\PRODUCTION computer, type

net view \\production

To see the resources available on the NetWare server \\MARKETING, type

net view /network:nw \\marketing

To see a list of the computers in the sales domain or workgroup, type

net view /domain:sales

To see all the servers in a NetWare network, type

net view /network:nw

More Information About Net View



[Net View--Notes](#)



[Net View](#)



TCP/IP Utilities and Services

Select a utility or service to get more information

TCP/IP Utilities

- | | |
|---|---|
| <input type="checkbox"/> arp | <input type="checkbox"/> nslookup |
| <input type="checkbox"/> finger | <input type="checkbox"/> ping |
| <input type="checkbox"/> ftp | <input type="checkbox"/> rcp |
| <input type="checkbox"/> hostname | <input type="checkbox"/> rexec |
| <input type="checkbox"/> ipconfig | <input type="checkbox"/> route |
| <input type="checkbox"/> lpq | <input type="checkbox"/> rsh |
| <input type="checkbox"/> lpr | <input type="checkbox"/> tftp |
| <input type="checkbox"/> nbtstat | <input type="checkbox"/> tracert |
| <input type="checkbox"/> netstat | |

TCP/IP Services

- | | |
|--|--|
| <input type="checkbox"/> net start dhcp client | <input type="checkbox"/> net start simple tcp/ip services |
| <input type="checkbox"/> net start ftp server | <input type="checkbox"/> net start snmp |
| <input type="checkbox"/> net start lpdsvc | <input type="checkbox"/> net start tcp/ip netbios helper |
| <input type="checkbox"/> net start microsoft dhcp server | <input type="checkbox"/> net start windows internet name service |

The TCP/IP utilities offer network connections to non-Microsoft computers such as UNIX workstations. You must have the TCP/IP network protocol installed to use the TCP/IP utilities.



Arp

Displays and modifies the IP-to-Ethernet or token ring physical address translation tables used by address resolution protocol (ARP). This command is available only if the TCP/IP protocol has been installed.

arp -a [*inet_addr*] [-N [*if_addr*]]

arp -d *in_addr* [*if_addr*]

arp -s *in_addr ether_addr* [*if_addr*]

Parameters

-a

Displays current ARP entries by querying TCP/IP. If *inet_addr* is specified, only the IP and physical addresses for the specified computer are displayed.

-g

Identical to **-a**.

inet_addr

Specifies an IP address in dotted decimal notation.

-N

Displays the ARP entries for the network interface specified by *if_addr*.

if_addr

Specifies, if present, the IP address of the interface whose address translation table should be modified. If not present, the first applicable interface will be used.

-d

Deletes the entry specified by *inet_addr*.

-s

Adds an entry in the ARP cache to associate the IP address *inet_addr* with the physical address *ether_addr*. The physical address is given as 6 hexadecimal bytes separated by hyphens. The IP address is specified using dotted decimal notation. The entry is permanent, that is, it will not be automatically removed from the cache after the timeout expires.

ether_addr

Specifies a physical address.



Finger

Displays information about a user on a specified system running the Finger service. Output varies based on the remote system. This command is available only if the TCP/IP protocol has been installed.

finger [-l] [*user*]*@computer* [...]

Parameters

-l

Displays information in long list format.

user

Specifies the user you want information about. Omit the user parameter to display information about all users on the specified computer.

@computer

Specifies the server on the remote system whose users you want information about.



Ftp

Transfers files to and from a computer running an FTP server service (sometimes called a daemon). **Ftp** can be used interactively. See ftp commands for a description of available **ftp** commands. This command is available only if the TCP/IP protocol has been installed.

ftp [-v] [-d] [-i] [-n] [-g] [-s:filename] [-a] [-w:windowsize] [computer]

Parameters

-v

Suppresses display of remote server responses.

-n

Suppresses auto-login upon initial connection.

-i

Turns off interactive prompting during multiple file transfers.

-d

Enables debugging, displaying all **ftp** commands passed between the client and server.

-g

Disables filename globbing, which permits the use of wildcard characters in local file and path names. (See the **glob** command in the online Command Reference.)

-s:filename

Specifies a text file containing **ftp** commands; the commands will automatically run after **ftp** starts. No spaces are allowed in this parameter. Use this switch instead of redirection (>).

-a

Use any local interface when binding data connection.

-w:windowsize

Overrides the default transfer buffer size of 4096.

computer

Specifies the computer name or IP address of the remote computer to connect to. The computer, if specified, must be the last parameter on the line.

More Information About Ftp



[Ftp Commands](#)



Ftp Commands

Select an **ftp** command to get more information

<input type="checkbox"/> Ftp: !	<input type="checkbox"/> Ftp: glob	<input type="checkbox"/> Ftp: put
<input type="checkbox"/> Ftp: ?	<input type="checkbox"/> Ftp: hash	<input type="checkbox"/> Ftp: pwd
<input type="checkbox"/> Ftp: append	<input type="checkbox"/> Ftp: help	<input type="checkbox"/> Ftp: quit
<input type="checkbox"/> Ftp: ascii	<input type="checkbox"/> Ftp: lcd	<input type="checkbox"/> Ftp: quote
<input type="checkbox"/> Ftp: bell	<input type="checkbox"/> Ftp: literal	<input type="checkbox"/> Ftp: recv
<input type="checkbox"/> Ftp: binary	<input type="checkbox"/> Ftp: ls	<input type="checkbox"/> Ftp: remotehelp
<input type="checkbox"/> Ftp: bye	<input type="checkbox"/> Ftp: mdelete	<input type="checkbox"/> Ftp: rename
<input type="checkbox"/> Ftp: cd	<input type="checkbox"/> Ftp: mdir	<input type="checkbox"/> Ftp: rmdir
<input type="checkbox"/> Ftp: close	<input type="checkbox"/> Ftp: mget	<input type="checkbox"/> Ftp: send
<input type="checkbox"/> Ftp: debug	<input type="checkbox"/> Ftp: mkdir	<input type="checkbox"/> Ftp: status
<input type="checkbox"/> Ftp: delete	<input type="checkbox"/> Ftp: mls	<input type="checkbox"/> Ftp: trace
<input type="checkbox"/> Ftp: dir	<input type="checkbox"/> Ftp: mput	<input type="checkbox"/> Ftp: type
<input type="checkbox"/> Ftp: disconnect	<input type="checkbox"/> Ftp: open	<input type="checkbox"/> Ftp: user
<input type="checkbox"/> Ftp: get	<input type="checkbox"/> Ftp: prompt	<input type="checkbox"/> Ftp: verbose



Ftp: !

Runs the specified command on the local computer.

! command

Parameter

command

Specifies the command to run on the local computer. If *command* is omitted, the local command prompt is displayed; type **exit** to return to **ftp**.



Ftp: ?

Displays descriptions for **ftp** commands. **?** is identical to **help**.

? [*command*]

Parameter

command

Specifies the name of the command about which you want a description. If *command* is not specified, **ftp** displays a list of all commands.



Ftp: append

Appends a local file to a file on the remote computer using the current file type setting.

append *local-file* [*remote-file*]

Parameters

local-file

Specifies the local file to add.

remote-file

Specifies the file on the remote computer to which *local-file* will be added. If *remote-file* is omitted, the local filename is used for the remote filename.




Ftp: ascii

Sets the file transfer type to ASCII, the default.

ascii

Note

 FTP supports two file transfer types, ASCII and binary image. ASCII should be used when transferring text files. See also **binary**.

In ASCII mode, character conversions to and from the network standard character set are performed. For example, end-of-line characters are converted as necessary, based on the target operating system.



Ftp: bell

Toggles a bell to ring after each file transfer command is completed. By default, the bell is off.

bell



Ftp: binary

Sets the file transfer type to binary.

binary

Note



FTP supports two file transfer types, ASCII and binary image. Binary should be used when transferring executable files. In binary mode, the file is moved byte-by-byte. See also [ascii](#).



Ftp: bye

Ends the FTP session with the remote computer and exits **ftp**.

bye



Ftp: cd

Changes the working directory on the remote computer.

cd *remote-directory*

Parameter

remote-directory

Specifies the directory on the remote computer to change to.



Ftp: close

Ends the FTP session with the remote server and returns to the command interpreter.

close



Ftp: delete

Deletes files on remote computers.

delete *remote-file*

Parameter

remote-file

Specifies the file to delete.



Ftp: debug

Toggles debugging. When debugging is on, each command sent to the remote computer is printed, preceded by the string --->. By default, debugging is off.

debug



Ftp: dir

Displays a list of a remote directory's files and subdirectories.

dir [*remote-directory*] [*local-file*]

Parameters

remote-directory

Specifies the directory for which you want to see a listing. If no directory is specified, the current working directory on the remote computer is used.

local-file

Specifies a local file to store the listing. If not specified, output is displayed on the screen.



Ftp: disconnect

Disconnects from the remote computer, retaining the **ftp** prompt.

disconnect



Ftp: get

Copies a remote file to the local computer using the current file transfer type.

get *remote-file* [*local-file*]

Parameters

remote-file

Specifies the remote file to copy.

local-file

Specifies the name to use on the local computer. If not specified, the file is given the *remote-file* name.



Ftp: glob

Toggles filename globbing. Globbing permits use of wildcard characters in local file or path names. By default, globbing is on.

glob



Ftp: hash

Toggles hash-sign (#) printing for each data block transferred. The size of a data block is 2048 bytes. By default, hash mark printing is off.

hash



Ftp: help

Displays descriptions for **ftp** commands.

help [*command*]

Parameter

command

Specifies the name of the command about which you want a description. If *command* is not specified, **ftp** displays a list of all commands.



Ftp: lcd

Changes the working directory on the local computer. By default, the working directory is the directory in which **ftp** was started.

lcd [*directory*]

Parameter

directory

Specifies the directory on the local computer to change to. If *directory* is not specified, the current working directory on the local computer is displayed.



Ftp: literal

Sends arguments, verbatim, to the remote FTP server. A single FTP reply code is expected in return.

literal *argument* [...]

Parameter

argument

Specifies the argument to send to the FTP server.

**Ftp: ls**

Displays an abbreviated list of a remote directory's files and subdirectories.

ls [*remote-directory*] [*local-file*]

Parameters

remote-directory

Specifies the directory for which you want to see a listing. If no directory is specified, the current working directory on the remote computer is used.

local-file

Specifies a local file to store the listing. If not specified, output is displayed on the screen.



Ftp: mdelete

Deletes files on remote computers.

mdelete *remote-files* [...]

Parameter

remote-files

Specifies the remote files to delete.



Ftp: mdir

Displays a list of a remote directory's files and subdirectories. **Mdir** allows you to specify multiple files.

mdir *remote-files* [...] *local-file*

Parameters

remote-files

Specifies the directory for which you want to see a listing. *Remote-files* must be specified; type - to use the current working directory on the remote computer.

local-file

Specifies a local file to store the listing. Type - to display the listing on the screen.



Ftp: mget

Copies remote files to the local computer using the current file transfer type.

mget *remote-files* [...]

Parameter

remote-files

Specifies the remote files to copy to the local computer.



Ftp: mkdir

Creates a remote directory.

mkdir *directory*

Parameter

directory

Specifies the name of the new remote directory.



Ftp: mls

Displays an abbreviated list of a remote directory's files and subdirectories.

mls *remote-files* [...] *local-file*

Parameters

remote-files

Specifies the files for which you want to see a listing. *Remote-files* must be specified; type - to use the current working directory on the remote computer.

local-file

Specifies a local file to store the listing. Type - to display the listing on the screen.



Ftp: mput

Copies local files to the remote computer using the current file transfer type.

mput *local-files* [...]

Parameter

local-files

Specifies the local files to copy to the remote computer.

**Ftp: open**

Connects to the specified FTP server.

open *computer* [*port*]

Parameters

computer

Specifies the remote computer to connect to. Computer can be specified by IP address or computer name (a DNS or HOSTS file must be available). If auto-login is on (default), FTP also attempts to automatically log the user in to the FTP server (see [Ftp](#) to disable auto-login).

port

Specifies a port number to use to contact an FTP server.



Ftp: prompt

Toggles prompting. **Ftp** prompts during multiple file transfers to allow you to selectively retrieve or store files; **mget** and **mput** transfer all files if prompting is turned off. By default, prompting is on.

prompt



Ftp: put

Copies a local file to the remote computer using the current file transfer type.

put *local-file* [*remote-file*]

Parameters

local-file

Specifies the local file to copy.

remote-file

Specifies the name to use on the remote computer. If not specified, the file is given the *local-file* name.



Ftp: pwd

Displays the current directory on the remote computer.

pwd



Ftp: quit

Ends the FTP session with the remote computer and exits **ftp**.

quit



Ftp: quote

Sends arguments, verbatim, to the remote FTP server. A single FTP reply code is expected in return. **Quote** is identical to **literal**.

quote *argument* [...]

Parameter

argument

Specifies the argument to send to the FTP server.



Ftp: `recv`

Copies a remote file to the local computer using the current file transfer type. **Recv** is identical to **get**.

recv *remote-file* [*local-file*]

Parameters

remote-file

Specifies the remote file to copy.

local-file

Specifies the name to use on the local computer. If not specified, the file is given the *remote-file* name.



Ftp: remotehelp

Displays help for remote commands.

remotehelp [*command*]

Parameter

command

Specifies the name of the command about which you want help. If *command* is not specified, **ftp** displays a list of all remote commands.



Ftp: rename

Renames remote files.

rename *filename newfilename*

Parameters

filename

Specifies the file you want to rename.

newfilename

Specifies the new filename.



Ftp: rmdir

Deletes a remote directory.

rmdir *directory*

Parameter

directory

Specifies the name of the remote directory to delete.



Ftp: send

Copies a local file to the remote computer using the current file transfer type. **Send** is identical to **put**.

send *local-file* [*remote-file*]

Parameters

local-file

Specifies the local file to copy.

remote-file

Specifies the name to use on the remote computer. If not specified, the file is given the *local-file* name.



Ftp: status

Displays the current status of FTP connections and toggles.

status



Ftp: trace

Toggles packet tracing; **trace** displays the route of each packet when running an **ftp** command.

trace

**Ftp: type**

Sets or displays the file transfer type.

type [*type-name*]

Parameter

type-name

Specifies the file transfer type; the default is ASCII. If *type-name* is not specified, the current type is displayed.

Notes

FTP supports two file transfer types, ASCII and binary image.



ASCII should be used when transferring text files. In ASCII mode, character conversions to and from the network standard character set are performed. For example, end-of-line characters are converted as necessary, based on the destination's operating system.



Binary should be used when transferring executable files. In binary mode, the file is moved byte-by-byte.

See Also

[ascii](#)



[binary](#)



Ftp: user

Specifies a user to the remote computer.

user *user-name* [*password*] [*account*]

Parameters

user-name

Specifies a user name with which to log in to the remote computer.

password

Specifies the password for *user-name*. If not specified, but required, **ftp** prompts for the password.

account

Specifies an account with which to log on to the remote computer. If *account* is not specified, but required, **ftp** prompts for the account.



Ftp: verbose

Toggles verbose mode. If on, all **ftp** responses are displayed; when a file transfer completes, statistics regarding the efficiency of the transfer are also displayed. By default, verbose is on.

verbose

**Hostname**

Prints the name of the current computer (host). This command is available only if the TCP/IP protocol has been installed.

hostname



Ipconfig

This diagnostic command displays all current TCP/IP network configuration values. This command is of particular use on systems running DHCP, allowing users to determine which TCP/IP configuration values have been configured by DHCP.

ipconfig [/all | /renew *[adapter]* | /release *[adapter]*]

Parameters

all

Produces a full display. Without this switch, ipconfig displays only the IP address, subnet mask, and default gateway values for each network card.

renew *[adapter]*

Renews DHCP configuration parameters. This option is available only on systems running the DHCP Client service. To specify an adapter name, type the adapter name that appears when you use ipconfig without parameters.

release *[adapter]*

Releases the current DHCP configuration. This option disables TCP/IP on the local system and is available only on DHCP clients. To specify an adapter name, type the adapter name that appears when you use ipconfig without parameters.

With no parameters, the ipconfig utility presents all of the current TCP/IP configuration values to the user, including IP address and subnet mask. This utility is especially useful on systems running DHCP, allowing users to determine which values have been configured by DHCP.



Lpq

This diagnostic utility is used to obtain status of a print queue on a computer running the LPD server.

lpq *-S*Server *-P*Printer [*-l*]

Parameters

*-S*Server

Specifies the name of the computer that has the printer attached to it.

*-P*Printer

Specifies the name of the printer for the desired queue.

-l

Specifies that a detailed status should be given.



Lpr

This connectivity utility is used to print a file to a computer running an LPD server.

lpr -S*Server* **-P***Printer* [**-C***Class*] [**-J***Jobname*] [**-O** *option*] *filename*

Parameters

-S*Server*

Specifies the name or IP address of the computer that has the printer attached to it.

-P*Printer*

Specifies the name of the printer for the desired queue.

-C*Class*

Specifies the content of the banner page for the class.

-J*Jobname*

Specifies the name of this job.

-O *option*

Indicates the type of file. The default is a text file. Use **-O***l* (lowercase *l*) for a binary file (for example, PostScript).

filename

The name of the file to be printed.



Nbtstat

This diagnostic command displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP). This command is available only if the TCP/IP protocol has been installed.

nbtstat [-a *remotename*] [-A *IP address*] [-c] [-n] [-R] [-r] [-S] [-s] [*interval*]

Parameters

-a *remotename*

Lists the remote computer's name table using its name.

-A *IP address*

Lists the remote computer's name table using its IP address.

-c

Lists the contents of the NetBIOS name cache giving the IP address of each name.

-n

Lists local NetBIOS names. Registered indicates that the name is registered by broadcast (Bnode) or WINS (other node types).

-R

Reloads the LMHOSTS file after purging all names from the NetBIOS name cache.

-r

Lists name resolution statistics for Windows networking name resolution. On a Windows NT computer configured to use WINS, this option returns the number of names resolved and registered via broadcast or via WINS.

-S

Displays both client and server sessions, listing the remote computers by IP address only.

-s

Displays both client and server sessions. It attempts to convert the remote computer IP address to a name using the HOSTS file.

interval

Redisplays selected statistics, pausing *interval* seconds between each display. Press **CTRL+C** to stop redisplaying statistics. If this parameter is omitted, nbtstat prints the current configuration information once.

More Information About Nbtstat



[Nbtstat--Notes](#)



Nbtstat--Notes

The column headings generated by the **Nbtstat** utility have the following meanings.

Input

Number of bytes received.

Output

Number of bytes sent.

In/Out

Whether the connection is from the computer (outbound) or from another system to the local computer (inbound).

Life

The remaining time that a name table cache entry will live before it is purged.

Local Name

The local NetBIOS name associated with the connection.

Remote Host

The name or IP address associated with the remote computer.

Type

Refers to the type of name. A name can either be a unique name or a group name.

<03>

Each NetBIOS name is 16 characters long. This last byte often has special significance since the same name may be present several times on a computer differing only in the last byte. This notation is simply the last byte converted to hexadecimal. <20> is a space in ASCII for example.


State

The state of NetBIOS connections. The possible states are:

State	Meaning
Connected	The session has been established.
Associated	A connection endpoint has been created and associated with an IP address.
Listening	This endpoint is available for an inbound connection.
Idle	This endpoint has been opened but cannot receive connections.
Connecting	The session is in the connecting phase where the name-to-IP address mapping of the destination is being resolved.
Accepting	An inbound session is currently being accepted and will be connected

	shortly.
Reconnecting	A session is trying to reconnect if it failed to connect on the first attempt.
Outbound	A session is in the connecting phase where the TCP connection is currently being created.
Inbound	An inbound session is in the connecting phase.
Disconnecting	A session is in the process of disconnecting.
Disconnected	The local computer has issued a disconnect, and it is waiting for confirmation from the remote system.

More Information About Nbtstat

 [Nbtstat](#)



Netstat

Displays protocol statistics and current TCP/IP network connections. This command is available only if the TCP/IP protocol has been installed.

netstat [-a] [-e] [-n] [-s] [-p *protocol*] [-r] [*interval*]

Parameters

-a

Displays all connections and listening ports; server connections are normally not shown.

-e

Displays Ethernet statistics. This may be combined with the **-s** option.

-n

Displays addresses and port numbers in numerical form (rather than attempting name look-ups).

-s

Displays per-protocol statistics. By default, statistics are shown for TCP, UDP, ICMP, and IP; the **-p** option may be used to specify a subset of the default.

-p *protocol*

Shows connections for the protocol specified by *proto*; *proto* may be **tcp** or **udp**. If used with the **-s** option to display per-protocol statistics, *proto* may be **tcp**, **udp**, **icmp**, or **ip**.

-r

Displays the contents of the routing table.

interval

Redisplays selected statistics, pausing *interval* seconds between each display. Press **CTRL+C** to stop redisplaying statistics. If this parameter is omitted, **netstat** prints the current configuration information once.

More Information About Netstat



[Netstat--Notes](#)



Netstat--Notes

The **Netstat** utility provides statistics on the following network components.

Foreign Address

The IP address and port number of the remote computer to which the socket is connected. The name corresponding to the IP address is shown instead of the number if the HOSTS file contains an entry for the IP address. In cases where the port is not yet established, the port number is shown as an asterisk (*).

Local Address

The IP address of the local computer, as well as the port number the connection is using. The name corresponding to the IP address is shown instead of the number if the HOSTS file contains an entry for the IP address. In cases where the port is not yet established, the port number is shown as an asterisk (*).

Proto

The name of the protocol used by the connection.

(state)

Indicates the state of TCP connections only. The possible states are

CLOSED	FIN_WAIT_1	SYN_RECEIVED
CLOSE_WAIT	FIN_WAIT_2	SYN_SEND
ESTABLISHED	LISTEN	TIMED_WAIT
LAST_ACK		

More Information About Netstat

 [Netstat](#)



Nslookup

This diagnostic tool displays information from Domain Name System (DNS) name servers. Before using this tool, you should be familiar with how DNS works. Nslookup is available only if the TCP/IP protocol has been installed.

nslookup [-option ...] [*computer-to-find* | - [server]]

Modes

Nslookup has two modes: interactive and non-interactive.

If you only need to look up a single piece of data, use non-interactive mode. For the first argument, type the name or IP address of the computer to be looked up. For the second argument, type the name or IP address of a DNS name server. If you omit the second argument, the default DNS name server will be used.

If you need to look up more than one piece of data, you can use interactive mode. Type a hyphen (-) for the first argument and the name or IP address of a DNS name server for the second argument. Or, omit both arguments (the default DNS name server will be used).

Parameters

-option ...

Specifies one or more nslookup commands as a command-line option. For a list of commands, see [Nslookup Commands](#). Each option consists of a hyphen (-) followed immediately by the command name and, in some cases, an equal sign (=) and then a value. For example, to change the default query type to host (computer) information and the initial timeout to 10 seconds, you would type:

```
nslookup -querytype=hinfo -timeout=10
```

The command line length must be less than 256 characters.

computer-to-find

Look up information for *computer-to-find* using the current default server or using *server* if specified. If *computer-to-find* is an IP address and the query type is **A** or **PTR**, the name of the computer is returned. If *computer-to-find* is a name and does not have a trailing period, the default DNS domain name is appended to the name. (This behavior depends on the state of the **set** options: **domains**, **srchlist**, **defname**, and **search**.) To look up a computer not in the current DNS domain, append a period to the name.

If you type a hyphen (-) instead of *computer-to-find*, the command prompt changes to nslookup interactive mode.

server

Use this server as the DNS name server. If you omit *server*, the default DNS name server is used.

More Information About Nslookup



[Nslookup Commands](#)



[Nslookup Notes](#)



Nslookup Commands

Select an **nslookup** command to get more information

- [Nslookup: help](#)
- [Nslookup: exit](#)
- [Nslookup: finger](#)
- [Nslookup: ls](#)
- [Nslookup: lserver](#)
- [Nslookup: root](#)
- [Nslookup: server](#)
- [Nslookup: set](#)
- [Nslookup: set all](#)
- [Nslookup: set class](#)
- [Nslookup: set \[no\]debug](#)
- [Nslookup: set \[no\]id2](#)
- [Nslookup: set \[no\]def\[name\]](#)
- [Nslookup: set do\[main\]](#)
- [Nslookup: set \[no\]ignore](#)
- [Nslookup: set port](#)
- [Nslookup: set querytype](#)
- [Nslookup: set \[no\]recurse](#)
- [Nslookup: set retry](#)
- [Nslookup: set rotate](#)
- [Nslookup: set \[no\]search](#)
- [Nslookup: set srchlist](#)
- [Nslookup: set timeout](#)
- [Nslookup: set type](#)
- [Nslookup: set \[no\]vc](#)
- [Nslookup: view](#)



Nslookup: help

Displays a brief summary of **nslookup** commands. The **?** is a synonym for the **help** command.

help | ?



Nslookup: exit

Exits **nslookup**.

exit



Nslookup: finger

Connects with the finger server on the current computer. The current computer is defined when a previous lookup for a computer was successful and returned address information (see the **set querytype=A** command).

finger [*username*] [> *filename*] | [>> *filename*]

Parameters

username

Specifies the name of the user to look up.

filename

Specifies a filename in which to save the output. (You can use the > and >> characters to redirect the output in the usual manner.)



Nslookup: Is

Lists information for a DNS domain. The default output contains computer names and their IP addresses. (When output is directed to a file, hash marks are printed for every 50 records received from the server.)

Is [*option*] *dnsdomain* [> *filename*] | [>> *filename*]

Parameters

option

-t *querytype*

Lists all records of the specified type. (For a description of *querytype*, see the **set querytype** topic.)

-a

Lists aliases of computers in the DNS domain. (It is a synonym for **-t CNAME**.)

-d

Lists all records for the DNS domain. (It is a synonym for **-t ANY**.)

-h

Lists CPU and operating system information for the DNS domain. (It is a synonym for **-t HINFO**.)

-s

Lists well-known services of computers in the DNS domain. (It is a synonym for **-t WKS**.)

dnsdomain

DNS domain for which you want information.

filename

Specifies a filename in which to save the output. (You can use the > and >> characters to redirect the output in the usual manner.)

**Nslookup: lserver**

Changes the default server to the specified DNS domain. **lserver** uses the initial server to look up the information about the specified DNS domain. (This is in contrast to the **server** command, which uses the current default server.)

lserver *dnsdomain*

Parameters

dnsdomain

New DNS domain for the default server.



Nslookup: root

Changes the default server to the server for the root of the DNS domain name space. Currently, the computer ns.nic.ddn.mil is used. (This command is a synonym for **lserver ns.nic.ddn.mil**.) The name of the root server can be changed with the **set root** command.

root

**Nslookup: server**

Changes the default server to the specified DNS domain. **Server** uses the current default server to look up the information about the specified DNS domain. (This is in contrast to the **lserver** command, which uses the initial server.)

dnsdomain

Parameters

dnsdomain

New DNS domain for the default server.



Nslookup: set

Changes configuration settings that affect how lookups function.

set *keyword*[=*value*]

Parameters

keyword

The keywords are listed under individual help topics. See **set** *keyword*. Use **set all** to see a listing of the current settings.



Nslookup: set all

Prints the current values of the configuration settings. Also prints information about the default server and computer (host).

set all



Nslookup: set cl[ass]

Change the query class. (The class specifies the protocol group of the information.)

set cl[ass]=value

Default = IN.

Parameters

value

IN

Internet class.

CHAOS

Chaos class.

HESIOD

MIT Athena Hesiod class.

ANY

Wildcard (any of the above).



Nslookup: set [no]deb[ug]

Turn debugging mode on or off. With debugging on, more information is printed about the packet sent to the server and the resulting answer.

set [no]deb[ug]

Default = nodebug.



Nslookup: set [no]d2

Turn exhaustive debugging mode on or off. Essentially all fields of every packet are printed

set [no]d2

Default = nod2.



Nslookup: set [no]def[name]

If set, append the default DNS domain name to a single-component lookup request. (A single component is a component that contains no periods.)

set [no]def[name]

Default = defname.



Nslookup: set do[main]

Change the default DNS domain to the name specified. The default DNS domain name is appended to a lookup request depending on the state of the **defname** and **search** options. The DNS domain search list contains the parents of the default DNS domain if it has at least two components in its name. For example, if the default DNS domain is mfg.widgets.com, the search list is mfg.widgets.com and widgets.com. Use the **set srchlist** command to specify a different list. Use the **set all** command to display the list.

set do[main]=*name*

Default = value from hostname.

Parameters

name

New name for the default DNS domain.



Nslookup: set [no]ig[nore]

If set, ignore packet truncation errors.

set [no]ig[nore]

Default = noignore.



Nslookup: set po[rt]

Change the default TCP/UDP DNS name server port to the value specified.

set po[rt]=*value*

Default = 53.

Parameters

value

New value for the default TCP/UDP DNS name server port.



Nslookup: set q[querytype]

Changes the type of information query. More information about types can be found in Request For Comment (RFC) 1035. (The **set type** command is a synonym for **set querytype**.)

set q[querytype]=value

Default = A.

Parameters

value

A

Computer's IP address.

ANY

All types of data.

CNAME

Canonical name for an alias.

GID

Group identifier of a group name.

HINFO

Computer's CPU and operating system type.

MB

Mailbox domain name.

MG

Mail group member.

MINFO

Mailbox or mail list information.

MR

Mail rename domain name.

MX

Mail exchanger.

NS

DNS name server for the named zone.

PTR

Computer name if the query is an IP address, otherwise the pointer to other information.

SOA

DNS domain's start-of-authority record

TXT

Text information.

UID

User ID.

UINFO

User information.

WKS

Well-known service description.



Nslookup: set [no]rec[urse]

If set, tells the DNS name server to query other servers if it does not have the information.

set [no]rec[urse]

Default = recurse.



Nslookup: set ret[ry]

Sets the number of retries. When a reply to a request is not received within a certain amount of time (changed with **set timeout**), the timeout period is doubled and the request is resent. The retry value controls how many times a request is resent before giving up.

set ret[ry]=*number*

Default = 4.

Parameters

number

New value for the number of retries.



Nslookup: set ro[ot]

Changes the name of the root server. This affects the **root** command.

set ro[ot]=computerr

Default = ns.nic.ddn.mil.

Parameters

computerr

New name for the root server.



Nslookup: set [no]sea[rch]

If set and the lookup request contains at least one period but does not end with a trailing period, append the DNS domain names in the DNS domain search list to the request until an answer is received.

set [no]sea[rch]

Default = search.

**Nslookup: set srchl[ist]**

Changes the default DNS domain name and search list. A maximum of 6 names separated by slashes (/) can be specified. This command overrides the default DNS domain name and search list of the **set domain** command. Use the **set all** command to display the list.

Set srchl[ist] *name1/name2/...*

Default = value based on hostname.

Parameters

name1/name2/...

New names for the default DNS domain and search list. For example,

set srchlist=*mfg.widgets.com/mrp2.widgets.com/widgets.com*

sets the DNS domain to *mfg.widgets.com* and the search list to the three names.



Nslookup: set ti[meout]

Changes the initial number of seconds to wait for a reply to a request. When a reply to a request is not received within this time period, the timeout is doubled and the request is resent. (The number of retries is controlled with the **set retry** option.)

set ti[meout]=*number*

Default = 5.

Parameters

number

New value for the number of seconds to wait for a reply.



Nslookup: set ty[pe]

Changes the type of information query. More information about types can be found in Request For Comment (RFC) 1035. (The **set type** command is a synonym for **set querytype**.)

set ty[pe]=value

Default = A.

Parameters

See **set q[ueyrytype]**.



Nslookup: set q[ueyrytype]



Nslookup: set [no]v[c]

If set, always use a virtual circuit when sending requests to the server.

set [no]v[c]

Default = novc.



Nslookup: view

Sorts and lists the output of previous **ls** command(s).

view *filename*

Parameters

filename

Name of the file containing output from the previous **ls** command(s).



Nslookup Notes

Interactive Commands

- 1 To interrupt interactive commands at any time, type **CTRL+C**.
- 2 To exit, type **exit**.
- 3 The command line length must be less than 256 characters.
- 4 To treat a built-in command as a computer name, precede it with the escape character (\).
- 5 An unrecognized command is interpreted as a computer name.

Diagnostics

If the lookup request fails, an error message prints. Possible errors are:



Timed out

The server did not respond to a request after a certain amount of time (changed with **set timeout=value**) and a certain number of retries (changed with **set retry=value**).



No response from server

No DNS name server is running on the server computer.



No records

The DNS name server does not have resource records of the current query type for the computer, although the computer name is valid. The query type is specified with the **set querytype** command.



Non-existent domain

The computer or DNS domain name does not exist.



Connection refused

-or-

Network is unreachable

The connection to the DNS name server or finger server could not be made. This error commonly occurs with **ls** and **finger** requests.



Server failure

The DNS name server found an internal inconsistency in its database and could not return a valid answer.



Refused

The DNS name server refused to service the request.



Format error

The DNS name server found that the request packet was not in the proper format. It may indicate an error in **nslookup**.

References

For in-depth coverage of **nslookup**, see *DNS and BIND* by Paul Albitz and Cricket Liu, published by O'Reilly and Associates.

More Information About Nslookup



[Nslookup](#)



Ping

Verifies connections to a remote computer or computers. This command is available only if the TCP/IP protocol has been installed.

ping [-t] [-a] [-n *count*] [-l *length*] [-f] [-i *ttl*] [-v *tos*] [-r *count*] [-s *count*] [[-j *computer-list*] | [-k *computer-list*]] [-w *timeout*] *destination-list*

Parameters

-t

Pings the specified computer until interrupted.

-a

Resolve addresses to computer names.

-n *count*

Sends the number of ECHO packets specified by *count*. The default is 4.

-l *length*

Sends ECHO packets containing the amount of data specified by *length*. The default is 64 bytes; the maximum is 8192.

-f

Sends a Do not Fragment flag in the packet. The packet will not be fragmented by gateways on the route.

-i *ttl*

Sets the Time To Live field to the value specified by *ttl*.

-v *tos*

Sets the Type Of Service field to the value specified by *tos*.

-r *count*

Records the route of the outgoing packet and the returning packet in the Record Route field. A minimum of **1** and a maximum of **9** computers may be specified by *count*.

-s *count*

Specifies the timestamp for the number of hops specified by *count*.

-j *computer-list*

Routes packets via the list of computers specified by *computer-list*. Consecutive computers may be separated by intermediate gateways (loose source routed). The maximum number allowed by IP is 9.

-k *computer-list*

Routes packets via the list of computers specified by *computer-list*. Consecutive computers may not be separated by intermediate gateways (strict source routed). The maximum number allowed by IP is 9.

-w *timeout*

Specifies a timeout interval in milliseconds.

destination-list

Specifies the remote computers to ping.

More Information About Ping



[Ping--Notes](#)



Ping--Notes

The **ping** command verifies connections to remote computer or computers, by sending ICMP echo packets to the computer and listening for echo reply packets. **Ping** waits for up to 1 second for each packet sent and prints the number of packets transmitted and received. Each received packet is validated against the transmitted message. By default, four echo packets containing 64 bytes of data (a periodic uppercase sequence of alphabetic characters) are transmitted.

You can use the ping utility to test both the computer name and the IP address of the computer. If the IP address is verified but the computer name is not, you may have a name resolution problem. In this case, be sure that the computer name you are querying is in either the local HOSTS file or in the DNS database.

The following shows sample output for **ping**:

```
C:\>ping ds.internic.net
```

```
Pinging ds.internic.net [192.20.239.132] with 32 bytes of data:
```

```
Reply from 192.20.239.132: bytes=32 time=101ms TTL=243
```

```
Reply from 192.20.239.132: bytes=32 time=100ms TTL=243
```

```
Reply from 192.20.239.132: bytes=32 time=120ms TTL=243
```

```
Reply from 192.20.239.132: bytes=32 time=120ms TTL=243
```

More Information About Ping



[Ping](#)



Rcp

This connectivity command copies files between a Windows NT computer and a system running **rshd**, the remote shell daemon. The **rcp** command can also be used for third-party transfer to copy files between two computers running **rshd** when the command is issued from a Windows NT computer. The **rshd** daemon is available on UNIX computers, but not on Windows NT, so the Windows NT computer can only participate as the system from which the commands are issued. The remote computer must also provide the **rcp** utility in addition by running **rshd**.

rcp [-a | -b] [-h] [-r] *source1 source2 ... sourceN destination*

Parameters

-a

Specifies ASCII transfer mode. This mode converts the carriage return/linefeed characters to carriage returns on outgoing files and linefeed characters to carriage return/linefeeds for incoming files. This is the default transfer mode.

-b

Specifies binary image transfer mode. No carriage return/linefeed conversion is performed.

-h

Transfers source files marked with the hidden attribute on the Windows NT computer. Without this option, specifying a hidden file on the **rcp** command line has the same effect as if the file did not exist.

-r

Recursively copies the contents of all subdirectories of the source to the destination. Both the *source* and *destination* must be directories, although using **-r** will work even if the source is not a directory. There just will be no recursion.

source and destination

Must be of the form [*computer*[.*user*]:]*filename*. If the [*computer*[.*user*]:] portion is omitted, the computer is assumed to be the local computer. If the *user* portion is omitted, the currently logged on Windows NT username is used. If a fully qualified computer name is used, which contains the period (.) separators, then the [*.user*] must be included. Otherwise, the last part of the computer name will be interpreted as the username. If multiple source files are specified, the *destination* must be a directory.

If the filename does not begin with a forward slash (/) for UNIX or a backward slash (\) for Windows NT systems, it is assumed to be relative to the current working directory. On Windows NT, this is the directory from which the command is issued. On the remote system, it is the logon directory for the remote user. A period (.) means the current directory. Use the escape characters (\ , " , or ') in remote paths to use wildcard characters on the remote computer.

More Information About Rcp



[Rcp--Notes](#)



[Rcp--Examples](#)



Rcp--Notes

Remote privileges

The **rcp** command does not prompt for passwords; the current or specified user name must exist on the remote computer and allow remote command execution via **rcp**.

The .rhosts File

The `.rhosts` file specifies which remote system or users can access a local account using **rsh** or **rcp**. This file (or a HOSTS equivalent) is required for access to a remote system using these commands. **Rsh** and **rcp** both transmit the local username to the remote system. The remote system uses this name plus the IP address (usually resolved to a computer name) or the requesting system to determine whether access is granted. There is no provision for specifying a password to access an account using these commands.

If the user is logged on to a Windows NT Server domain, the primary domain controller must be available to resolve the currently logged on name, because the logged on name is not cached on the local computer. Because the username is required as part of the **rsh** protocol, the command will fail if the username cannot be obtained.

The `.rhosts` file is a text file where each line is an entry. An entry consists of the local computer name, the local user name, and any comments about the entry. Each entry is separated by a tab or space, and comments begin with a hash mark (`#`), for example:

```
computer5    marie    #This computer is in room 31A
```

The `.rhosts` file must be in the user's home directory on the remote computer. For more information about a remote computer's specific implementation of the `.rhosts` file, see the remote system's documentation.

Additionally, have your computer name added to the remote system's `/ETC/HOSTS` file. This will allow the remote system to authenticate remote requests for your computer using the Microsoft TCP/IP utilities.

Specifying Computers (Hosts)

Use the `computer.user` variables to use a user name other than the current user name. If `computer.user` is specified with `source`, the `.rhosts` file on the remote computer must contain an entry for `user`. For example,

```
rcp rhino.johnb:file1  buffalo.admin:file2
```

The `.rhosts` file on BUFFALO should have an entry for Johnb on RHINO.

If a computer name is supplied as a full domain name containing dots, a user name must be appended to the computer name, as previously described. This prevents the last element of the domain name from being interpreted as a user name. For example,

```
rcp domain-name1.user:johnm  domain-name2.user:billr
```

Remote Processing

Remote processing is performed by a command run from the user's logon shell on most UNIX systems. The user's `.profile` or `.cshrc` is executed before parsing filenames, and exported shell variables may be used (using the escape character or quotation marks) in remote filenames.

Copying Files

If you attempt to copy a number of files to a file rather than a directory, only the last file is copied. Also, the **rcp** command cannot copy a file onto itself.

If you logged on the Windows NT computer using a domain other than the local domain, and the primary domain controller is unavailable, the command will fail because **rcp** cannot determine the local user name. The same restriction applies to **rsh**.

More Information About Rcp



[Rcp](#)



[Rcp--Examples](#)



Rcp--Examples

These examples show syntax for some common uses of **rcp**.

To copy a local file to the logon directory of a remote computer:

```
rcp filename remotecomputer:
```

To copy a local file to an existing directory and new filename on a remote computer:

```
rcp filename remotecomputer:/directory/newfilename
```

To copy multiple local files to a subdirectory of a remote logon directory:

```
rcp file1 file2 file3 remotecomputer:subdirectory/filesdirectory
```

To copy from a remote source to the current directory of the local computer:

```
rcp remotecomputer:filename .
```

To copy from multiple files from multiple remote sources to a remote destination with different usernames:

```
rcp remote1.user1:file1 remote2.user2:file2 remotedest.destuser:directory
```

To copy from a remote system using an IP address to a local computer (where the username is mandatory because a period is used in the remote system name):

```
rcp 11.101.12.1.user:filename filename
```



Rexec

Runs commands on remote computers running the REXEC service. **Rexec** authenticates the user name on the remote computer before executing the specified command. This command is available only if the TCP/IP protocol has been installed.

rexec *computer* [-**l** *username*] [-**n**] *command*

Parameters

computer

Specifies the remote computer on which to run *command*.

-**l** *username*

Specifies the user name on the remote computer.

-**n**

Redirects the input of **rexec** to NULL.

command

Specifies the command to run.

More Information About Rexec



[_Rexec--Notes](#)



Rexec--Notes

Standard Operation

Rexec prompts the user for a password and authenticates the given password on the remote computer. If the authentication succeeds, the command is executed.

Rexec copies standard input to the remote command, standard output to its standard output, and standard error to its standard error. Interrupt, quit, and terminate signals are propagated to the remote command. **Rexec** normally terminates when the remote command does.

Using Redirection Symbols

Use quotation marks around redirection symbols to redirect onto the remote computer. If quotation marks are not used, redirection occurs on the local computer. For example, the following command appends the remote file *remotefile* to the local file *localfile*:

```
rexec othercomputer cat remotefile >> localfile
```

The following command appends the remote file *remotefile* to the remote file *otherremotefile*:

```
rexec othercomputer cat remotefile ">>" otherremotefile
```

Using Interactive Commands

You cannot run most interactive commands. For example, **vi** or **emacs** cannot be run using **rexec**. Use **telnet** to run interactive commands.

More Information About Rexec



[_Rexec](#)



Route

Manipulates network routing tables. This command is available only if the TCP/IP protocol has been installed.

route [-f] [-p] [*command* [*destination*] [**mask** *subnetmask*] [*gateway*] [**metric** *costmetric*]]

Parameters

-f

Clears the routing tables of all gateway entries. If this is used in conjunction with one of the commands, the tables are cleared prior to running the command.

-p

When used with the ADD command, makes a route persistent across boots of the system. By default, routes are not preserved when the system is restarted. When used with the PRINT command, displays the list of registered persistent routes. Ignored for all other commands, which always affect the appropriate persistent routes.

command

Specifies one of four commands

Command	Purpose
print	Prints a route
add	Adds a route
delete	Deletes a route
change	Modifies an existing route

destination

Specifies the computer to send *command*.

mask *subnetmask*

Specifies a subnet mask to be associated with this route entry. If not specified, 255.255.255.255 is used.

gateway

Specifies gateway.

All symbolic names used for *destination* or *gateway* are looked up in the network and computer name database files NETWORKS and HOSTS, respectively. If the command is **print** or **delete**, wildcards may be used for the destination and gateway, or the gateway argument may be omitted.

metric *costmetric*

Assigns an integer cost metric (ranging from 1 to 9999) to be used in calculating the fastest, most reliable, and/or least expensive routes.



Rsh

Runs commands on remote computers running the RSH service. This command is available only if the TCP/IP protocol has been installed.

rsh *computer* [-l *username*] [-n] *command*

Parameters

computer

Specifies the remote computer on which to run *command*.

-l *username*

Specifies the user name to use on the remote computer. If omitted, the logged on user name is used.

-n

Redirects the input of rsh to NULL.

command

Specifies the command to run.

More Information About Rsh



[Rsh--Notes](#)



Rsh--Notes

Standard operation

Rsh copies standard input to the remote *command*, standard output of the remote *command* to its standard output, and the standard error of the remote *command* to its standard error. **Rsh** normally terminates when the remote command does.

Using redirection symbols

Use quotation marks around redirection symbols to redirect onto the remote computer. If quotation marks are not used, redirection occurs on the local computer. For example, the following command appends the remote file *remotefile* to the local file *localfile*:

```
rsh othercomputer cat remotefile >> localfile
```

The following command appends the remote file *remotefile* to the remote file *otherremotefile*:

```
rsh othercomputer cat remotefile ">>" otherremotefile
```

Using rsh

When using a computer running Windows NT Server, the primary domain controller for the logged on domain must be available to verify the user name or the **rsh** command will fail.

The .rhosts File

The *.rhosts* file generally permits network access rights on UNIX systems. The *.rhosts* file lists computer names and associated logon names that have access to remote computers. When issuing **rcp**, **rexec**, or **rsh** commands to a remote system with a properly configured *.rhosts* file, you do not need to provide logon and password information for the remote computer.

The *.rhosts* file is a text file where each line is an entry. An entry consists of the local computer name, the local user name, and any comments about the entry. Each entry is separated by a tab or space, and comments begin with a hash mark (#), for example:

```
computer5    marie    #This computer is in room 31A
```

The *.rhosts* file must be in the user's home directory on the remote computer. For further information about a remote computer's specific implementation of the *.rhosts* file, see the remote system's documentation.

More Information About Rsh



[Rsh](#)



Tftp

Transfers files to and from a remote computer running the TFTP service. This command is available only if the TCP/IP protocol has been installed.

tftp [-i] *computer* [**get** | **put**] *source* [*destination*]

Parameters

-i

Specifies binary image transfer mode (also called octet). In binary image mode the file is moved literally, byte by byte. Use this mode when transferring binary files.

If **-i** is omitted, the file is transferred in ASCII mode. This is the default transfer mode. This mode converts the EOL characters to a carriage return for UNIX and a carriage return/line feed for personal computers. This mode should be used when transferring text files. If a file transfer is successful, the data transfer rate is displayed.

computer

Specifies the local or remote computer.

put

Transfers the file *destination* on the local computer to the file *source* on the remote computer.

get

Transfers the file *destination* on the remote computer to the file *source* on the local computer.

Specify **put** if transferring file *file-two* on the local computer to file *file-one* on remote computer. Specify **get** if transferring file *file-two* on the remote computer to file *file-one* on the remote computer.

If the local file is -, the remote file is printed out on *stdout* (if getting), or is read from *stdin* (if putting). If *file-two* is omitted, it is assumed to have the same name as *file-one*.

Since the **tftp** protocol does not support user authentication, the user must be logged on, and the files must be writable on the remote computer.

source

Specifies the file to transfer.

destination

Specifies where to transfer the file.



Tracert

tracert [-d] [-h *maximum_hops*] [-j *computer-list*] [-w *timeout*] *target_name*

This diagnostic utility determines the route taken to a destination by sending Internet Control Message Protocol (ICMP) echo packets with varying Time-To-Live (TTL) values to the destination. Each router along the path is required to decrement the TTL on a packet by at least 1 before forwarding it, so the TTL is effectively a hop count. When the TTL on a packet reaches 0, the router is supposed to send back an ICMP Time Exceeded message to the source system. **Tracert** determines the route by sending the first echo packet with a TTL of 1 and incrementing the TTL by 1 on each subsequent transmission until the target responds or the maximum TTL is reached. The route is determined by examining the ICMP Time Exceeded messages sent back by intermediate routers. Notice that some routers silently drop packets with expired time-to-live (TTLs) and will be invisible to **tracert**.

Parameters

-d

Specifies not to resolve addresses to computer names.

-h *maximum_hops*

Specifies maximum number of hops to search for target.

-j *computer-list*

Specifies loose source route along *computer-list*.

-w *timeout*

Waits the number of milliseconds specified by *timeout* for each reply.

target_name

Name of the target computer.



Debug

Starts Debug, a program that allows you to test and debug MS-DOS executable files.

debug [[drive:][path] filename [testfile-parameters]]

Parameters

[drive:][path] filename

Specifies the location and name of the executable file you want to test.

testfile-parameters

Specifies any command-line information required by the executable file you want to test.

More Information About Debug



[Debug--Notes](#)



[Debug Commands](#)



Debug--Notes

Using the debug command without specifying a file to be tested

If you use the **debug** command without a location and filename, you then type all Debug commands in response to the Debug prompt, a hyphen (-).

Debug commands

The following is a list of Debug commands.

- ? Displays a list of the Debug commands.
- a Assembles 8086/8087/8088 mnemonics.
- c Compares two portions of memory.
- d Displays the contents of a portion of memory.
- e Enters data into memory starting at a specified address.
- f Fills a range of memory with specified values.
- g Runs the executable file that is in memory.
- h Performs hexadecimal arithmetic.
- i Displays one byte value from a specified port.
- l Loads the contents of a file or disk sectors into memory.
- m Copies the contents of a block of memory.
- n Specifies a file for an **l** or **w** command, or specifies the parameters for the file you are testing.
- o Sends one byte value to an output port.
- p Executes a loop, a repeated string instruction, a software interrupt, or a subroutine.
- q Stops the Debug session.
- r Displays or alters the contents of one or more registers.
- s Searches a portion of memory for a specified pattern of one or more byte values.
- t Executes one instruction and

then displays the contents of all registers, the status of all flags, and the decoded form of the instruction that Debug will execute next.

- u Disassembles bytes and displays the corresponding source statements.
- w Writes the file being tested to a disk.
- xa Allocates expanded memory.
- xd Deallocates expanded memory.
- xm Maps expanded memory pages.
- xs Displays the status of expanded memory.

Separating command parameters

All Debug commands accept parameters, except the **q** command. You can separate parameters with commas or spaces, but these separators are required only between two hexadecimal values. Therefore, the following commands are equivalent:

```
dcs:100 110  
d cs:100 110  
d,cs:100,110
```

Specifying valid address entries

An *address* parameter in a Debug command specifies a location in memory. *Address* is a two-part designation containing either an alphabetic segment register or a 4-digit segment address, plus an offset value. You can omit the segment register or segment address. The default segment for the **a**, **g**, **l**, **t**, **u**, and **w** commands is CS. The default segment for all other commands is DS. All numeric values are in hexadecimal format.

The following are valid addresses:

```
CS:0100  
04BA:0100
```

The colon between the segment name and the offset value is required.


Specifying valid range entries

A *range* parameter in a Debug command specifies a range of memory. You can choose from two formats for *range*: a starting address and an ending address, or a starting address and the length (denoted by **l**) of the range.

For example, both of the following syntaxes specify a 16-byte range beginning at CS:100:

```
cs:100 10f  
cs:100 l 10
```

More Information About Debug

 [Debug Commands](#)

 [Debug](#)



Debug: A (Assemble)

Assembles 8086/8087/8088 mnemonics directly into memory.

This command creates executable machine code from assembly-language statements. All numeric values are in hexadecimal format, and you must type them as 1 to 4 characters. You specify a prefix mnemonic in front of the operation code (opcode) to which it refers.

a [*address*]

Parameters

address

Specifies the location where you type assembly-language mnemonics. You use hexadecimal values for *address* and type each value without the trailing "h" character. If you do not specify an address, **a** starts assembling where it last stopped.

See Also

For information about entering data into specific bytes, see the [Debug E \(Enter\)](#) command.

For information about disassembling bytes, see the [Debug U \(Unassemble\)](#) command.

More Information About Debug: A



[Debug: A--Notes](#)



[Debug: A--Examples](#)



Debug: A--Notes

Using mnemonics

The segment-override mnemonics are **cs:**, **ds:**, **es:**, and **ss:**. The mnemonic for the far return is **retf**. String-manipulation mnemonics must explicitly state the string size. For example, use **movsw** to move word strings (16 bits), and use **movsb** to move byte strings (8 bits).

Assembling jumps and calls

The assembler automatically assembles a short, near, or far jump or call, depending on byte displacement, to the destination address. You can override such a jump or call by using a **near** or **far** prefix, as the following example shows:

```
-a0100:0500
0100:0500 jmp 502 ; a 2-byte short jump
0100:0502 jmp near 505 ; a 3-byte near jump
0100:0505 jmp far 50a ; a 5-byte far jump
```

You can abbreviate the **near** prefix to **ne**.

Distinguishing word and byte memory locations

When an operand can refer to either a word memory location or a byte memory location, you must specify the data type with the prefix **word ptr** or the prefix **byte ptr**. Acceptable abbreviations are **wo** and **by**, respectively. The following example shows the two formats:

```
dec wo [si]
neg by ptr [128]
```

Specifying operands

Debug uses the common convention that an operand enclosed in brackets ([]) refers to a memory location. This is because Debug cannot otherwise differentiate between an immediate operand and an operand that is a memory location. The following example shows the two formats:

```
mov ax,21 ; load AX with 21h
mov ax,[21] ; load AX with the
; contents of
; memory location 21h
```

Using pseudoinstructions

Two popular pseudoinstructions are available with the **a** command: the **db** opcode, which assembles byte values directly into memory, and the **dw** opcode, which assembles word values directly into memory. Following are examples of both pseudoinstructions:

```
db 1,2,3,4,"THIS IS AN EXAMPLE"
db 'THIS IS A QUOTATION MARK: "'
db "THIS IS A QUOTATION MARK: '"
dw 1000,2000,3000,"BACH"
```

More Information About Debug: A



[Debug: A--Examples](#)



[Debug: A](#)



Debug: A--Examples

The **a** command supports all forms of register-indirect commands, as the following example shows:

```
add    bx,34[bp+2].[si-1]
pop    [bp+di]
push   [si]
```

All opcode synonyms are also supported, as the following example shows:

```
loopz  100
loope  100
ja     200
jnbe   200
```

For 8087 opcodes, you must specify the **wait** or **fwait** prefix, as the following example shows:

```
fwait fadd st,st(3)    ; this line assembles
                        ; an fwait prefix
```

More Information About Debug: A



[Debug: A--Notes](#)



[Debug: A](#)



Debug: C (Compare)

Compares two portions of memory.

c *range address*

Parameters

range

Specifies the starting and ending addresses, or the starting address and length, of the first area of memory you want to compare. For information about valid *range* values, see [Debug--Notes](#).

address

Specifies the starting *address* of the second area of memory you want to compare. For information about valid *address* values, see [Debug--Notes](#)

More Information About Debug: C



[Debug: C--Note](#)



[Debug: C--Example](#)



Debug: C--Note

If the *range* and *address* memory areas are identical, Debug displays nothing and returns directly to the Debug prompt. If there are differences, Debug displays them in the following format:

address1 byte1 byte2 address2

More Information About Debug: C



[Debug: C--Example](#)



[Debug: C](#)



Debug: C--Example

The following commands have the same effect:

```
c100,10f 300  
c100!10 300
```

Each command compares the block of memory from 100h through 10Fh with the block of memory from 300h through 30Fh.

Debug responds to either of the previous commands with a display similar to the following (assuming DS = 197F):

```
197F:0100 4D E4 197F:0300  
197F:0101 67 99 197F:0301  
197F:0102 A3 27 197F:0302  
197F:0103 35 F3 197F:0303  
197F:0104 97 BD 197F:0304  
197F:0105 04 35 197F:0305  
197F:0107 76 71 197F:0307  
197F:0108 E6 11 197F:0308  
197F:0109 19 2C 197F:0309  
197F:010A 80 0A 197F:030A  
197F:010B 36 7F 197F:030B  
197F:010C BE 22 197F:030C  
197F:010D 83 93 197F:030D  
197F:010E 49 77 197F:030E  
197F:010F 4F 8A 197F:030F
```

Notice that the addresses 197F:0106 and 197F:0306 are missing from the list. This means that the values in those addresses are identical.

More Information About Debug: C



[Debug: C--Note](#)



[Debug: C](#)



Debug: D (Dump)

Displays the contents of a range of memory addresses.

d [*range*]

Parameter

range

Specifies the starting and ending addresses, or the starting address and length, of the memory area whose contents you want to display. For information about valid *range* values, see [Debug--Notes](#). If you do not specify *range*, Debug displays the contents of 128 bytes, starting at the end of the address range specified in the previous **d** command.

See Also

For information about displaying the contents of registers, see the [Debug R \(Register\)](#) command.

More Information About Debug: D



[Debug: D--Note](#)




[Debug: D--Examples](#)




Debug: D--Note

When you use the **d** command, Debug displays memory contents in two portions: a hexadecimal portion (each byte value is shown in hexadecimal format) and an ASCII portion (each byte value is shown as an ASCII character). Each nonprinting character is denoted by a period (.) in the ASCII portion of the display. Each display line shows the contents of 16 bytes, with a hyphen between the eighth and ninth bytes. Each display line begins on a 16-byte boundary.

More Information About Debug: D

 [Debug: D--Examples](#)

 [Debug: D](#)



Debug: D--Examples

Suppose you type the following command:

```
dcs:100 10f
```

Debug displays the contents of the range in the following format:

```
04BA:0100 54 4F 4D 00 53 41 57 59-45 52 00 00 00 00 00 00 TOM.SAWYER.....
```

If you type the **d** command without parameters, Debug formats the display as described in the previous example. Each line of the display begins with an address that is 16 bytes greater than the address on the previous line (or 8 bytes if you have a 40-column screen).

For each subsequent **d** command you type without parameters, Debug displays the bytes immediately following those last displayed.

If you type the following command, Debug displays the contents of 20h bytes, starting at CS:100:

```
dcs:100 l 20
```

If you type the following command, Debug displays the contents of all bytes in the range of lines from 100h through 115h in the CS segment:

```
dcs:100 115
```

More Information About Debug: D



[Debug: D--Note](#)



[Debug: D](#)



Debug: E (Enter)

Enters data into memory at the address you specify.

You can type data in either hexadecimal or ASCII format. Any data previously stored at the specified address is lost.

e *address* [*list*]

Parameters

address

Specifies the first memory location where you want to enter data.

list

Specifies the data you want to enter into successive bytes of memory.

See Also

For information about assembling mnemonics, see the [Debug A \(Assemble\)](#) command.

For information about displaying the contents of a portion of memory, see the [Debug D \(Dump\)](#) command.

More Information About Debug: E



[Debug: E--Notes](#)







[Debug: E--Examples](#)



Debug: E--Notes

Using the *address* parameter

If you specify a value for *address* without specifying a value for the optional *list* parameter, Debug displays the address and its contents, repeats the address on the next line, and waits for your input. At this point, you can perform one of the following actions:


-  Replace the byte value. To do this, you type a new value after the current value. If the value you type is not a valid hexadecimal value or if it contains more than two digits, Debug does not echo the invalid or extra character.
-  Advance to the next byte. To do this, you press the **SPACEBAR**. To change the value in that byte, type a new value after the current value. If you move beyond an 8-byte boundary when you press the **SPACEBAR**, Debug starts a new display line and displays the new address at the beginning of the line.
-  Return to the preceding byte. To do this, you press the **HYPHEN** key. You can press the **HYPHEN** key repeatedly to move back more than 1 byte. When you press **HYPHEN**, Debug starts a new line and displays the current address and byte value.
-  Stop the **e** command. To do this, you press the ENTER key. You can press ENTER at any byte position.

Using the *list* parameter

If you specify values for the *list* parameter, the **e** command sequentially replaces the existing byte values with the values from the list. If an error occurs, no byte values are changed.

List values can be either hexadecimal byte values or strings. You separate values by using a space, a comma, or a tab character. You must enclose strings within single or double quotation marks.

More Information About Debug: E

 [Debug: E--Examples](#)

 [Debug: E](#)



Debug: E--Examples

Suppose you type the following command:

```
ecs:100
```

Debug displays the contents of the first byte in the following format:

```
04BA:0100 EB._
```

To change this value to 41, type **41** at the cursor, as follows:

```
04BA:0100 EB.41_
```

You can type consecutive byte values with one **e** command. Instead of pressing ENTER after typing the new value, press the SPACEBAR. Debug displays the next value. In this example, if you press the SPACEBAR three times, Debug displays the following values:

```
04BA:0100 EB.41 10. 00. BC._
```

To change the hexadecimal value BC to 42, type **42** at the cursor, as follows:

```
04BA:0100 EB.41 10. 00. BC.42_
```

Now suppose that you decide the value 10 should be 6F. To correct this value, press the HYPHEN key twice to return to address 0101 (value 10). Debug displays the following:

```
04BA:0100 EB.41 10. 00. BC.42-
```

```
04BA:0102 00.-
```

```
04BA:0101 10._
```

Type **6f** at the cursor to change the value, as follows:

```
04BA:0101 10.6f_
```

Press **ENTER** to stop the **e** command and return to the Debug prompt.

The following is an example of a string entry:

```
eds:100 "This is the text example"
```

This string will fill 24 bytes, starting at DS:100.

More Information About Debug: E



[Debug: E--Notes](#)



[Debug: E](#)



Debug: F (Fill)

Fills addresses in the specified memory area with values you specify.

You can specify data in either hexadecimal or ASCII format. Any data you previously stored at the specified address is lost.

f *range list*

Parameters

range

Specifies the starting and ending addresses, or the starting address and length, of the memory area you want to fill. For information about valid *range* values, see [Debug--Notes](#).

list

Specifies the data you want to enter. *List* can consist of hexadecimal numbers or a string enclosed in quotation marks.

More Information About Debug: F



[Debug: F--Notes](#)



[Debug: F--Example](#)



Debug: F--Notes

Using the *range* parameter

If *range* contains more bytes than the number of values in *list*, Debug assigns the values in *list* repeatedly until all bytes in *range* are filled.

If any of the memory in *range* is bad or doesn't exist, Debug displays an error message and stops the **f** command.

Using the *list* parameter

If *list* contains more values than the number of bytes in *range*, Debug ignores the extra values in *list*.

More Information About Debug: F



[Debug: F--Example](#)



[Debug: F](#)



Debug: F--Example

Suppose you type the following command:

```
f04ba:100i100 42 45 52 54 41
```

In response, Debug fills memory locations 04BA:100 through 04BA:1FF with the values specified. Debug repeats the five values until all the 100h bytes are filled.

More Information About Debug: F



[Debug: F--Notes](#)



[Debug: F](#)



Debug: G (Go)

Runs the program currently in memory.

g [=address] [breakpoints]

Parameters

=address

Specifies the address in the program currently in memory at which you want execution to begin. If you do not specify *address*, Windows NT begins program execution at the current address in the CS:IP registers.

breakpoints

Specifies 1 to 10 temporary breakpoints that you can set as part of the **g** command.

See Also

For information about executing a loop, a repeated string instruction, a software interrupt, or a subroutine, see the [Debug P \(Proceed\)](#) command.

For information about executing one instruction, see the [Debug T \(Trace\)](#) command.

More Information About Debug: G



[Debug: G--Notes](#)



[Debug: G--Examples](#)



Debug: G--Notes

Using the *address* parameter

You must precede the *address* parameter with an equal sign (=) to distinguish the starting address (*address*) from the breakpoint addresses (*breakpoints*).

Specifying breakpoints

The program stops at the first breakpoint it encounters, regardless of where you typed that breakpoint in the *breakpoints* list. Debug replaces the original instruction at each breakpoint with an interrupt code.

When the program reaches a breakpoint, Debug restores all breakpoint addresses to their original instructions and displays the contents of all registers, the status of all flags, and the decoded form of the last instruction executed. Debug displays the same information as it would display if you used the Debug **r** (register) command and specified the breakpoint address.

If you do not stop the program at one of the breakpoints, Debug does not replace the interrupt codes with the original instructions.

Limitations on setting breakpoints

You can set breakpoints only at addresses containing the first byte of an 8086 operation code (opcode). If you set more than 10 breakpoints, Debug displays the following message:

bp Error

Requirements for the user stack pointer

The user stack pointer must be valid and must have 6 bytes available for the **g** command. This command uses an **iret** instruction to jump to the program being tested. Debug sets the user stack pointer and pushes the user flags, the code segment register, and the instruction pointer onto the user stack. (If the user stack is not valid or is too small, the operating system might fail.) Debug places an interrupt code (0CCh) at the specified breakpoint address(es).

Restarting a program

Do not attempt to restart a program after Windows NT displays the following message:

Program terminated normally

To run the program properly, you must reload it by using the Debug **n** (name) and **l** (load) commands.

More Information About Debug: G



[Debug: G--Examples](#)



[Debug: G](#)



Debug: G--Examples

Suppose you type the following command:

```
gcs:7550
```

Windows NT runs the program currently in memory up to the breakpoint address 7550 in the CS segment. Debug then displays the contents of the registers and the status of the flags and stops the **g** command.

The following command sets two breakpoints:

```
gcs:7550, cs:8000
```

If you type the **g** command again after Debug encounters a breakpoint, execution begins at the instruction after the breakpoint, rather than at the usual starting address.

More Information About Debug: [G](#)



[Debug: G--Notes](#)



[Debug: G](#)

**Debug: H (Hex)**

Performs hexadecimal arithmetic on two parameters you specify.

h *value1 value2*

Parameters

value1

Represents any hexadecimal number in the range 0 through FFFFh.

value2

Represents a second hexadecimal number in the range 0 through FFFFh.

More Information About Debug: H



[Debug: H--Note](#)




[Debug: H--Example](#)




Debug: H--Note

Debug first adds the two parameters you specify and then subtracts the second parameter from the first. The results of these calculations are displayed on one line--first the sum, then the difference.

More Information About Debug: H

 [Debug: H--Example](#)

 [Debug: H](#)



Debug: H--Example

Suppose you type the following command:

```
h19f 10a
```

Debug performs the calculations and displays the following result:

```
02A9 0095
```

More Information About Debug: H



[Debug: H--Note](#)



[Debug: H](#)



Debug: I (Input)

Reads and displays one byte value from the port you specify.

i *port*

Parameters

port

Specifies the input port by address. The address can be a 16-bit value.

See Also

For information about sending the value of a byte to an output port, see the [Debug O \(Output\)](#) command.

More Information About Debug: I



[Debug: I--Example](#)



Debug: I--Example

Suppose you type the following command:

```
i2f8
```

Suppose also that the byte value at the port is 42h. Debug reads the byte and then displays the value, as follows:

```
42
```

More Information About Debug: I



[Debug: I](#)



Debug: L (Load)

Loads a file or the contents of specific disk sectors into memory.

To load the contents of the number of bytes specified in the BX:CX registers from a disk file, use the following syntax:

I [*address*]

To bypass the Windows NT file system and directly load specific sectors, use the following syntax:

I *address drive start number*

Parameters

address

Specifies the memory location where you want to load the file or the sector contents. If you do not specify *address*, Debug uses the current address in the CS register.

drive

Specifies the drive that contains the disk from which specific sectors are to be read. This value is numeric: 0 = A, 1 = B, 2 = C, and so on. You use the *drive*, *start*, and *number* parameters only if you want to load the contents of specific sectors rather than load the file specified on the **debug** command line or in the most recent **Debug n** (name) command.

start

Specifies the hexadecimal number of the first sector whose contents you want to load.

number

Specifies the hexadecimal number of consecutive sectors whose contents you want to load.

See Also

For information about specifying a file for the **I** command, see the [Debug N \(Name\)](#) command.

For information about writing the file being debugged to a disk, see the [Debug W \(Write\)](#) command.

More Information About Debug: L



[Debug: L--Notes](#)



[Debug: L--Examples](#)



Debug: L--Notes

Using the I command without parameters

When you use the **I** command without parameters, the file you specified on the **debug** command line is loaded into memory, beginning at address CS:100. Debug also sets the BX and CX registers to the number of bytes loaded. If you did not specify a file on the **debug** command line, the file loaded is the one you most recently specified by using the **n** command.

Using the I command with the address parameter

If you use the **I** command with the *address* parameter, Debug begins loading the file or the contents of the specified sectors at the memory location *address*.

Using the I command with all parameters

If you use the **I** command with all parameters, Debug loads the contents of specific disk sectors instead of loading a file.

Loading the contents of specific sectors

Each sector in the range you specify is read from *drive*. Debug begins loading with *start* and continues until the contents of the number of sectors specified in *number* have been loaded.

Loading an .EXE file

Debug ignores the *address* parameter for .EXE files. If you specify an .EXE file, Debug relocates the file to the loading address specified in the header of the .EXE file. The header itself is stripped off the .EXE file before the file is loaded into memory, so the size of an .EXE file on disk differs from its size in memory. If you want to examine a complete .EXE file, rename the file with a different extension.

Opening a hex file

A hex file is a file that uses the Intel hexadecimal format, as described in *The MS-DOS Encyclopedia* (Microsoft Press, 1988). Debug assumes that files with the .HEX extension are hexadecimal-format files. You can type the **I** command with no parameters to load a hex file beginning at the address specified in the hex file. If the **I** command you type includes the *address* parameter, Debug adds the specified address to the address found in the hex file to determine the starting address.

More Information About Debug: L



[Debug: L--Examples](#)



[Debug: L](#)



Debug: L--Examples

Suppose you start Debug and type the following command:

```
nfile.com
```

You can now type the **I** command to load FILE.COM. Debug loads the file and displays the Debug prompt.

Suppose that you want to load the contents of 109 (6Dh) sectors from drive C, beginning with logical sector 15 (0Fh), into memory beginning at address 04BA:0100. To do this, type the following command:

```
l04ba:100 2 0f 6d
```

More Information About Debug: L



[Debug: L--Notes](#)



[Debug: L](#)



Debug: M (Move)

Copies the contents of a block of memory to another block of memory.

m *range address*

Parameters

range

Specifies the starting and ending addresses, or the starting address and the length, of the memory area whose contents you want to copy.

address

Specifies the starting address of the location to which you want to copy the contents of *range*.

More Information About Debug: M



[Debug: M--Notes](#)



[Debug: M--Example](#)



Debug: M--Notes


Effects of the copy operation on existing data


If the addresses in the block being copied do not have new data written to them, the original data remains intact. However, if the destination block already contains data (as it might in an overlapping copy operation), that data is overwritten. (Overlapping copy operations are those in which part of the destination block overlaps part of the source block.)

Performing overlapping copy operations

The **m** command performs overlapping copy operations without losing data at the destination addresses. The contents of addresses that will be overwritten are copied first. Thus, if data is to be copied from higher addresses to lower addresses, the copy operation begins at the source block's lowest address and progresses toward the highest address. Conversely, if data is to be copied from lower addresses to higher addresses, the copy operation begins at the source block's highest address and progresses toward the lowest address.

More Information About Debug: M

 [Debug: M--Example](#)

 [Debug: M](#)



Debug: M--Example

Suppose you type the following command:

```
mcs:100 110 cs:500
```

Debug first copies the contents of address CS:110 to CS:510, then copies the contents of CS:10F to CS:50F, and so on until it has copied the contents of CS:100 to CS:500. To view the results, you can use the Debug **d** (dump) command, specifying the destination address you used with the **m** command.

More Information About Debug: M



[Debug: M--Notes](#)



[Debug: M](#)



Debug: N (Name)

Specifies the name of an executable file for a Debug **l** (load) or **w** (write) command, or specifies parameters for the executable file being debugged.

n [*drive:*][*path*] *filename*

To specify parameters for the executable file you are testing, use the following syntax:

n *file-parameters*

Parameters

Used without parameters, the **n** command clears the current specifications.

[*drive:*][*path*] *filename*

Specifies the location and name of the executable file you want to test.

file-parameters

Specifies parameters and switches for the executable file you are testing.

See Also

For information about loading the contents of a file or of specific disk sectors into memory, see the [Debug L \(Load\)](#) command.

For information about writing the file being debugged to a disk, see the [Debug W \(Write\)](#) command.

More Information About Debug: N



[Debug: N--Notes](#)



[Debug: N--Examples](#)



Debug: N--Notes

The two uses of the n command

You can use the **n** command in two ways. First, you can use it to specify a file to be used by a later **l** (load) or **w** (write) command. If you start Debug without naming a file to be debugged, you must use the command **nfilename** before you can use the **l** command to load the file. The filename is correctly formatted for a file control block (FCB) at CS:5C. Second, you can use the **n** command to specify command-line parameters and switches for the file being debugged.

Memory areas

The following four areas of memory can be affected by the **n** command:

Memory location	Contents
CS:5C	File control block (FCB) for file 1
CS:6C	File control block (FCB) for file 2
CS:80	Length of n command line (in characters)
CS:81	Beginning of n command-line characters

The first filename you specify for the **n** command is placed in a FCB at CS:5C. If you specify a second filename, this name is placed in an FCB at CS:6C. The number of characters typed on the **n** command line (exclusive of the first character, **n**) is stored at location CS:80. The actual characters on the **n** command line (again, exclusive of the letter **n**) are stored beginning at CS:81. Note that these characters can be any switches and delimiters that would be legal in a command typed at the Windows NT prompt.

More Information About Debug: N



[Debug: N--Examples](#)



[Debug: N](#)



Debug: N--Examples

Suppose you've started Debug and loaded the program PROG.COM for debugging. You subsequently decide to specify two parameters for PROG.COM and run the program. Following is the sequence of commands for this example:

```
debug prog.com  
nparam1 param2  
g
```

In this case, the Debug **g** (go) command runs the program as if you had typed the following command at the Windows NT prompt:

```
prog param1 param2
```

Testing and debugging therefore reflect a typical run-time environment for PROG.COM.

In the following sequence of commands, the first **n** command specifies FILE1.EXE as the file for the subsequent **l** (load) command, which loads FILE1.EXE into memory. The second **n** command specifies the parameters to be used by FILE1.EXE. Finally, the **g** command runs FILE1.EXE as if you had typed **FILE1 FILE2.DAT FILE3.DAT** at the Windows NT prompt.

```
nfile1.exe  
l  
nfile2.dat file3.dat  
g
```

Note



You do not use the **l** command after the second form of the **n** command. Also note that if you now use the **w** (write) command, Windows NT saves FILE1.EXE, the file being debugged, with the name FILE2.DAT. To avoid this result, you should always use the first form of the **n** command immediately before either an **l** or a **w** command.

More Information About Debug: N



[Debug: N--Notes](#)



[Debug: N](#)



Debug: O (Output)

Sends the value of a byte to an output port.

o *port byte-value*

Parameters

port

Specifies the output port by address. The port address can be a 16-bit value.

byte-value

Specifies the byte value you want to direct to *port*.

See Also

For information about reading the value of a byte from an input port, see the [Debug I \(Input\)](#) command.

More Information About Debug: O



[Debug: O--Example](#)



Debug: O--Example

To send the byte value 4Fh to the output port at address 2F8h, type the following command:

```
o2f8 4f
```

More Information About Debug: O

 [Debug: O](#)



Debug: P (Proceed)

Executes a loop, a repeated string instruction, a software interrupt, or a subroutine; or traces through any other instruction.

p [=address] [number]

Parameters

=address

Specifies the location of the first instruction to execute. If you do not specify an address, the default address is the current address specified in the CS:IP registers.

number

Specifies the number of instructions to execute before returning control to Debug. The default value is 1.

See Also

For information about running the program currently in memory, see the [Debug G \(Go\)](#) command.

For information about executing one instruction, see the [Debug T \(Trace\)](#) command.

More Information About Debug: P



[Debug: P--Notes](#)



[Debug: P--Example](#)



Debug: P--Notes

Transferring control to the program being tested

When the **p** command transfers control from Debug to the program being tested, that program runs without interruption until the loop, repeated string instruction, software interrupt, or subroutine at the specified address is completed, or until the specified number of machine instructions have been executed. Control then returns to Debug.

Limitations on the address parameter

If the *address* parameter does not specify a segment, Debug uses the CS register of the program being tested. If you omit *address*, the program is executed beginning at the address specified by its CS:IP registers. You must precede the *address* parameter with an equal sign (=) to distinguish it from the *number* parameter. If the instruction at the specified address is not a loop, a repeated string instruction, a software interrupt, or a subroutine, the **p** command works the same way as the Debug **t** (trace) command.

Messages displayed with the p command

After **p** executes an instruction, Debug displays the contents of the program's registers, the status of its flags, and the decoded form of the next instruction to be executed.

Caution: You cannot use the **p** command to trace through read-only memory (ROM).

More Information About Debug: P



[Debug: P--Example](#)



[Debug: P](#)



Debug: P--Example

Suppose that the program you're testing contains a **call** instruction at address CS:143F. To run the subroutine that is the destination of **call** and then return control to Debug, type the following command:

```
p=143f
```

Debug displays the results in the following format:

```
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=2246 ES=2246 SS=2246 CS=2246 IP=1443 NV UP EI PL NZ AC PO NC
2246:1442 7505          JNZ    144A
```

More Information About Debug: P



[Debug: P--Notes](#)



[Debug: P](#)

**Debug: Q (Quit)**

Stops the Debug session, without saving the file currently being tested.

After you type **q**, control returns to the Windows NT command prompt.

q

Parameters

This command takes no parameters.

See Also

For information about saving a file, see the [Debug W \(Write\)](#) command.



Debug: R (Register)

Displays or alters the contents of one or more central-processing-unit (CPU) registers.

r [*register-name*]

Parameter

none

Used without parameters, the **r** command displays the contents of all registers and flags in the register storage area.

register-name

Specifies the name of the register whose contents you want to display.

See Also

For information about displaying the contents of a portion of memory, see the [Debug D \(Dump\)](#) command.

For information about disassembling bytes, see the [Debug U \(Unassemble\)](#) command.

More Information About Debug: R



[Debug: R--Notes](#)



[Debug: R--Examples](#)



Debug: R--Notes

Using the R command

If you specify a register name, Windows NT displays the 16-bit value of that register in hexadecimal notation and displays a colon as the prompt. If you want to change the value contained in the register, type a new value and press ENTER; otherwise, just press ENTER to return to the Debug prompt.

Valid register names

The following are valid values for *register-name*: **ax, bx, cx, dx, sp, bp, si, di, ds, es, ss, cs, ip, pc**, and **f**. Both **ip** and **pc** refer to the instruction pointer.

If you specify a register name other than one from the preceding list, Windows NT displays the following message:

br error

Using the f character instead of a register name

If you type the **f** character instead of a register name, Debug displays the current setting of each flag as a two-letter code and then displays the Debug prompt. To change the setting of a flag, type the appropriate two-letter code from the following table:

Flag name	Set	Clear
Overflow	ov	nv
Direction	dn (decrement)	up (increment)
Interrupt	ei (enabled)	di (disabled)
Sign	ng (negative)	pl (positive)
Zero	zr	nz
Auxiliary Carry	ac	na
Parity	pe (even)	po (odd)
Carry	cy	nc

You can type new flag values in any order. You need not leave spaces between these values. To stop the **r** command, press ENTER. Any flags for which you did not specify new values remain unchanged.

Messages displayed with the r command

If you specify more than one value for a flag, Debug displays the following message:

df error

If you specify a flag code not listed in the preceding table, Debug displays the following message:

bf error

In both cases, Debug ignores all settings specified after the invalid entry.

Default settings for Debug

When you start Debug, the segment registers are set to the bottom of free memory, the instruction pointer is set to 0100h, all flags are cleared, and the remaining registers are set to zero, except for **sp**, which is set to FFEEh.

More Information About Debug: R



[Debug: R--Examples](#)



[Debug: R](#)



Debug: R--Examples

To view the contents of all registers, the status of all flags, and the decoded form of the instruction at the current location, type the following command:

```
r
```

If the current location is CS:11A, the display will look similar to the following:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000 SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A  CD21          INT    21
```

To view only the status of the flags, type the following command:

```
rf
```

Debug displays the information in the following format:

```
NV UP DI NG NZ AC PE NC - _
```

Now you can type one or more valid flag values, in any order, with or without spaces, as in the following command:

```
nv up di ng nz ac pe nc - pleicy
```

Debug stops the **r** command and displays the Debug prompt. To see the changes, type either the **r** or **rf** command. Debug then displays the following:

```
NV UP EI PL NZ AC PE CY - _
```

Press ENTER to return to the Debug prompt.

More Information About Debug: R



[Debug: R--Notes](#)



[Debug: R](#)



Debug: S (Search)

Searches a range of addresses for a pattern of one or more byte values.

s range list

Parameters

range

Specifies the beginning and ending addresses of the range you want to search. For information about valid values for the *range* parameter, see the [debug](#) command.

list

Specifies the pattern of one or more byte values or a string you want to search for. Separate each byte value from the next with a space or a comma. Enclose string values in quotation marks.

More Information About Debug: S



[Debug: S--Note](#)





[Debug: S--Examples](#)

**Debug: S--Note**

If the *list* parameter contains more than one byte value, Debug displays only the first address where the byte value occurs. If *list* contains only one byte value, Debug displays all addresses where the value occurs in the specified range.

More Information About Debug: S

 [Debug: S--Examples](#)

 [Debug: S](#)



Debug: S--Examples

Suppose you want to find all addresses in the range CS:100 through CS:110 that contain the value 41. To do this, type the following command:

```
scs:100 110 41
```

Debug displays the results in the following format:

```
04BA:0104
```

```
04BA:010D
```

```
-
```

The following command searches for the string "Ph" in the range CS:100 through CS:1A0:

```
scs:100 1a0 "Ph"
```

More Information About Debug: S



[Debug: S--Note](#)



[Debug: S](#)



Debug: T (Trace)

Executes one instruction and displays the contents of all registers, the status of all flags, and the decoded form of the instruction executed.

t [=address] [number]

Parameters

=address

Specifies the address at which Debug is to start tracing instructions. If you omit the *address* parameter, tracing begins at the address specified by your program's CS:IP registers. For information about valid values for the *address* parameter, see the [debug](#) command.

number

Specifies the number of instructions to be traced. This value must be a hexadecimal number. The default value is 1.

See Also

For information about executing a loop, a repeated string instruction, a software interrupt, or a subroutine, see the [Debug P \(Proceed\)](#) command.

For information about executing the program currently in memory, see the [Debug G \(Go\)](#) command.

More Information About Debug: T



[Debug: T--Notes](#)



[Debug: T--Example](#)



Debug: T--Notes

Tracing instructions in read-only memory

The **t** command uses the hardware trace mode of the 8086 or 8088 microprocessor. Therefore, you can also trace instructions stored in read-only memory (ROM).

Using the address parameter

You must precede the *address* parameter with an equal sign (=) to distinguish it from the *number* parameter.

More Information About Debug: T



[Debug: T--Example](#)



[Debug: T](#)



Debug: T-Example

To execute one instruction (pointed to by CS:IP), and then display the contents of the registers, the status of the flags, and the decoded form of the instruction, type the following command:

t

If the position of the instruction in the program were 04BA:011A, Debug might display the following information:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000 SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A  CD21          INT    21
```

More Information About Debug: T



[Debug: T--Notes](#)



[Debug: T](#)



Debug: U (Unassemble)

Disassembles bytes and displays their corresponding source statements, including addresses and byte values. The disassembled code looks like a listing for an assembled file.

u [*range*]

Parameter

none

Used without a parameter, the **u** command disassembles 20h bytes (the default number), beginning at the first address after the address displayed by the previous **u** command.

range

Specifies the starting and ending addresses, or the starting address and length, of the code you want to disassemble. For information about valid values for the *range* parameter, see the [debug](#) command.

See Also

For information about assembling mnemonics, see the [Debug A \(Assemble\)](#) command.

For information about displaying the contents of a portion of memory, see the [Debug D \(Dump\)](#) command.

More Information About Debug: U



[Debug: U--Examples](#)



Debug: U--Examples

To disassemble 16 (10h) bytes, beginning at address 04BA:0100, type the following command:

```
u04ba:100|10
```

Debug displays the results in the following format:

```
04BA:0100 206472  AND [SI+72],AH
04BA:0103 69      DB 69
04BA:0104 7665    JBE 016B
04BA:0106 207370  AND [BP+DI+70],DH
04BA:0109 65      DB 65
04BA:010A 63      DB 63
04BA:010B 69      DB 69
04BA:010C 66      DB 66
04BA:010D 69      DB 69
04BA:010E 63      DB 63
04BA:010F 61      DB 61
```

To display only the information for the specific addresses 04BA:0100 through 04BA:0108, type the following command:

```
u04ba:0100 0108
```

Debug displays the following:

```
04BA:0100 206472  AND [SI+72],AH
04BA:0103 69      DB 69
04BA:0104 7665    JBE 016B
04BA:0106 207370  AND [BP+DI+70],DH
```

More Information About Debug: U



[Debug: U](#)



Debug: W (Write)

Writes a file or specific sectors to disk.

To write the contents of the number of bytes specified in the BX:CX registers to a disk file, use the following syntax:

w [*address*]

To bypass the Windows NT file system and directly write to specific sectors, use the following syntax:

w *address drive start number*

Parameters

address

Specifies the beginning memory address of the file, or portion of the file, you want to write to a disk file. If you do not specify *address*, Debug starts from CS:100. For information about valid values for the *address* parameter, see the [debug](#) command.

drive

Specifies the drive that contains the destination disk. This value is numeric: 0 = A, 1 = B, 2 = C, and so on.

start

Specifies the hexadecimal number of the first sector to which you want to write.

number

Specifies the number of sectors to which you want to write.

See Also

For information about specifying a file for the **w** command, see the [Debug N \(Name\)](#) command.

For information about loading the contents of a file or file sectors into memory, see the [Debug L \(Load\)](#) command.

More Information About Debug: W



[Debug: W--Notes](#)



[Debug: W--Example](#)



Debug: W--Notes

Caution: Writing to specific sectors is extremely risky because it bypasses the Windows NT file handler. The disk's file structure can easily be damaged if the wrong values are typed.

You must have specified the name of the disk file when you started Debug or in the most recent Debug **n** (name) command. Both of these methods properly format a filename for a file control block at address CS:5C.

Resetting BX:CX before using the w command without parameters

If you have used a Debug **g** (go), **t** (trace), **p** (proceed), or **r** (register) command, you must reset the BX:CX registers before using the **w** command without parameters.

Writing a modified file to a disk

If you modify the file but do not change the name, length, or starting address, Debug can still correctly write the file to the original disk location.

Limitation on the w command

You cannot write an .EXE or .HEX file with this command.

More Information About Debug: W



[Debug: W--Example](#)



[Debug: W](#)



Debug: W--Example

Suppose you want to write the contents of memory, beginning at the address CS:100, to the disk in drive B. You want the data to begin in the disk's logical sector number 37h and continue for 2Bh sectors. To do this, type the following command:

```
wcs:100 1 37 2b
```

When the write operation is complete, Debug displays the Debug prompt again.

More Information About Debug: W



[Debug: W--Notes](#)



[Debug: W](#)



Debug: XA (Allocate Expanded Memory)

Allocates a specified number of pages of expanded memory.

To use expanded memory, you must have installed an expanded-memory device driver that conforms to version 4.0 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS).

xa [*count*]

Parameter

count

Specifies the number of 16-kilobyte pages of expanded memory to allocate.

See Also

For information about other Debug commands that work with expanded memory, see the Debug commands:

[XD \(deallocate expanded memory\)](#)

[XM \(map expanded-memory pages\)](#)

[XS \(display expanded-memory status\)](#)

More Information About Debug: XA



[Debug: XA--Note](#)



[Debug: XA--Example](#)




Debug: XA--Note

If the specified number of pages is available, Debug displays a message indicating the hexadecimal number of the handle created; otherwise, Debug displays an error message.

More Information About Debug: XA

 [Debug: XA--Example](#)

 [Debug: XA](#)



Debug: XA--Example

To allocate 8 pages of expanded memory, type the following command:

```
xa8
```

If the command is successful, Debug displays a message similar to the following:

```
Handle created=0003
```

More Information About Debug: XA



[Debug: XA--Note](#)



[Debug: XA](#)



Debug: XD (Deallocate Expanded Memory)

Deallocates a handle to expanded memory.

To use expanded memory, you must have installed an expanded-memory device driver that conforms to version 4.0 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS).

xd [*handle*]

Parameter

handle

Specifies the handle you want to deallocate.

See Also

For information about other Debug commands that work with expanded memory, see the Debug commands:

[XA \(allocate expanded memory\)](#)

[XM \(map expanded-memory pages\)](#)

[XS \(display expanded-memory status\)](#)

More Information About Debug: XD



[Debug: XD--Example](#)



Debug: XD--Example

To deallocate handle 0003, type the following command:

```
xd 0003
```

If the command is successful, Debug displays the following message:

```
Handle 0003 deallocated
```

More Information About Debug: XD



[Debug: XD](#)



Debug: XM (Map Expanded Memory Pages)

Maps a logical page of expanded memory, belonging to the specified handle, to a physical page of expanded memory.

To use expanded memory, you must have installed an expanded-memory device driver that conforms to version 4.0 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS).

xm [*lpage*] [*ppage*] [*handle*]

Parameters

lpage

Specifies the number of the logical page of expanded memory that you want to map to physical page *ppage*.

ppage

Specifies the number of the physical page to which *lpage* is to be mapped.

handle

Specifies the handle.

See Also


For information about other Debug commands that work with expanded memory, see the Debug commands:

[XA \(allocate expanded memory\)](#)

[XD \(deallocate expanded memory\)](#)

[XS \(display expanded-memory status\)](#)

More Information About Debug: XM

 [Debug: XM--Example](#)



Debug: XM--Example


To map logical page 5 of handle 0003 to physical page 2, type the following command:

```
xm 5 2 0003
```

If the command is successful, Debug displays the following message:

Logical page 05 mapped to physical page 02

More Information About Debug: XM

 [Debug: XM](#)



Debug: XS (Display Expanded-Memory Status)

Displays information about the status of expanded memory.

To use expanded memory, you must have installed an expanded-memory device driver that conforms to version 4.0 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS).

xs

Parameters

This command takes no parameters.

See Also

For information about other Debug commands that work with expanded memory, see the Debug commands:

[XA \(allocate expanded memory\)](#)

[XD \(deallocate expanded memory\)](#)

[XM \(map expanded-memory pages\)](#)

More Information About Debug: XS



[Debug: XS--Note](#)



[Debug: XS--Example](#)



Debug: XS--Note

The information that Debug displays has the following format:


Handle xx has xx pages allocated


Physical page xx = Frame segment xx

xx of a total xx EMS pages have been allocated

xx of a total xx EMS handles have been allocated

More Information About Debug: XS

 [Debug: XS--Example](#)

 [Debug: XS](#)



Debug: XS--Example

To display expanded-memory information, type the following command:

xs

Debug displays information similar to the following:

Handle 0000 has 0000 pages allocated

Handle 0001 has 0002 pages allocated

Physical page 00 = Frame segment C000

Physical page 01 = Frame segment C400

Physical page 02 = Frame segment C800

Physical page 03 = Frame segment CC00

2 of a total 80 EMS pages have been allocated

2 of a total FF EMS handles have been allocated

More Information About Debug: XS



[Debug: XS--Note](#)



[Debug: XS](#)



Debug Commands

Select a Debug command to get more information.

- | | |
|--|---|
| <input type="checkbox"/> Debug: A (Assemble) | <input type="checkbox"/> Debug: P (Proceed) |
| <input type="checkbox"/> Debug: C (Compare) | <input type="checkbox"/> Debug: Q (Quit) |
| <input type="checkbox"/> Debug: D (Dump) | <input type="checkbox"/> Debug: R (Register) |
| <input type="checkbox"/> Debug: E (Enter) | <input type="checkbox"/> Debug: S (Search) |
| <input type="checkbox"/> Debug: F (Fill) | <input type="checkbox"/> Debug: T (Trace) |
| <input type="checkbox"/> Debug: G (Go) | <input type="checkbox"/> Debug: U (Unassemble) |
| <input type="checkbox"/> Debug: H (Hex) | <input type="checkbox"/> Debug: W (Write) |
| <input type="checkbox"/> Debug: I (Input) | <input type="checkbox"/> Debug: XA (Allocate Expanded Memory) |
| <input type="checkbox"/> Debug: L (Load) | <input type="checkbox"/> Debug: XD (Deallocate Expanded Memory) |
| <input type="checkbox"/> Debug: M (Move) | <input type="checkbox"/> Debug: XM (Map Expanded Memory Pages) |
| <input type="checkbox"/> Debug: N (Name) | <input type="checkbox"/> Debug: XS (Display Expanded Memory Status) |
| <input type="checkbox"/> Debug: O (Output) | |

**A**

- [append](#)
- [arp](#)
- [assoc](#)
- [at](#)
- [attrib](#)

B

- [backup](#)
- [break](#)
- [batch commands](#)
- [buffers](#)

C

- [cacls](#)
- [call](#)
- [chcp](#)
- [chdir \(cd\)](#)
- [chkdsk](#)
- [cls](#)
- [cmd](#)
- [codepage](#)
- [color](#)
- [command symbols](#)
- [comp](#)
- [compact](#)
- [conditional processing symbols](#)
- [convert](#)
- [copy](#)
- [country](#)

D

- [date](#)
- [debug](#)
- [debug commands](#)
- [del](#)
- [device](#)
- [devicehigh](#)
- [devinfo](#)
- [dir](#)
- [diskcomp](#)
- [diskcopy](#)
- [diskperf](#)
- [dos](#)
- [doskey](#)
- [dosonly](#)
- [driveparm](#)

E

- [echo](#)
- [echoconfig](#)
- [edit](#)
- [edlin](#)
- [edlin commands](#)
- [endlocal](#)
- [erase](#)
- [exe2bin](#)
- [exit](#)
- [expand](#)

F

- [fastopen](#)
- [fc](#)
- [fcbs](#)
- [files](#)
- [filter commands](#)
- [find](#)
- [findstr](#)
- [finger](#)
- [for](#)
- [forcedos](#)
- [format](#)
- [ftp](#)
- [ftp commands](#)
- [ftype](#)

G

- [goto](#)
- [grftabl](#)
- [graphics](#)

H

- [help](#)
- [hostname](#)

I

- [if](#)
- [install](#)
- [ipconfig](#)
- [ipxroute](#)

J

No entries

K

- [keyb](#)

L

- [label](#)
- [lastdrive](#)
- [libpath](#)
- [loadfix](#)
- [loadhigh \(lh\)](#)
- [lpq](#)
- [lpr](#)

M

- [mem](#)
- [mkdir \(md\)](#)
- [mode](#)
- [more](#)
- [move](#)
- [MS-DOS configuration commands](#)
- [MS-DOS subsystem commands](#)

N

- [nbtstat](#)
- [netstat](#)
- [net \(command options\)](#)
- [net accounts](#)
- [net computer](#)
- [net config](#)

net config server
net config workstation
net continue
net file
net group
net help
net helpmsg
net localgroup
net name
net pause
net print
net send
net session
net share
net start
net start alerter
net start client service for netware
net start clipboard server
net start computer browser
net start dhcp client
net start directory replicator
net start eventlog
net start file server for macintosh
net start ftp publishing service
net start gateway service for netware
net start lpdsvc
net start messenger
net start microsoft dhcp server
net start net logon
net start network dde
net start network dde dsdm
net start network monitor agent
net start nt lm security support provider
net start print server for macintosh
net start remoteboot
net start remote access connection manager
net start remote access isnsap service
net start remote access server
net start remote procedure call (rpc) locator
net start remote procedure call (rpc) service
net start schedule
net start server
net start simple tcp/ip services
net start snmp
net start spooler
net start tcp/ip netbios helper
net start ups
net start windows internet name service
net start workstation
net statistics
net stop
net time
net use
net user
net view
nlsfunc
nslookup
nslookup commands
ntbooks

[_ntcmdprompt](#)

O

[_OS/2 configuration commands](#)

P

[_path](#)

[_pause](#)

[_pax](#)

[_ping](#)

[_pentnt](#)

[_popd](#)

[_portuas](#)

[_print](#)

[_prompt](#)

[_protshell](#)

[_pushd](#)

Q

[_qbasic](#)

R

[_rcp](#)

[_recover](#)

[_redirection symbols](#)

[_rem](#)

[_rename \(ren\)](#)

[_replace](#)

[_restore](#)

[_rexec](#)

[_rmdir \(rd\)](#)

[_route](#)

[_rsh](#)

S

[_set](#)

[_setlocal](#)

[_setver](#)

[_share](#)

[_shell](#)

[_shift](#)

[_sort](#)

[_stacks](#)

[_start](#)

[_subst](#)

[_switches](#)

[_symbols](#)

T

[_TCP/IP utilities](#)

[_tftp](#)

[_time](#)

[_title](#)

[_tracert](#)

[_tree](#)

[_type](#)

U



No entries

V


[_ver](#)

 verify
 vol

W

 winnt
 winnt32

X

 xcopy

Y

No entries.

Z

No entries.

