Microsoft

Windows XP Professional Product Documentation > Performance and maintenance

Command-line reference A-Z

To find information about a command, on the A-Z button menu at the top of this page, click the letter that the command starts with, and then click the command name.

In addition to the tools installed with Windows XP, there are over 40 support tools included on the Windows XP CD. You can use these tools to diagnose and resolve computer problems. For more information about these support tools, see Windows Support Tools For information about installing support tools, see Install Windows Support Tools

For more information about changes to the functionality of MS-DOS commands, new command-line tools, command shell functionality, configuring the command prompt, and automating commmand-line tasks, see Command-line reference

Some command-line tools require the user to have administrator-level privileges on source and/or target computers.

Command-line tools must be run at the prompt of the Cmd.exe command interpreter. To open Command Prompt, click **Start**, click **Run**, type **cmd**, and then click **OK**. To view help at the command-line, at the command prompt, type the following:

***CommandName* /?**

A

Arp

Assoc

At

Atmadm

Attrib

Top of page

B

Batch files

⇡Top of page

C

Top of page

D

Top of page

E

Top of page

F

Fc

Filter commands

Find

Findstr

Finger

Flattemp

For

Format

Fsutil

Ftp

Ftp subcommands

Ftype

⇡Top of page

G

Getmac

Goto

Gpresult

Gpupdate

Graftabl

⇡Top of page

H

Help

Top of page

Top of page

Top of page

Mountvol

Move

MS-DOS subsystem configuration commands

Msiexec

Msinfo32

⇡Top of page

N

Nbtstat

Net services overview

Net services commands

Netsh command overview

Netsh commands for AAAA

Netsh commands for DHCP

Netsh diagnostic (diag) commands

Netsh commands for Interface IP

Netsh commands for RAS

Netsh commands for Routing

Netsh commands for WINS

Netstat

Nslookup

Nslookup subcommands

Ntbackup

Ntcmdprompt

Ntsd

⇧Top of page

O

Openfiles

⇧Top of page

P

Pagefileconfig

Path

Pathping

Pause

Pbadmin

Pentnt

Perfmon

Ping

Popd

Print

Prncnfg

Prndrvr

Prnjobs

Prnmngr

Prnport

Prnqctl

⇡Top of page

Q

⇡Top of page

R

Runas

⇧Top of page

S

Sc

Schtasks

Secedit

Set

Setlocal

Shift

Shutdown

Sort

Start

Subst

Systeminfo

System File Checker (sfc)

⇧Top of page

T

Taskkill

Tasklist

Tcmsetup

TCP / IP utilities and services

Telnet commands

Terminal Services commands

Tftp

Time

Title

Tracerpt

Tracert

Tree

Type

Typeperf

⇡Top of page

U

Unlodctr

⇡Top of page

V

Ver

Verify

Vol

Vssadmin

⇡Top of page

W

W32tm

Winnt

Winnt32

WMIC overview

Top of page

X

Xcopy

Top of page

Manage Your Profile

Arp

Displays and modifies entries in the Address Resolution Protocol (ARP) cache, which contains one or more tables that are used to store IP addresses and their resolved Ethernet or Token Ring physical addresses. There is a separate table for each Ethernet or Token Ring network adapter installed on your computer. Used without parameters, **arp** displays help.

Syntax

arp [**-a** [*InetAddr*] [**-N** *IfaceAddr*]] [**-g** [*InetAddr*] [**-N** *IfaceAddr*]] [**-d** *InetAddr* [*IfaceAddr*]] [**-s** *InetAddr EtherAddr* [*IfaceAddr*]]

⇡Top of page

Parameters

-a [*InetAddr*] [-N *IfaceAddr*] : Displays current ARP cache tables for all interfaces. To display the ARP cache entry for a specific IP address, use **arp -a** with the *InetAddr* parameter, where *InetAddr* is an IP address. To display the ARP cache table for a specific interface, use the **-N** *IfaceAddr* parameter where *IfaceAddr* is the IP address assigned to the interface. The **-N** parameter is case-sensitive.

-g [*InetAddr*] [-N *IfaceAddr*] : Identical to **-a**.

-d *InetAddr* [*IfaceAddr*] : Deletes an entry with a specific IP address, where *InetAddr* is the IP address. To delete an entry in a table for a specific interface, use the *IfaceAddr* parameter where *IfaceAddr* is the IP address assigned to the interface. To delete all entries, use the asterisk (*) wildcard character in place of *InetAddr*.

-s *InetAddr EtherAddr* [*IfaceAddr*] : Adds a static entry to the ARP cache that resolves the IP address *InetAddr* to the physical address *EtherAddr*. To add a static ARP cache entry to the table for a specific interface, use the *IfaceAddr* parameter where *IfaceAddr* is an IP address assigned to the interface.

/? : Displays help at the command prompt.

Top of page

Remarks

- The IP addresses for *InetAddr* and *IfaceAddr* are expressed in dotted decimal notation.
- The physical address for *EtherAddr* consists of six bytes expressed in hexadecimal notation and separated by hyphens (for example, 00-AA-00-4F-2A-9C).
- Entries added with the **-s** parameter are static and do not time out of the ARP cache. The entries are removed if the TCP/IP protocol is stopped and started. To create permanent static ARP cache entries, place the appropriate **arp** commands in a batch file and use **Scheduled Tasks** to run the batch file at startup.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To display the ARP cache tables for all interfaces, type:

arp -a

To display the ARP cache table for the interface that is assigned the IP address 10.0.0.99, type:

arp -a -N 10.0.0.99

To add a static ARP cache entry that resolves the IP address 10.0.0.80 to the physical address 00-AA-00-4F-2A-9C, type:

arp -s 10.0.0.80 00-AA-00-4F-2A-9C

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Schtasks

Top of page

Manage Your Profile |Legal |Contact Us

Assoc

Displays or modifies file name extension associations. Used without parameters, **assoc** displays a list of all the current file name extension associations.

Syntax

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

⇡Top of page

Parameters

***.ext* :** Specifies the file name extension.

***filetype* :** Specifies the file type with which you want to associate the specified file name extension.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Use a white space after the equal sign to delete the file type association for a file name extension.
- Use the **ftype** command to view current file types that have open command strings defined.
- Use the > redirection operator to redirect the output of **assoc** to a text file.

⇡Top of page

Examples

To view the current file type association for the file name extension .txt, type:

assoc .txt

To delete the file type association for the file name extension .txt, type:

assoc .*txt*=

To view current file type associations one screen at a time, type:

assoc | more

To send the output of **assoc** to the file Assoc.cfg, type:

assoc>assoc.cfg

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftype

More

Start

Command-line reference A-Z

Using command redirection operators

Top of page

Manage Your Profile |Legal |Contact Us

At

Schedules commands and programs to run on a computer at a specified time and date. You can use **at** only when the Schedule service is running. Used without parameters, **at** lists scheduled commands.

Syntax

at [\\ComputerName] [{[*ID*] [**/delete**]|**/delete** [**/yes**]}]

at [[\\ComputerName] *hours***:***minutes* [**/interactive**] [{**/every:***date*[**,**...]|**/next:***date*[**,**...]}] *command*]

⇡Top of page

Parameters

\\***ComputerName* :** Specifies a remote computer. If you omit this parameter, **at** schedules the commands and programs on the local computer.

***ID* :** Specifies the identification number assigned to a scheduled command.

/delete : Cancels a scheduled command. If you omit *ID*, all of the scheduled commands on the computer are canceled.

/yes : Answers yes to all queries from the system when you delete scheduled events.

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

/interactive : Allows *command* to interact with the desktop of the user who is logged on at the time *command* runs.

/every: : Runs *command* on every specified day or days of the week or month (for example, every Thursday, or the third day of every month).

***date* :** Specifies the date when you want to run the command. You can specify one or more days of the week (that is, type **M**,**T**,**W**,**Th**,**F**,**S**,**Su**) or one or more days of the month (that is, type 1 through 31). Separate multiple date entries with commas. If you omit *date*, **at** uses the current day of the month.

/next: : Runs *command* on the next occurrence of the day (for example, next Thursday).

command **:** Specifies the Windows command, program (that is, .exe or .com file), or batch program (that is, .bat or .cmd file) that you want to run. When the command requires a path as an argument, use the absolute path (that is, the entire path beginning with the drive letter). If the command is on a remote computer, specify Universal Naming Convention (UNC) notation for the server and share name, rather than a remote drive letter.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- **Schtasks** is a more powerful superset command-line scheduling tool and includes all the functionality found in the **at** command-line utility. Use **schtasks** instead of **at** for all command-line scheduling tasks. For more information about **schtasks**, see Related Topics.
- Using **at**

  To use **at**, you must be a member of the local Administrators group.
- Loading Cmd.exe

  At does not automatically load Cmd.exe, the command interpreter, before running commands. If you are not running an executable (.exe) file, you must explicitly load Cmd.exe at the beginning of the command as follows:

  cmd /c dir > c:\test.out
- Viewing scheduled commands

  When you use **at** without command-line options, scheduled tasks appear in a table formatted similar to the following:

```
Status  ID   Day       Time        Command Line
OK      1    Each F    4:30 PM     net send group leads status due
OK      2    Each M    12:00 AM    chkstor > check.file
OK      3    Each F    11:59 PM    backup2.bat
```

- Including identification number (*ID*)

  When you include identification number (*ID*) with **at** at a command prompt, information for a single entry appears in a format similar to the following:

```
Task ID:      1

Status:       OK

Schedule:     Each  F

Time of Day:  4:30 PM

Command:      net send group leads status due
```

  After you schedule a command with **at**, especially a command that has command-line options, check that the command syntax is correct by typing **at** without command-line options. If the information in the Command Line column is incorrect, delete the command and retype it. If it is still incorrect, retype the command with fewer command-line options.
- Viewing results

  Commands scheduled with **at** run as background processes. Output is not displayed on the computer screen. To redirect output to a file, use the redirection symbol (>). If you redirect output to a file, you need to use the escape symbol (^) before the redirection symbol, whether you are using **at** at the command line or in a batch file. For example, to redirect output to Output.text, type:

  at 14:45 c:\test.bat ^>c:\output.txt

  The current directory for the executing command is the systemroot folder.
- Changing system time

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

  Scheduled commands are stored in the registry. As a result, you do not lose scheduled tasks if you restart the Schedule service.

•Connecting to network drives

Do not use a redirected drive for scheduled jobs that access the network. The Schedule service might not be able to access the redirected drive, or the redirected drive might not be present if a different user is logged on at the time the scheduled task runs. Instead, use UNC paths for scheduled jobs. For example:

at 1:00pm my_backup \\server\share

Do not use the following syntax, where **x:** is a connection made by the user:

at 1:00pm my_backup x:

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

Top of page

Examples

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

at \\marketing

To learn more about a command with the identification number 3 on the Corp server, 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 Maintenance server, in the Reports shared directory, and the Corp.txt file, type:

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

To back up the hard drive of the Marketing server to a tape drive at midnight every five days, create a batch program called Archive.cmd, which contains the backup commands, and then schedule the batch program to run, type:

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 as follows:

at /delete

To run a command that is not an executable (that is, .exe) file, precede the command with **cmd /c** to load Cmd.exe as follows:

cmd /c dir > c:\test.out

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Using command redirection operators

Schtasks

Top of page

Manage Your Profile |Legal |Contact Us

Atmadm

Monitors connections and addresses that are registered by the ATM Call Manager on an asynchronous transfer mode (ATM) network. You can use **atmadm** to display statistics for incoming and outgoing calls on ATM adapters. Used without parameters, **atmadm** displays statistics for monitoring the status of active ATM connections.

Syntax

atmadm [**/c**][**/a**] [**/s**]

⇡Top of page

Parameters

/c : Displays call information for all current connections to the ATM network adapter installed on this computer.

/a : Displays the registered ATM network service access point (NSAP) address for each adapter installed in this computer.

/s : Displays statistics for monitoring the status of active ATM connections.

/? : Displays help at the command prompt.

⇡Top of page

Examples

To display call information

To display call information for all current connections to the ATM network adapter installed on this computer, type:

atmadm /c

The **atmadm /c** command produces output similar to the following:

```
Windows ATM Call Manager Statistics

ATM Connections on Interface : [009] Olicom ATM PCI 155 Adapter

   Connection   VPI/VCI   Remote Address/
                          Media Parameters (rates in bytes/sec)
   In  PMP SVC    0/193   47000580FFE1000000F21A2E180020481A2E180B
                          Tx:UBR,Peak 0,Avg 0,MaxSdu 1516
                          Rx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 1516
   Out P-P SVC    0/192   47000580FFE1000000F21A2E180020481A2E180B
                          Tx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 1516
                          Rx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 1516
   In  PMP SVC    0/191   47000580FFE1000000F21A2E180020481A2E180B
                          Tx:UBR,Peak 0,Avg 0,MaxSdu 1516
                          Rx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 1516
   Out P-P SVC    0/190   47000580FFE1000000F21A2E180020481A2E180B
                          Tx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 1516
                          Rx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 1516
   In  P-P SVC    0/475   47000580FFE1000000F21A2E180000C110081501
                          Tx:UBR,Peak 16953984,Avg 16953984,
MaxSdu 9188
                          Rx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 9188
   Out PMP SVC    0/194
47000580FFE1000000F21A2E180000C110081501 (0)
                          Tx:UBR,Peak 16953984,Avg 16953984,
MaxSdu 9180
                          Rx:UBR,Peak 0,Avg 0,MaxSdu 0
   Out P-P SVC    0/474   4700918100000000613E5BFE010000C110081500
                          Tx:UBR,Peak 16953984,Avg 16953984,
MaxSdu 9188
                          Rx:UBR,Peak 16953984,Avg 16953984,
MaxSdu 9188
   In  PMP SVC    0/195   47000580FFE1000000F21A2E180000C110081500
                          Tx:UBR,Peak 0,Avg 0,MaxSdu 0
```

```
                                  Rx:UBR,Peak 16953936,Avg 16953936,
MaxSdu 9180
```

The following table contains descriptions of each element in the **atmadm /c** sample output.

Type of Data	Screen Display	Description
Connection Information	In/Out	Direction of the call. **In** is to the ATM network adapter from another device. **Out** is from the ATM network adapter to another device.
	PMP	Point-to-multipoint call.
	P-P	Point-to-point call.
	SVC	Connection is on a switched virtual circuit.
	PVC	Connection is on a permanent virtual circuit.
VPI/VCI Information	VPI/VCI	Virtual path and virtual channel of the incoming or outgoing call.
Remote Address/ Media Parameters	47000580FFE1000000F21A2E180000C110081500	NSAP address of the calling **(In)** or called **(Out)** ATM device.

	Tx	The *Tx* parameter includes the following three elements: •Default or specified bit-rate type (UBR, CBR, VBR, or ABR) •Default or specified line speed •Specified service data unit (SDU) size
	Rx	The *Rx* parameter includes the following three elements: •Default or specified bit-rate type (UBR, CBR, VBR, or ABR) •Default or specified line speed •Specified SDU size

To list all registered addresses

To display the registered ATM network service access point (NSAP) address for each adapter installed in this computer, type:

atmadm /a

The **atmadm /a** command produces output similar to the following:

```
Windows ATM Call Manager Statistics
```

```
ATM Addresses for Interface : [009] Olicom ATM PCI 155 Adapter

47000580FFE1000000F21A2E180000C110081500
```

To display statistics

To display statistics for monitoring the status of active ATM connections, type:

atmadm /s

The **atmadm /s** command produces output similar to the following:

```
Windows ATM Call Manager Statistics

ATM Call Manager statistics for Interface : [009] Olicom ATM PCI
155 Adapter

        Current Active calls                        = 4
        Total successful Incoming calls             = 1332
        Total successful Outgoing calls             = 1297
        Unsuccessful Incoming calls                 = 1
        Unsuccessful Outgoing calls                 = 1
        Calls Closed by Remote                      = 1302
        Calls Closed Locally                        = 1323
        Signalling and ILMI Packets Sent            = 33655
        Signalling and ILMI Packets Received        = 34989
```

The following table contains descriptions of each element in the **atmadm /s** sample output.

Call Manager Statistic	Description
Current Active calls	Calls currently active on the ATM adapter installed on this computer.
Total successful Incoming calls	Calls successfully received from other devices on this ATM network.
Total successful Outgoing calls	Calls successfully completed to other ATM devices on this network from this computer.
Unsuccessful Incoming calls	Incoming calls that failed to connect to this computer.

Unsuccessful Outgoing calls	Outgoing calls that failed to connect to another device on the network.
Calls Closed by Remote	Calls closed by a remote device on the network.
Calls Closed Locally	Calls closed by this computer.
Signaling and ILMI Packets Sent	Number of integrated local management interface (ILMI) packets sent to the switch to which this computer is attempting to connect.
Signaling and ILMI Packets Received	Number of ILMI packets received from the ATM switch.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command shell overview

Top of page

Manage Your Profile |Legal |Contact Us

Attrib

Displays, sets, or removes the read-only, archive, system, and hidden attributes assigned to files or directories. Used without parameters, **attrib** displays attributes of all files in the current directory.

Syntax

attrib [{**+r**|**-r**}] [{**+a**|**-a**}] [{**+s**|**-s**}] [{**+h**|**-h**}] [[*Drive***:**][*Path*] *FileName*] [**/s**[**/d**]]

Top of page

Parameters

+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 system file attribute.

-s : Clears the system file attribute.

+h : Sets the hidden file attribute.

-h : Clears the hidden file attribute.

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the directory, file, or set of files for which you want to display or change attributes. You can use wildcard characters (that is, ? and *) in the *FileName* parameter to display or change the attributes for a group of files.

/s : Applies **attrib** and any command-line options to matching files in the current directory and all of its subdirectories.

/d : Applies **attrib** and any command-line options to directories.

/? : Displays help at the command prompt.

Top of page

Remarks

- Working with groups of files

  You can use wildcard characters (that is, ? 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 these attributes before you can change any other attributes for that file.
- Using the archive attribute

  The archive attribute (that is, **+a**) marks files that have changed since the last time they were backed up. The **xcopy** command uses archive attributes. For more information about archive attributes and **xcopy**, see Related Topics.
- The **attrib** command, with different parameters, is available from the Recovery Console.

Top of page

Examples

To display the attributes of a file named News86 located on the current drive, type:

```
attrib news86
```

To assign the read-only attribute to the file named Report.txt, type:

```
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:

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

Consider a scenario where 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. First, you need to set the archive attribute for all files on drive A.

Second, you need to clear the archive attribute for those files with the .bak extension. For example, type:

attrib +a a:*.* attrib -a a:*.bak

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

xcopy a: b: /a

If you want **xcopy** to clear each file's **archive** attribute after it copies the files, use the **/m** command-line option instead of **/a**. For example, type:

xcopy a: b: /m

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Dir

Copy

Using batch files

With batch files, which are also called batch programs or scripts, you can simplify routine or repetitive tasks. A batch file is an unformatted text file that contains one or more commands and has a .bat or .cmd file name extension. When you type the file name at the command prompt, Cmd.exe runs the commands sequentially as they appear in the file.

You can include any command in a batch file. Certain commands, such as **for**, **goto**, and **if**, enable you to do 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 files.

The standard error codes that most applications return are 0 if no error occurred and 1 (or higher value) if an error occurred. Please refer to your application help documentation to determine the meaning of specific error codes.

For more information about batch file operations, see the following topics:

•Using batch parameters
•Using filters
•Using command redirection operators

For more information about commands that you can use in batch files, click a command:

•Call
•Echo
•Endlocal
•For
•Goto
•If
•Pause
•Rem
•Setlocal
•Shift

Top of page

Manage Your Profile |Legal |Contact Us

Bootcfg

Configures, queries, or changes Boot.ini file settings.

To view the command syntax, click a command:

bootcfg addsw

Adds operating system load options for a specified operating system entry.

Syntax

bootcfg /addsw [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/mm** *MaximumRAM*] [**/bv**] [**/so**] [**/ng**] **/id** *OSEntryLineNum*

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/mm *MaximumRAM* : Adds the /maxmem switch to the specified *OSEntryLineNum* and sets the maximum amount of RAM that the operating system can use.

/bv : Adds the /basevideo switch to the specified *OSEntryLineNum*, directing the operating system to use standard VGA mode for the installed video driver.

/so : Adds the /sos switch to the specified *OSEntryLineNum*, directing the operating system to display device driver names while they are being loaded.

/ng : Adds the /noguiboot switch to the specified *OSEntryLineNum*, disabling the Windows XP Professional progress bar that appears before the CTRL+ALT+DEL logon prompt.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to which the operating system load options are added. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **bootcfg /addsw** command:

bootcfg /addsw /mm 64 /id 2
bootcfg /addsw /so /id 3
bootcfg /addsw /so /ng /s srvmain /u hiropln /id 2
bootcfg /addsw /ng /id 2
bootcfg /addsw /mm 96 /ng /s srvmain /u maindom\hiropln /p p@ssW23 /id 2

bootcfg copy

Makes another operating system instance copy, for which switches can be added.

Top of page

Syntax

bootcfg /copy [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/d** *Description*] [**/ id** *OSEntryLineNum*]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/d *Description* : Specifies the description for the new operating system entry.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to copy. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

⇡Top of page

Examples

The following examples show how you can use the **bootcfg /copy** command:

bootcfg /copy /d "\ABC Server\" /id 1
bootcfg /copy /s srvmain /u maindom\hiropln /p p@ssW23 /d "Windows XP" /id 2
bootcfg /copy /u hiropln /p p@ssW23 /d "AB Ver 1.001" /id 2

bootcfg dbg1394

Configures 1394 port debugging for a specified operating system entry.

⇡Top of page

Syntax

bootcfg /dbg1394 {**ON**|**OFF**|**EDIT**} [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/ch** *channel*] **/id** *OSEntryLineNum*

⇡Top of page

Parameters

ON|OFF|EDIT : Specifies the value for 1394 port debugging.

Value	Description
ON	Enables remote debugging support by adding the /dbg1394 switch to the specified *OSEntryLineNum*.
OFF	Disables remote debugging support by removing the /dbg1394 switch from the specified *OSEntryLineNum*.
EDIT	Allows changes to port and baud rate settings by changing the channel values associated with the /dbg1394 switch for the specified *OSEntryLineNum*.

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/ch *channel* : Specifies the channel to use for debugging. Valid values are integers between 1 and 64. Do not use the **/ch** *channel* parameter if 1394 port debugging is being disabled.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to which the 1394 port debugging options are added. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **bootcfg /dbg1394** command:

bootcfg /dbg1394 /id 2
bootcfg /dbg1394 on /ch 1 /id 3
bootcfg /dbg1394 edit /ch 8 /id 2
bootcfg /s srvmain /u maindom\hiropln /p p@ssW23 /dbg1394 off /id 2

bootcfg debug

Adds or changes the debug settings for a specified operating system entry.

Top of page

Syntax

bootcfg /debug {**ON**|**OFF**|**EDIT**} [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/port** {**COM1**|**COM2**|**COM3**|**COM4**}] [**/baud** {**9600**|**19200**|**38400**|**57600**|**115200**}] [**/id** *OSEntryLineNum*]

Top of page

Parameters

ON|OFF|EDIT : Specifies the value for debugging.

Value	Description
ON	Enables remote debugging support by adding the /debug switch to the specified *OSEntryLineNum*.
OFF	Disables remote debugging support by removing the /debug switch from the specified *OSEntryLineNum*.
EDIT	Allows changes to port and baud rate settings by changing the values associated with the /debug switch for the specified *OSEntryLineNum*.

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/port {COM1|COM2|COM3|COM4} : Specifies the COM port to be used for debugging. Do not use the **/port** parameter if debugging is being disabled.

/baud {9600|19200|38400|57600|115200} : Specifies the baud rate to be used for debugging. Do not use the **/baud** parameter if debugging is being disabled.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to which the debugging options are added. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

Top of page

Remarks

•If 1394 port debugging is required, use **bootcfg / dbg1394**.

Top of page

Examples

The following examples show how you can use the **bootcfg /debug** command:

```
bootcfg /debug on /port com1 /id 2
bootcfg /debug edit /port com2 /baud 19200 /id 2
bootcfg /s srvmain /u maindom\hiropln /p p@ssW23 /debug off /id 2
```

bootcfg default

Specifies the operating system entry to designate as the default.

Top of page

Syntax

bootcfg /default [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/id** *OSEntryLineNum*]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain*\*User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to designate as default. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **bootcfg /default** command:

bootcfg /default /id 2
bootcfg /default /s srvmain /u maindom\hiropln /p p@ssW23 /id 2

bootcfg delete

Deletes an operating system entry in the [operating systems] section of the Boot.ini file.

Top of page

Syntax

bootcfg /delete [**/s** *Computer* [**/u** *Domain*\User **/p** *Password*]] [**/id** *OSEntryLineNum*]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to delete. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

⇡Top of page

Examples

The following examples show how you can use the **bootcfg /delete** command:

```
bootcfg /delete /id 1
bootcfg /delete /s srvmain /u maindom\hiropln /p p@ssW23 /id 3
```

bootcfg ems

Enables the user to add or change the settings for redirection of the EMS console to a remote computer. By enabling EMS, you add a "redirect=Port#" line to the [boot loader] section of the Boot.ini file and a /redirect switch to the specified operating system entry line. The EMS feature is enabled only on servers.

⇡Top of page

Syntax

bootcfg /ems {**ON**|**OFF**|**EDIT**} [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/port** {**COM1**|**COM2**|**COM3**|**COM4**|**BIOSSET**}] [**/baud** {**9600**|**19200**|**38400**|**57600**|**115200**}] [/**id** *OSEntryLineNum*]

⇡Top of page

Parameters

ON|OFF|EDIT : Specifies the value for EMS redirection.

Value	Description
ON	Enables remote output for the specified *OSEntryLineNum*. Adds a /redirect switch to the specified *OSEntryLineNum* and a redirect=com*X* setting to the [boot loader] section. The value of com*X* is set by the **/port** parameter.
OFF	Disables output to a remote computer. Removes the /redirect switch from the specified *OSEntryLineNum* and the redirect=com*X* setting from the [boot loader] section.
EDIT	Allows changes to port settings by changing the redirect=com*X* setting in the [boot loader] section. The value of com*X* is reset to the value specified by the **/port** parameter.

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/port {COM1|COM2|COM3|COM4|BIOSSET} : Specifies the COM port to be used for redirection. **BIOSSET** directs EMS to get the BIOS settings to determine which port should be used for redirection. Do not use the **/port** parameter if remotely administered output is being disabled.

/baud {9600|19200|38400|57600|115200} : Specifies the baud rate to be used for redirection. Do not use the **/baud** parameter if remotely administered output is being disabled.

/id *OSEntryLineNum* : Specifies the operating system entry line number to which the EMS option is added in the [operating systems] section of the Boot.ini file. The first line after the [operating systems] section header is 1. This parameter is required when the EMS value is set to **ON** or **OFF**.

/?: Displays help at the command prompt.

Top of page

Remarks

- For more information on the Emergency Management Services (EMS), see Help and Support Center.

⇡Top of page

Examples

The following examples show how you can use the **bootcfg /ems** command:

bootcfg /ems on /port com1 /baud 19200 /id 2
bootcfg /ems on /port biosset /id 3
bootcfg /s srvmain /ems off /id 2
bootcfg /ems edit /port com2 /baud 115200
bootcfg /s srvmain /u maindom\hiropln /p p@ssW23 /ems off /id 2

bootcfg query

Queries and displays the [boot loader] and [operating systems] section entries from Boot.ini.

⇡Top of page

Syntax

bootcfg /query [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/?: Displays help at the command prompt.

Top of page

Remarks

- The following is a sample of **bootcfg /query** output:

```
Boot Loader Settings
--------------------
timeout: 30
default: multi(0)disk(0)rdisk(0)partition(1)\WINDOWS

Boot Entries
------------
Boot entry ID:   1
Friendly Name:   "Microsoft Windows XP Professional"
Path:            multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
OS Load Options: /fastdetect /debug /debugport=com1:
```

- The Boot Loader Settings portion of the **bootcfg query** output displays each entry in the [boot loader] section of Boot.ini.
- The Boot Entries portion of the **bootcfg query** output displays the following detail for each operating system entry in the [operating systems] section of Boot.ini: Boot entry ID, Friendly Name, Path, and OS Load Options.

Top of page

Examples

The following examples show how you can use the **bootcfg /query** command:

bootcfg /query
bootcfg /query /s srvmain /u maindom\hiropln /p p@ssW23
bootcfg /query /u hiropln /p p@ssW23

bootcfg raw

Adds operating system load options specified as a string to an operating system entry in the [operating systems] section of the Boot.ini file.

Top of page

Syntax

bootcfg [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] **/raw** *OSLoadOptionsString* [**/id** *OSEntryLineNum*]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

***OSLoadOptionsString* :** Specifies the operating system load options to add to the operating system entry. These load options will replace any existing load options associated with the operating system entry. No validation of *OSLoadOptions* is done.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to update. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

Top of page

Remarks

- **Bootcfg raw** is used to add text to the end of an operating system entry, overwriting any existing operating system entry options. This text should contain valid OS Load Options such as /debug, /fastdetect, /nodebug, /baudrate, /crashdebug, and /sos. For example, the following command adds "/debug /fastdetect" to the end of the first operating system entry, replacing any previous operating system entry options:

  bootcfg /raw "/debug /fastdetect" /id 1
- For more information about optional OS Load Options and how Windows uses the Boot.ini file during startup, see article 170756, "Available Switch Options for Windows NT Boot.ini File," in the Microsoft Knowledge Base.

Top of page

Examples

The following examples show how you can use the **bootcfg /raw** command:

bootcfg /raw "/debug /sos" /id 2
bootcfg /raw /s srvmain /u maindom\hiropln /p p@ssW23 "/crashdebug " /id 2

bootcfg rmsw

Removes operating system load options for a specified operating system entry.

Top of page

Syntax

bootcfg /rmsw [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/mm**] [**/bv**] [**/so**] [**/ng**] **/id** *OSEntryLineNum*

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/mm : Removes the /maxmem switch and its associated maximum memory value from the specified *OSEntryLineNum*. The /maxmem switch specifies the maximum amount of RAM that the operating system can use.

/bv : Removes the /basevideo switch from the specified *OSEntryLineNum*. The /basevideo switch directs the operating system to use standard VGA mode for the installed video driver.

/so : Removes the /sos switch from the specified *OSEntryLineNum*. The /sos switch directs the operating system to display device driver names while they are being loaded.

/ng : Removes the /noguiboot switch from the specified *OSEntryLineNum*. The /noguiboot switch disables the Windows XP Professional progress bar that appears before the CTRL+ALT+DEL logon prompt.

/id *OSEntryLineNum* : Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file from which the OS Load Options are removed. The first line after the [operating systems] section header is 1.

/?: Displays help at the command prompt.

⇡Top of page

Examples

The following examples show how you can use the **bootcfg /rmsw** command:

```
bootcfg /rmsw /mm 64 /id 2
bootcfg /rmsw /so /id 3
bootcfg /rmsw /so /ng /s srvmain /u hiropln /id 2
bootcfg /rmsw /ng /id 2
bootcfg /rmsw /mm 96 /ng /s srvmain /u maindom\hiropln /p p@ssW23 /id 2
```

bootcfg timeout

Changes the operating system time-out value.

⇡Top of page

Syntax

bootcfg /timeout *TimeOutValue* [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]]

Top of page

Parameters

***TimeOutValue* :** Specifies the timeout value in the [boot loader] section. The value is the number of seconds the user has to select an operating system from the boot loader screen before NTLDR loads the default. If the value is 0, then NTLDR immediately starts the default operating system without displaying the boot loader screen.

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **bootcfg /timeout** command:

bootcfg /timeout 30
bootcfg /s srvmain /u maindom\hiropln /p p@ssW23 /timeout 50

Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Break

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

Top of page

Manage Your Profile |Legal |Contact Us

Cacls

Displays or modifies discretionary access control list (DACL) files.

Syntax

cacls *FileName* [**/t**] [**/e**] [**/c**] [**/g** *User***:***permission*] [**/r** *User* [...]] [**/p** *User***:***permission* [...]] [**/d** *User* [...]]

⇡Top of page

Parameters

***FileName* :** Required. Displays DACLs of specified files.

/t : Changes DACLs of specified files in the current directory and all subdirectories.

/e : Edits a DACL instead of replacing it.

/c : Continues to change DACLs, ignoring errors.

/g *User*:*permission* : Grants access rights to the specified user. The following table lists valid values for *permission*.

Value	Description
n	None
r	Read
w	Write
c	Change (Write)
f	Full Control

/r *User* : Revokes access rights for the specified user.

/p *User*:*permission* : Replaces access rights for the specified user. The following table lists valid values for *permission*.

Value	Description
n	None
r	Read
w	Write
c	Change (Write)
f	Full Control

/d *User* : Denies access for the specified user.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•Use the following table to interpret the results.

Output	ACE applies to
OI	This folder and files
CI	This folder and subfolders
IO	The ACE does not apply to the current file/directory.
No output message	This folder only
(IO)(CI)	This folder, subfolders and files
(OI)(CI)(IO)	Subfolders and files only
(CI)(IO)	Subfolders only
(OI)(IO)	Files only

•You can use wildcards (that is, ? and *) to specify multiple files.

- You can specify more than one user.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Call

Calls one batch program from another without stopping the parent batch program. The **call** command accepts labels as the target of the call. **Call** has no effect at the command-line when used outside of a script or batch file.

Syntax

call [[*Drive*:][*Path*] *FileName* [*BatchParameters*]] [**:***label* [*arguments*]]

⇡Top of page

Parameters

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the batch program you want to call. The *FileName* parameter must have a .bat or .cmd extension.

***BatchParameters* :** Specifies any command-line information required by the batch program, including command-line options, file names, batch parameters (that is, %0 through %9), or variables (for example, *%baud%*).

:*label* : Specifies the label to which you want a batch program control to jump. By using **call** with this parameter, you create a new batch file context and pass control to the statement after the specified label. The first time the end of the batch file is encountered (that is, after jumping to the label), control returns to the statement after the **call** statement. The second time the end of the batch file is encountered, the batch script is exited. For a description of the **goto :eof** extension that allows you to return from a batch script, see Related Topics.

arguments* :** Specifies any command-line information that you pass to the new instance of the batch program that begins at **:label*, including command-line options, file names, batch parameters (that is, *%1* through *%9*), or variables (for example, *%baud%*).

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Using batch parameters

  Batch parameters can contain any information that you can pass to a batch program, including command-line options, file names, batch parameters (that is, %0 through %9), and variables (for example, *%baud%*). For more information about batch parameters, see Related Topics.
- Using pipes and redirection symbols

  Do not use pipes and redirection symbols with **call**.
- 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.
- Working with command extensions

  With command extensions enabled (that is, the default), **call** accepts a *label* as the target of the call. The correct syntax is as follows:

  call :*label arguments*

  For more information about enabling and disabling command extensions, see **cmd** in Related Topics.

⇡Top of page

Examples

To run the Checknew.bat program from another batch program, type the following command in the parent batch program:

call checknew

If the parent batch program accepts two batch parameters and you want it to pass those parameters to Checknew.bat, use the following command in the parent batch program:

call checknew %1 %2

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Using batch parameters

Cmd

Goto

Using batch parameters

Command-line reference A-Z

Top of page

Terminal Services change commands

For more information about the Terminal Services change commands, see the following topics:

- Change logon
- Change port
- Change user

Top of page

Manage Your Profile |Legal |Contact Us

Change logon

Enables or disables logons from client sessions, or displays current logon status. This utility is useful for system maintenance.

Syntax

change logon {**/enable**|**/disable**|**/query**}

⇡Top of page

Parameters

/enable : Enables logons from client sessions, but not from the console.

/disable : Disables subsequent logons from client sessions, but not from the console. Currently logged on users are not affected.

/query : Displays the current logon status, whether enabled or disabled.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Only administrators can use the **change logon** command.
- Logons are re-enabled when you restart the system. If you are connected to the terminal server from a client session and disable logons, and then log off before re-enabling logons, you will not be able to reconnect from your session. Log on at the console in order to re-enable logons from client sessions.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Terminal Services commands

Top of page

Manage Your Profile |Legal |Contact Us

Change port

Lists or changes the COM port mappings to be compatible with MS-DOS applications.

Syntax

change port [{*portx*=*porty*|**/d** *portx*|**/query**}]

⇡Top of page

Parameters

***portx*=*porty* :** Maps COM port *x* to *porty*.

/d *portx* : Deletes the mapping for COM port *x*.

/query : Displays the current port mappings.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Most MS-DOS applications support only COM1 through COM4 serial ports. The **change port** command maps a serial port to a different port number, allowing applications that do not support high-numbered COM ports to access the serial port. For example, to map COM12 to COM1 for use by an MS-DOS-based application, type **change port com12=com1**. Remapping works only for the current session and is not retained if you log off from a session and then log on again.
- Use **change port** without any parameters to display the available COM ports and their current mappings.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Terminal Services commands

Top of page

Manage Your Profile |Legal |Contact Us

Change user

Changes the setting for .ini file mapping.

Syntax

change user {**/execute** | **/install** | **/query**}

⇡Top of page

Parameters

/execute : Enables .ini file mapping to the home directory. This is the default setting.

/install : Disables .ini file mapping to the home directory. All .ini files are read and written to the system directory. You must disable .ini file mapping when installing applications on a terminal server.

/query : Displays the current setting for .ini file mapping.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Use **change user /install** before installing an application to create .ini files for the application in the system directory. These files are used as master copies for user-specific .ini files. After installing the application, use **change user /execute** to revert to standard .ini file mapping.
- The first time you run the application, it searches the home directory for its .ini files. If the .ini files are not found in the home directory, but are found in the system directory, Terminal Services copies the .ini files to the home directory, ensuring that each user has a unique copy of the application .ini files. Any new .ini files are created in the home directory.
- Each user should have a unique copy of the .ini files for an application. This prevents instances where different users might have incompatible application configurations (for example, different default directories or screen resolutions).

- When the system is in install mode (**change user /install**), several things occur. All registry entries that are created are shadowed under **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\TerminalServer\Install**. Keys added to **HKEY_CURRENT_USER** are copied under the **SOFTWARE** key, and keys added to **HKEY_LOCAL_MACHINE** are copied under **MACHINE**. If the application queries the Windows directory using system calls, such as **GetWindowsDirectory**, the Terminal server returns the *systemroot* directory. If any .ini file entries are added using system calls, such as **WritePrivateProfileString**, they are added to the .ini files under the *systemroot* directory.
- When the system returns to execution mode (**change user /execute**), and the application tries to read a registry entry under **HKEY_CURRENT_USER** that does not exist, Terminal Services checks to see whether a copy of the key exists under the **\TerminalServer\Install** key. If it does, the keys are copied to the appropriate location under **HKEY_CURRENT_USER**. If the application tries to read from an .ini file that does not exist, Terminal Services searches for that .ini file under the system root. If the .ini file is in the system root, it is copied to the \Windows subdirectory of the user's home directory. If the application queries the Windows directory, the Terminal server returns the \Windows subdirectory of the user's home directory.
- When you log on, Terminal Services checks whether its system .ini files are newer than the .ini files on your computer. If the system version is newer, your .ini file is either replaced or merged with the newer version. This depends on whether or not the INISYNC bit, 0x40, is set for this .ini file. Your previous version of the .ini file is renamed as Inifile.ctx. If the system registry values under the **\TerminalServer\Install** key are newer than your version under **HKEY_CURRENT_USER**, then your version of the keys is deleted and replaced with the new keys from under **\TerminalServer\Install**.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Chcp

Displays the number of the active console code page, or changes the console's active console code page. Used without parameters, **chcp** displays the number of the active console code page.

Syntax

chcp [*nnn*]

Top of page

Parameters

nnn **:** Specifies the code page. The following table lists each code page supported and its country/region or language:

Code page	Country/region 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

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Only the original equipment manufacturer (OEM) code page installed with Windows XP appears correctly in a command prompt window that uses Raster fonts. Other code pages appear correctly in full-screen mode or command prompt windows that use TrueType fonts.
- You do not need to prepare code pages, as in MS-DOS.
- Programs that you start after you assign a new code page use the new code page, however, programs (except Cmd.exe) that you started before assigning the new code page use the original code page.

⇡Top of page

Examples

To view the active code page setting, type:

chcp

A message similar to the following appears:

Active code page: 437

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

chcp 850

If the specified code page is invalid, the following error message appears:

Invalid code page

⇡Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Country

Device

Devicehigh

Mode

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Chdir (Cd)

Displays the name of the current directory or changes the current folder. Used with only a drive letter (for example, **chdir** C:), **chdir** displays the names of the current drive and folder. Used without parameters, **chdir** displays the current drive and directory.

Syntax

chdir [[/**d**] [*Drive***:**][*Path*] [**..**]] [[/**d**] [*Drive***:**][*Path*] [**..**]]

cd [[/**d**] [*Drive***:**][*Path*] [**..**]] [[/**d**] [*Drive***:**][*Path*] [**..**]]

↑Top of page

Parameters

/d : Changes the current drive or the current directory for a drive.

[*drive:*][*Path*] : Specifies the drive (that is, if it is different from the current drive) and directory to which you want to change.

[..] : Specifies that you want to change to the parent folder.

/? : Displays help at the command prompt.

↑Top of page

Remarks

- Working with command extensions

  With command extensions enabled (that is, the default), the current directory path matches the folder names exactly as they appear on your hard drive, using the same uppercase or lowercase folder-name format. For example, if the folder on your hard drive is called C:\Temp, **CD C:\TEMP** sets the current directory to C:\Temp to match the folder-name format of the folder on your hard drive.

  To disable command extensions for a particular process, type:

  cmd e:off

  When you disable command extensions, **chdir** does not treat white spaces as delimiters. As a result, you can change to a subdirectory name that contains a white space without having to surround [*Path*] in quotation marks. For example, the following path changes to the \Start menu subdirectory:

  cd \winnt\profiles\username\programs\start menu

  For more information about enabling and disabling command extensions, see **cmd** in Related Topics.
- 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:

  cd
- Changing the default directory on one drive from another drive

  To change the default directory on a drive different from the one you are on, type one of the following:

  chdir [*Drive***:\**[*directory*]]

  cd [*Drive***:\**[*directory*]]

  To verify the change to the directory, type one of the following:

  chdir [*Drive***:**]

  cd [*Drive***:**]

•The **chdir** command, with different parameters, is available from the Recovery Console.

Top of page

Examples

When you use it with a drive name, **chdir** displays the current directory for that drive. For example, if you type **cd c:** at the C:\Temp directory prompt, the following appears:

```
C:\Temp
```

To change your current directory to a directory named Reports, type one of the following commands:

chdir \reports

cd \reports

To change your current directory to a subdirectory \Specials\Sponsors, type:

cd \specials\sponsors

Or, if your current directory is \Specials, type the following command to change to the \Specials \Sponsors subdirectory:

cd sponsors

To change from a subdirectory to its parent directory, type:

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 drive B, typing **chdir** the following appears:

```
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:

chdir c:\public\jones

```
copy c:*.* d:\

chdir c:\public\lewis

copy c:*.* d:\
```

If you want to copy all files in the \Public\Jones and \Public\Lewis directories to your current location on drive D, type:

```
chdir c:\public\jones

copy c:*.* d:

chdir c:\public\lewis

copy c:*.* d:
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Mkdir

Cmd

Chkdsk

Creates and displays a status report for a disk based on the file system. **Chkdsk** also lists and corrects errors on the disk. Used without parameters, **chkdsk** displays the status of the disk in the current drive.

Syntax

chkdsk [*volume***:**][[*Path*] *FileName*] [**/f**] [**/v**] [**/r**] [**/x**] [**/i**] [**/c**] [**/l**[**:***size*]]

⇡Top of page

Parameters

volume: **:** Specifies the drive letter (followed by a colon), mount point, or volume name.

[*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 wildcard characters (that is, * and ?) to specify multiple files.

/f : Fixes errors on the disk. The disk must be locked. If **chkdsk** cannot lock the drive, a message appears that asks you if you want to check the drive the next time you restart the computer.

/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.

/x : Use with NTFS only. Forces the volume to dismount first, if necessary. All open handles to the drive are invalidated. **/x** also includes the functionality of **/f**.

/i : Use with NTFS only. Performs a less vigorous check of index entries, reducing the amount of time needed to run **chkdsk**.

/c : Use with NTFS only. Skips the checking of cycles within the folder structure, reducing the amount of time needed to run **chkdsk**.

/l[:*size*] : Use with NTFS only. Changes the log file size to the size you type. If you omit the size parameter, **/l** displays the current size.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Running **chkdsk**

  To run **chkdsk** on a fixed disk, you must be a member of the Administrators group.
- Checking a locked drive at restart

  If you want **chkdsk** to correct disk errors, you cannot have open files on the drive. If files are open, the following error message appears:

  Chkdsk cannot run because the volume is in use by another process. Would you like to schedule this volume to be checked the next time the system restarts? (Y/N)

  If you choose to check the drive the next time you restart the computer, **chkdsk** checks the drive and corrects errors automatically when you restart the computer. If the drive partition is a boot partition, **chkdsk** automatically restarts the computer after it checks the drive.
- Reporting disk errors

  Chkdsk examines disk space and disk use for the file allocation table (FAT) and NTFS file systems. **Chkdsk** provides information specific to each file system in a status report. The status report shows errors found in the file system. If you run **chkdsk** without the **/f** command-line option on an active partition, it might report spurious errors because it cannot lock the drive. You should use **chkdsk** occasionally on each disk to check for errors.

- Fixing disk errors

  Chkdsk corrects disk errors only if you specify the **/f** command-line option. **Chkdsk** must be able to lock the drive to correct errors. Because repairs usually change a disk's file allocation table and sometimes cause a loss of data, **chkdsk** sends 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 saves each lost chain in the root directory as a file with a name in the format File*nnnn*.chk. When **chkdsk** finishes, you can check these files to see if they contain any data you need. If you press **N**, Windows fixes the disk, but it does not save the contents of the lost allocation units.

  If you do not use the **/f** command-line option, **chkdsk** sends a message if a file needs to be fixed, but it does not fix any errors.

  If you use **chkdsk /f** on a very large disk (for example, 70 gigabytes) or a disk with a very large number of files (for example, millions of files), **chkdsk** might take a long time (for example, over several days) to complete. The computer is not available during this time because **chkdsk** does not relinquish control until it is finished.
- Checking a FAT disk

  Windows 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 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.
```

- Using **chkdsk** with open files

  If you specify the **/f** command-line option, **chkdsk** sends an error message if there are open files on the disk. If you do not specify the **/f** command-line option 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.
- Finding physical disk errors

  Use the **/r** command-line option to find physical disk errors in the file system. For information about recovering physically damaged files with **recover**, see Related Topics.
- Reporting bad disk sectors

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

  The following table lists the exit codes that **chkdsk** reports after it has finished.

Exit code	Description
0	No errors were found.
1	Errors were found and fixed.
2	Disk cleanup, such as garbage collection, was performed, or cleanup was not performed because **/f** was not specified.
3	Could not check the disk, errors could not be fixed, or errors were not fixed because **/f** was not specified.

- The **chkdsk** command, with different parameters, is available from the Recovery Console.

⇡Top of page

Examples

If you want to check the disk in drive D and have Windows fix errors, type:

```
chkdsk d: /f
```

If it encounters errors, **chkdsk** pauses and displays messages. **Chkdsk** finishes by displaying a report that lists the status of the disk. You cannot open any files on the specified drive until **chkdsk** finishes.

To check all of files on a FAT disk in the current directory for noncontiguous blocks, type:

chkdsk *.*

Chkdsk displays a status report, and then lists the files matching the file specification that have noncontiguous blocks.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Chkntfs

Displays or specifies whether automatic system checking is scheduled to be run on a FAT, FAT32, or NTFS volume when the computer is started.

Syntax

chkntfs *volume***:** [...]

chkntfs [**/d**]

chkntfs [**/t**[**:***time*]]

chkntfs [**/x** *volume***:** [...]]

chkntfs [**/c** *volume***:** [...]]

⇡Top of page

Parameters

volume: **[...] :** Required. Specifies the drive letter (followed by a colon), mount point, or volume name. Displays a message that identifies the file system of the specified volume. If automatic file checking is scheduled to be run, this parameter displays a message indicating whether the volume has been corrupted, which requires you to run **chkdsk**.

/d : Restores all **chkntfs** default settings, except the countdown time for automatic file checking. Default behavior is to check all volumes when the computer is started.

/t[:*time*] : Changes the Autochk.exe initiation countdown time to the specified amount of time entered in seconds. If you do not specify **:***time*, **/t** displays the current countdown time.

/x *volume*: [...] : Excludes the specified volume from being checked when the computer starts, even if the volume is marked as requiring **chkdsk**.

/c *volume*: [...] : Schedules the specified volume to be checked when the computer starts.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- To run **chkntfs**, you must be a member of the Administrators group.
- Although you can set the Autochk.exe initiation countdown time to zero, you cannot cancel a potentially time-consuming automatic file check if you set it to zero.

⇡Top of page

Examples

To display the Autochk.exe initiation countdown time, type:

chkntfs /t

When changing the Autochk.exe initiation countdown time, type the time in seconds. For example, to change the countdown time to 30 seconds, type:

chkntfs /t:30

The **/x** command-line option is not accumulative. If you type it more than once, the most recent entry overrides the previous entry. To exclude multiple volumes, list them all in one command. For example, to exclude both the E: and F: volumes, type:

chkntfs /x e: f:

To exclude the F: volume only, type:

chkntfs /x e:

chkntfs /x f:

The **/c** command-line option is accumulative. If you want to check the specified volume only, reset the defaults to clear any previous commands, exclude all volumes from being checked, and then schedule automatic file checking on the desired volume.

For example, to schedule automatic file checking on the D: volume only, type:

chkntfs /d

chkntfs /x c: d: e:

chkntfs /c d:

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Chkdsk

Chkdsk

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Cipher

Displays or alters the encryption of folders and files on NTFS volumes. Used without parameters, **cipher** displays the encryption state of the current folder and any files it contains.

Syntax

cipher [{**/e**|**/d**}] [**/s:***dir*] [**/a**] [**/i**] [**/f**] [**/q**] [**/h**] [**/k**] [**/u**[**/n**]] [*PathName* [...]] | [**/r:** *PathNameWithoutExtension*] | [**/w:***PathName*]

⇡Top of page

Parameters

/e : Encrypts the specified folders. Folders are marked so that files that are added to the folder later are encrypted too.

/d : Decrypts the specified folders. Folders are marked so that files that are added to the folder later are encrypted too.

/s: *dir* : Performs the selected operation in the specified folder and all subfolders.

/a : Performs the operation for files and directories.

/i : Continues performing the specified operation even after errors occur. By default, **cipher** stops when it encounters an error.

/f : Forces the encryption or decryption of all specified objects. By default, **cipher** skips files that have been encrypted or decrypted already.

/q : Reports only the most essential information.

/h : Displays files with hidden or system attributes. By default, these files are not encrypted or decrypted.

/k : Creates a new file encryption key for the user running **cipher**. If you use this option, **cipher**

ignores all of the other options.

/u : Updates the user's file encryption key or recovery agent's key to the current ones in all of the encrypted files on local drives (that is, if the keys have been changed). This option only works with **/n**.

/n : Prevents keys from being updated. Use this option to find all of the encrypted files on the local drives. This option only works with **/u**.

PathName **:** Specifies a pattern, file, or folder.

/r:*PathNameWithoutExtension* **:** Generates a new recovery agent certificate and private key, and then writes them to files with the file name specified in *PathNameWithoutExtension*. If you use this option, **cipher** ignores all of the other options.

/w:*PathName* **:** Removes data on unused portions of a volume. *PathName* can indicate any directory on the desired volume. If you use this option, **cipher** ignores all of the other options.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Using **/w**

  /w removes data from portions of the volume it can access and have not been allocated to files or directories. It does not lock the drive, so other programs can obtain space on the drive, which **cipher** cannot erase. Because this option writes to a large portion of the hard volume, it might take a long time to complete and should only be used when necessary.
- Encrypting or decrypting files

  To prevent an encrypted file from becoming decrypted when it is modified, it is recommended that you encrypt both the file and the folder in which it resides.
- Using read-only files and folders

  Cipher cannot encrypt files that are marked as read-only.
- Using multiple folder names

  You can use multiple folder names and wildcard characters.

- Using multiple parameters

You must separate multiple parameters by at least one space.

Top of page

Examples

To use **cipher** to encrypt a subfolder named May in a folder named MonthlyReports, type:

```
cipher /e monthlyreports\may
```

To encrypt the MonthlyReports folder, the January through December subfolders, and the Manufacturing subfolders within the month subfolders, type:

```
cipher /e /s:monthlyreports
```

To encrypt only the Marketing.xls file in the May subfolder, type:

```
cipher /e /a monthlyreports\may\marketing.xls
```

To encrypt the Marketing.xls file, the Maintenance.doc file, and the Manufacturing subfolder (located in the May folder), type:

```
cipher /e /a monthlyreports\may\ma*
```

To determine whether the May folder is encrypted, type:

```
cipher monthlyreports\may
```

To determine which files in the May folder are encrypted, type:

```
cipher monthlyreports\may\*
```

Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Cls

Clears the Command Prompt window.

Syntax

cls

Top of page

Parameters

/? : Displays help at the command prompt.

Top of page

Examples

To clear all information that appears in the Command Prompt window and return to a blank window, type:

cls

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Cmd

Starts a new instance of the command interpreter, Cmd.exe. Used without parameters, **cmd** displays Windows XP version and copyright information.

Syntax

cmd [[{**/c**|**/k**}] [**/s**] [**/q**] [**/d**] [{**/a**|**/u**}] [**/t:***fg*] [**/e:**{**on**|**off**}] [**/f:**{**on**|**off**}] [**/v:**{**on**|**off**}] *string*]

⇡Top of page

Parameters

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

/k : Carries out the command specified by *string* and continues.

/s : Modifies the treatment of *string* after **/c** or **/k**.

/q : Turns the echo off.

/d : Disables execution of AutoRun commands.

/a : Creates American National Standards Institute (ANSI) output.

/u : Creates Unicode output.

/t:*fg* **:** Sets the foreground *f* and background *g* colors. The following tables lists valid hexadecimal digits that you can use as the values for *f* and *g*.

Value	Color
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

/e:on : Enables command extensions.

/e:off : Disables commands extensions.

/f:on : Enables file and directory name completion.

/f:off : Disables file and directory name completion.

/v:on : Enables delayed environment variable expansion.

/v:off : Disables delayed environment variable expansion.

***string* :** Specifies the command you want to carry out.

/? : Displays help at the command prompt.

Top of page

Remarks

- Using multiple commands

  You can use multiple commands separated by the command separator **&&** for *string*, but you must enclose them in quotation marks (for example, **"***command***&&***command***&&***command***"**).
- Processing quotation marks

  If you specify **/c** or **/k**, **cmd** processes the remainder of *string* and quotation marks are preserved only if all of the following conditions are met:

  - You do not use **/s**.
  - You use exactly one set of quotation marks.
  - You do not use any special characters within the quotation marks (for example: &<>( ) @ ^ |).
  - You use one or more white-space characters within the quotation marks.
  - The *string* within quotation marks is the name of an executable file.

  If the previous conditions are not met, *string* is processed by examining the first character to verify whether or not it is an opening quotation mark. If the first character is an opening quotation mark, it is stripped along with the closing quotation mark. Any text following the closing quotation marks is preserved.
- Executing registry subkeys

  If you do not specify **/d** in *string*, Cmd.exe looks for the following registry subkeys:

  HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor\AutoRun\REG_SZ

  HKEY_CURRENT_USER\Software\Microsoft\Command Processor \AutoRun REG_EXPAND_SZ

  If either one or both registry subkeys are present, they are executed before all other variables.

  Caution

  - Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

- Enabling and disabling command extensions

Command extensions are enabled by default in Windows XP. You can disable them for a particular process by using **/e:off**. You can enable or disable extensions for all **cmd** command-line options on a computer or user session by setting the following **REG_DWORD** values:

HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor\EnableExtensions \REG_DWORD

HKEY_CURRENT_USER\Software\Microsoft\Command Processor\EnableExtensions \REG_DWORD

Set the **REG_DWORD** value to either **0×1** (that is, enabled) or **0×0** (that is, disabled) in the registry by using Regedit.exe. User-specified settings take precedence over computer settings, and command-line options take precedence over registry settings.

Caution

- Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

When you enable command extensions, the following commands are affected:

- **assoc**
- **call**
- **chdir** (**cd**)
- **color**
- **del** (**erase**)
- **endlocal**
- **for**
- **ftype**
- **goto**
- **if**
- **mkdir** (**md**)
- **popd**
- **prompt**
- **pushd**
- **set**
- **setlocal**
- **shift**
- **start** (also includes changes to external command processes)

For more information about these commands, see Related Topics.

- Enabling delayed environment variable expansion

  If you enable delayed environment variable expansion, you can use the exclamation character to substitute the value of an environment variable at run time.

- Enabling file and directory name completion

  File and directory name completion is not enabled by default. You can enable or disable file name completion for a particular process of the **cmd** command with **/f:**{**on**|**off**}. You can enable or disable file and directory name completion for all processes of the **cmd** command on a computer or user logon session by setting the following **REG_DWORD** values:

  HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor\CompletionChar \REG_DWORD

  HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor \PathCompletionChar\REG_DWORD

  HKEY_CURRENT_USER\Software\Microsoft\Command Processor\CompletionChar \REG_DWORD

  HKEY_CURRENT_USER\Software\Microsoft\Command Processor\PathCompletionChar \REG_DWORD

  To set the **REG_DWORD** value, run Regedit.exe and use the hexadecimal value of a control character for a particular function (for example, **0×9** is TAB and **0×08** is BACKSPACE). User-specified settings take precedence over computer settings, and command-line options take precedence over registry settings.

  Caution

  - Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

  If you enable file and directory name completion by using **/f:on**, use CTRL+D for directory name completion and CTRL+F for file name completion. To disable a particular completion character in the registry, use the value for white space [**0×20**] because it is not a valid control character.

  When you press CTRL+D or CTRL+F, **cmd** processes file and directory name completion. These key combination functions append a wildcard character to *string* (that is, if one is not present), build a list of paths that match, and then display the first matching path. If none of the paths match, the file and directory name completion function beeps and does not change the display. To move through the list of matching paths, press CTRL+D or CTRL+F repeatedly. To move through the list backwards, press the SHIFT key and CTRL+D or CTRL+F simultaneously. To discard the saved list of matching paths and generate a new list, edit *string* and press CTRL+D or CTRL+F. If you switch between CTRL+D and CTRL+F, the saved list of matching paths is discarded and a new list is generated. The only difference between the key combinations CTRL+D and CTRL+F is that CTRL+D only matches directory names and CTRL+F matches both file and directory names. If you use file and directory name completion on any of the built-in directory commands (that is, **CD**, **MD**, or **RD**), directory completion is assumed.

File and directory name completion correctly processes file names that contain white space or special characters if you place quotation marks around the matching path.

The following special characters require quotation marks: & < > [ ] { } ^ = ; ! ' + , ` ~ [white space]

If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

If you process file and directory name completion from within *string*, any part of the [*Path*] to the right of the cursor is discarded (that is, at the point in *string* where the completion was processed).

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Assoc

Call

Chdir

Color

Del (erase)

Endlocal

For

Ftype

Goto

If

Mkdir

Popd

Prompt

Pushd

Set

Setlocal

Shift

Start

Command-line reference A-Z

Command shell overview

Cmstp

Installs or removes a Connection Manager service profile. Used without optional parameters, **cmstp** installs a service profile with default settings appropriate to the operating system and to the user's permissions.

Syntax

Syntax 1: *ServiceProfileFileName***.exe /q:a /c:"cmstp.exe** *ServiceProfileFileName***.inf** [**/nf**] [**/ni**] [**/ns**] [**/s**] [**/su**] [**/u**]**"**

Syntax 2: cmstp.exe [**/nf**] [**/ni**] [**/ns**] [**/s**] [**/su**] [**/u**] **"**[*Drive***:**][*Path*]*ServiceProfileFileName***.inf"**

⇡Top of page

Parameters

***ServiceProfileFileName*.exe :** Required for Syntax 1 but not valid for Syntax 2. Specifies, by name, the installation package that contains the profile that you want to install.

/q:a : Required for Syntax 1 but not valid for Syntax 2. Specifies that the profile should be installed without prompting the user. The verification message that the installation has succeeded will still appear.

[*Drive*:][*Path*]*ServiceProfileFileName*.inf : Required. Specifies, by name, the configuration file that determines how the profile should be installed. The [*Drive***:**][*Path*] parameter is not valid for Syntax 1.

/nf : Specifies that the support files should not be installed.

/ni : Specifies that a desktop icon should not be created. This parameter is only valid for computers running Windows 95, Windows 98, Windows NT 4.0, or Windows Millennium Edition.

/ns : Specifies that a desktop shortcut should not be created. This parameter is only valid for computers running Windows 2000 or Windows XP.

/s : Specifies that the service profile should be installed or uninstalled silently (without prompting for user response or displaying verification message).

/su : Specifies that the service profile should be installed for a single user rather than for all users. This parameter is only valid for computers running Windows 2000 or Windows XP.

/u : Specifies that the service profile should be uninstalled.

/? : Displays help at the command prompt.

Top of page

Remarks

- **/s** is the only parameter that you can use in combination with **/u**.
- Syntax 1 is the typical syntax used in a custom installation application. To use this syntax, you must run **cmstp** from the directory that contains the *ServiceProfileFileName*.exe file.

Top of page

Examples

The following examples show possible uses of **cmstp** in relation to a fictional service profile called Fiction and its associated files.

- To install the Fiction service profile without any support files, type:

  fiction.exe /c:"cmstp.exe fiction.inf /nf"
- To silently install the Fiction service profile for a single user, type:

  fiction.exe /c:"cmstp.exe fiction.inf /s /su"
- To silently uninstall the Fiction service profile, type:

  fiction.exe /c:"cmstp.exe fiction.inf /s /u"

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command shell overview

Top of page

Manage Your Profile |Legal |Contact Us

Color

Changes the Command Prompt window foreground and background colors for the current session. Used without parameters, **color** restores the default Command Prompt window foreground and background colors.

Syntax

color *[fb]*

↑Top of page

Parameters

***fb* :** Sets the foreground *f* and background *g* colors. The following tables lists valid hexadecimal digits that you can use as the values for *f* and *g*.

Value	Color
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

/? : Displays help at the command prompt.

Top of page

Remarks

- To set the default Command Prompt window color, click the upper-left corner of the command prompt window, click **Defaults**, click the **Colors** tab, and then click the colors that you want to use for the **Screen Text** and **Screen Background**.
- You can use **cmd /t:***fb* to change the Command Prompt window colors for the current session.
- If the foreground and background colors are the same, the **color** command sets ERRORLEVEL to one (1).

Top of page

Examples

To change the Command Prompt window foreground color to red and background to white, type:

color FC

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply

Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Cmd

Top of page

Manage Your Profile |Legal |Contact Us

Command shell overview

The command shell is a separate software program that provides direct communication between the user and the operating system. The non-graphical command shell user interface provides the environment in which you run character-based applications and utilities. The command shell executes programs and displays their output on the screen by using individual characters similar to the MS-DOS command interpreter Command.com. The Windows XP command shell uses the command interpreter Cmd.exe, which loads applications and directs the flow of information between applications, to translate user input into a form that the operating system understands.

You can use the command shell to create and edit batch files (also called scripts) to automate routine tasks. For example, you can use scripts to automate the management of user accounts or nightly backups. You can also use the Windows Script Host, CScript.exe, to run more sophisticated scripts in the command shell. You can perform operations more efficiently by using batch files than you can by using the user interface. Batch files accept all commands that are available at the command line. For more information about batch files and scripting, see Using batch files

You can customize the command prompt window for easier viewing and to increase control over how you run programs. For more information about customizing the command prompt window, see To configure the command prompt

Using command syntax

Syntax appears in the order in which you must type a command and any parameters that follow it. The following example of the **xcopy** command illustrates a variety of syntax text formats:

xcopy *Source* [*Destination*] [**/w**] [**/p**] [**/c**] [**/v**] [**/q**] [**/f**] [**/l**] [**/g**] [**/d**[**:***mm-dd-yyyy*]] [**/u**] [**/i**] [**/s** [**/e**]] [**/t**] [**/k**] [**/r**] [**/h**] [{**/a**|**/m**}] [**/n**] [**/o**] [**/x**] [**/exclude:***file1*[+[*file2*]][+[*file3*]] [{**/y**|**/-y**}] [**/z**]

The following table explains how to interpret the different text formats.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one

Courier font	Code or program output

Top of page

Using multiple commands and conditional processing symbols

You can run multiple commands from a single command line or script using conditional processing symbols. When you run multiple commands with conditional processing symbols, the commands to the right of the conditional processing symbol act based upon the results of the command to the left of the conditional processing symbol. For example, you might want to run a command only if the previous command fails. Or, you might want to run a command only if the previous command is successful.

You can use the special characters listed in the following table to pass multiple commands.

Character	Syntax	Definition
& [...]	*command1* **&** *command2*	Use to separate multiple commands on one command line. Cmd.exe runs the first command, and then the second command.
&& [...]	*command1* **&&** *command2*	Use to run the command following **&&** only if the command preceding the symbol is successful. Cmd.exe runs the first command, and then runs the second command only if the first command completed successfully.
|| [...]	*command1* || *command2*	Use to run the command following || only if the command preceding || fails. Cmd.exe runs the first command, and then runs the second command only if the first command did not complete successfully (receives an error code greater than zero).
( ) [...]	**(***command1* **&** *command2***)**	Use to group or nest multiple commands.
; or **,**	*command1 parameter1***;***parameter2*	Use to separate command parameters.

Note

- The ampersand (&), pipe (|), and parentheses ( ) are special characters that must be preceded by the escape character (^) or quotation marks when you pass them as arguments.
- If a command completes an operation successfully, it returns an exit code of zero (0) or no exit code. For more information about exit codes, see Microsoft Windows Resource Kits

Top of page

Nesting command shells

You can nest command shells within Cmd.exe by opening a new instance of Cmd.exe at the command prompt. By default, each instance of Cmd.exe inherits the environment of its parent Cmd.exe application. By nesting instances of Cmd.exe, you can make changes to the local environment without affecting the parent application of Cmd.exe. This allows you to preserve the original environment of Cmd.exe and return to it after you terminate the nested command shell. The changes you make in the nested command shell are not saved.

To nest a command shell, at the command prompt, type:

cmd

A message similar to the following appears:

```
Microsoft (R) Windows XP (TM)
(C) Copyright 1985-2001 Microsoft Corp.
```

To close the nested command shell, type **exit**.

You can localize changes even further in an instance of Cmd.exe (or in a script) by using the **setlocal** and **endlocal** commands. **Setlocal** creates a local scope and **endlocal** terminates the local scope. Any changes made within the **setlocal** and **endlocal** scope are discarded, thereby leaving the original environment unchanged. You can nest these two commands to a maximum of 32 levels. For more information about the **setlocal** and **endlocal** commands, see Setlocal and Endlocal

⇡Top of page

Using environment variables with Cmd.exe

The Cmd.exe command-shell environment is defined by variables that determine the behavior of the command shell and the operating system. You can define the behavior of the command-shell environment or the entire operating system environment by using two types of environment variables, system and local. System environment variables define the behavior of the global operating system environment. Local environment variables define the behavior of the environment of the current instance of Cmd.exe.

System environment variables are preset in the operating system and available to all Windows XP processes. Only users with administrative privileges can change system variables. These variables are most commonly used in logon scripts.

Local environment variables are only available when the user for whom they were created is logged on to the computer. Local variables set in the **HKEY_CURRENT_USER** hive are valid only for the current user, but define the behavior of the global operating system environment.

The following list describes the various types of variables in descending order of precedence:

1. Built-in system variables
2. System variables found in the **HKEY_LOCAL_MACHINE** hive
3. Local variables found in the **HKEY_CURRENT_USER** hive
4. All environment variables and paths set in the Autoexec.bat file
5. All environment variables and paths set in a logon script (if present)
6. Variables used interactively in a script or batch file

In the command shell, each instance of Cmd.exe inherits the environment of its parent application. Therefore, you can change the variables in the new Cmd.exe environment without affecting the environment of the parent application.

The following table lists the system and local environment variables for Windows XP.

Variable	Type	Description
%ALLUSERSPROFILE%	Local	Returns the location of the All Users Profile.
%APPDATA%	Local	Returns the location where applications store data by default.
%CD%	Local	Returns the current directory string.
%CMDCMDLINE%	Local	Returns the exact command line used to start the current Cmd.exe.
%CMDEXTVERSION%	System	Returns the version number of the current Command Processor Extensions.
%COMPUTERNAME%	System	Returns the name of the computer.
%COMSPEC%	System	Returns the exact path to the command shell executable.
%DATE%	System	Returns the current date. Uses the same format as the **date /t** command. Generated by Cmd.exe. For more information about the **date** command, see Date
%ERRORLEVEL%	System	Returns the error code of the most recently used command. A non zero value usually indicates an error.
%HOMEDRIVE%	System	Returns which local workstation drive letter is connected to the user's home directory. Set based on the value of the home directory. The user's home directory is specified in Local Users and Groups.
%HOMEPATH%	System	Returns the full path of the user's home directory. Set based on the value of the home directory. The user's home directory is specified in Local Users and Groups.
%HOMESHARE%	System	Returns the network path to the user's shared home directory. Set based on the value of the home directory. The user's home directory is specified in Local Users and Groups.
%LOGONSEVER%	Local	Returns the name of the domain controller that validated the current logon session.
%NUMBER_OF_PROCESSORS%	System	Specifies the number of processors installed on the computer.
%OS%	System	Returns the operating system name. Windows 2000 displays the operating system as Windows_NT.
%PATH%	System	Specifies the search path for executable files.
%PATHEXT%	System	Returns a list of the file extensions that the operating system considers to be executable.
%PROCESSOR_ARCHITECTURE%	System	Returns the chip architecture of the processor. Values: x86, IA64.

%PROCESSOR_IDENTFIER%	System	Returns a description of the processor.
%PROCESSOR_LEVEL%	System	Returns the model number of the processor installed on the computer.
%PROCESSOR_REVISION%	System	Returns the revision number of the processor.
%PROMPT%	Local	Returns the command prompt settings for the current interpreter. Generated by Cmd.exe.
%RANDOM%	System	Returns a random decimal number between 0 and 32767. Generated by Cmd.exe.
%SYSTEMDRIVE%	System	Returns the drive containing the Windows XP root directory (that is, the system root).
%SYSTEMROOT%	System	Returns the location of the Windows XP root directory.
%TEMP% and %TMP%	System and User	Returns the default temporary directories that are used by applications available to users who are currently logged on. Some applications require TEMP and others require TMP.
%TIME%	System	Returns the current time. Uses the same format as the **time /t** command. Generated by Cmd.exe. For more information about the **time** command, see Time
%USERDOMAIN%	Local	Returns the name of the domain that contains the user's account.
%USERNAME%	Local	Returns the name of the user who is currently logged on.
%USERPROFILE%	Local	Returns the location of the profile for the current user.
%WINDIR%	System	Returns the location of the operating system directory.

Top of page

Setting environment variables

Use the **set** command to create, change, delete, or display environment variables. The **set** command alters variables in the current shell environment only.

To view a variable, at a command prompt, type:

set *VariableName*

To add a variable, at a command prompt, type:

set variablename=*value*

To delete a variable, at a command prompt, type:

set *VariableName*=

You can use most characters as variable values, including white space. If you use the special characters <, >, |, &, or ^, you must precede them with the escape character (^) or quotation marks. If you use quotation marks, they are included as part of the value because everything following the equal sign is taken as the value. Consider the following examples:

- To create the variable value **new&name**, type:

  set varname=new^&name
- To create the variable value **"new&name"**, type:

  set varname="new&name"
- If you type **set varname=new&name** at the command prompt, an error message similar to the following appears:

```
"'name' is not recognized as an internal or external command, operable program or
batch file."
```

Variable names are not case-sensitive. However, **set** displays the variable exactly as you typed it. You can combine uppercase and lowercase letters in your variable names to make your code more readable (for example, UserName).

Note

- The maximum individual environment variable size is 8192bytes.
- The maximum total environment variable size for all variables, which includes variable names and the equal sign, is 65,536KB.

Top of page

Substituting environment variable values

To enable the substitution of variable values at the command line or in scripts, enclose the variable name in percent signs (that is, **%***variablename***%**). By using percent signs, you ensure that Cmd.exe references the variable values instead of making a literal comparison. After you define variable values for a variable name, enclose the variable name in percent signs. Cmd.exe searches for all instances of the variable name and replaces it with the defined variable value. For example, if you create a script that contains different values (for example, user names) and you want to define the USERNAME environment variable for each user with these values, you can write one script using the variable USERNAME enclosed in percent signs. When you run this script, Cmd.exe replaces %USERNAME% with the variable values, which eliminates the need to perform this task manually for each user. Variable substitution is not recursive. Cmd.exe checks variables once. For more information about variable substitution, see For and Call

Top of page

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, and in the same directory or in different directories. When **comp** compares the files, it displays their locations and file names. Used without parameters, **comp** prompts you to enter the files to compare.

Syntax

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

Top of page

Parameters

data1 **:** Specifies the location and name of the first file or set of files you want to compare. You can use wildcard characters (* 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 wildcard characters (* 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.

/? : Displays help at the command prompt.

Top of page

Remarks

- Comparing files with the same names

  The files that you compare can have the same file name, provided they are in different directories or on different drives. If you do not specify a file name for *data2*, the default file name for *data2* is the same as the file name in *data1*. You can use wildcard characters (that is, * and ?) to specify file names.
- 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 file name, **comp** compares all of 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 file name 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 (that is, in hexadecimal notation unless you specify the **/a** or **/d** command-line option). The message appears in 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** command-line option. 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 command-line options 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** command-line option to compare only the first portion of each file.
- Comparing files sequentially

  If you use wildcard characters (* and ?) 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, and 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 command-line options to use. If you don't specify any command-line options, **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.

Top of page

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

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Diskcomp

Fc

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Compact

Displays and alters the compression of files or directories on NTFS partitions. Used without parameters, **compact** displays the compression state of the current directory.

Syntax

compact [{**/c**|**/u**}] [**/s**[**:***dir*]] [**/a**] [**/i**] [**/f**] [**/q**] [*FileName*[...]]

Top of page

Parameters

/c : Compresses the specified directory or file.

/u : Uncompresses the specified directory or file.

/s:*dir* : 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.

/a : Displays hidden or system files.

/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. To force the file to be compressed in its entirety, use the **/c** and **/f** parameters and specify the partially compressed file.

/q : Reports only the most essential information.

***FileName* :** Specifies the file or directory. You can use multiple file names and wildcard characters (* and ?).

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- The **compact** command, the command-line version of the NTFS file system 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.

⇡Top of page

Examples

To set the compression state of the current folder and its subfolders and existing files, from the current folder, type:

compact /c /s

To set the compression state of files in the current folder, subfolders in the current folder, and files within all subfolders without altering the compression state of the current folder, from the current folder, type:

compact /c /s *.*

To compress a volume, from the root folder of the volume, type:

compact /c /i /s:\

This example sets the compression state of the root folder and all folders on the volume and compresses every file on the volume. By using the **/i** parameter, you ensure that error messages do not interrupt the compression process.

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

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Convert

Converts FAT and FAT32 volumes to NTFS

Syntax

convert [*volume*] **/fs:ntfs** [**/v**] [**/cvtarea:***FileName*] [**/nosecurity**] [**/x**]

⇡Top of page

Parameters

volume Specifies the drive letter (followed by a colon), mount point, or volume name to convert to NTFS. **/fs:ntfs** Required. Converts the volume to NTFS. **/v** Specifies verbose mode, that is, all messages will be displayed during conversion. **/cvtarea:***FileName* For advanced users only. Specifies that the Master File Table (MFT) and other NTFS metadata files are written to an existing, contiguous placeholder file. This file must be in the root directory of the file system to be converted. Use of the **/CVTAREA** parameter can result in a less fragmented file system after conversion. For best results, the size of this file should be 1 KB multiplied by the number of files and directories in the file system, however, the convert utility accepts files of any size.

For more information about using the **/cvtarea** parameter, see "File Systems" at the Microsoft Windows XP Resource Kits Web site.

Important

- You must create the placeholder file using the **fsutil file createnew** command prior to running **convert**. **Convert** does not create this file for you. **Convert** overwrites this file with NTFS metadata. After conversion, any unused space in this file is freed. For more information about the **fsutil file** command, see Related Topics.

/nosecurity Specifies that the converted files and directory security settings are accessible by everyone. **/x** Dismounts the volume, if necessary, before it is converted. Any open handles to the volume will no longer be valid.

⇡Top of page

Remarks

- You must specify that the drive should be converted when the computer is restarted. Otherwise, you cannot convert the current drive.
- If **convert** cannot lock the drive (for example, the system volume or the current drive), it offers to convert the drive the next time the computer restarts.
- The location of the MFT is different on volumes that have been converted from previous version of NTFS, so volume performance might not be as good on volumes converted from Windows NT.
- Volumes converted from FAT to NTFS lack some performance benefits compared to volumes initially formatted with NTFS. On converted volumes, the MFT might become fragmented. In addition, on converted boot volumes, NTFS permissions are not applied after the volume is converted.

↑Top of page

Examples

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

convert e: /fs:ntfs /v

Command-line reference A-Z

Fsutil: file

↑Top of page

Manage Your Profile |Legal |Contact Us

Copy

Copies one or more files from one location to another.

Syntax

copy [**/d**] [**/v**] [**/n**] [{**/y**|**/-y**}] [**/z**] [{**/a**|**/b**}] *Source* [{**/a**|**/b**}] [**+** *Source* [{**/a**|**/b**}] [**+** ...]] [*Destination* [{**/a**|**/b**}]]

⇡Top of page

Parameters

/d : Allows the encrypted files being copied to be saved as decrypted files at the destination.

/v : Verifies that new files are written correctly.

/n : Uses a short file name, if available, when copying a file with a name longer than eight characters, or with a file extension longer than three characters.

/y : Suppresses prompting to confirm that you want to overwrite an existing destination file.

/-y : Prompts you to confirm that you want to overwrite an existing destination file.

/z : Copies networked files in restartable mode.

/a : Indicates an ASCII text file.

/b : Indicates a binary file.

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

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

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- You can copy an ASCII text file that uses an end-of-file character (that is, CTRL+Z) to indicate the end of the file.
- Using **/a**

  When **/a** precedes a list of files on the command line, it applies to all files listed until **copy** encounters **/b**. In this case, **/b** applies to the file preceding **/b**.

  When **/a** follows a list of files on the command line, it applies to all listed files until **copy** encounters **/b**. In this case, **/b** applies to the file preceding **/b**.

  The effect of **/a** depends on its position in the command-line string. When **/a** follows *Source*, **copy** treats the file as an ASCII file and copies data that precedes the first end-of-file character.

  When **/a** follows *Destination*, **copy** adds an end-of-file character as the last character of the file.
- Using **/b**

  /b directs the command interpreter to read the number of bytes specified by the file size in the directory. **/b** is the default value for **copy**, unless **copy** combines files.

  When **/b** precedes a list of files on the command line, it applies to all listed files until **copy** encounters **/a**. In this case, **/a** applies to the file preceding **/a**.

  When **/b** follows a list of files on the command line, it applies to all listed files until **copy** encounters **/a**. In this case, **/a** applies to the file preceding **/a**.

  The effect of **/b** depends on its position in the commandline string. When **/b** follows *Source*, **copy** copies the entire file, including any end-of-file character.

  When **/b** follows *Destination*, **copy** does not add an end-of-file character.

- Using **/v**

  If a write operation cannot be verified an error message appears. Although recording errors rarely occur with **copy**, you can use **/v** to verify that critical data has been correctly recorded. The **/v** command-line option also slows down the **copy** command, because each sector recorded on the disk must be checked.
- Using **/y** and **/-y**

  If **/y** is preset in the COPYCMD environment variable, you can override this setting by using **/-y** at the command line. By default, you are prompted when you replace this setting, unless the **copy** command is executed in a batch script.
- Appending files

  To append files, specify a single file for *Destination*, but multiple files for *Source* (using wildcard characters or *file1*+*file2*+*file3* format).
- Using **/z**

  If the connection is lost during the copy phase (for example, if the server going offline severs the connection), **copy /z** resumes after the connection is reestablished. **/z** also displays the percentage of the copy operation that is completed for each file.
- Copying to and from devices

  You can substitute a device name for one or more occurrences of *Source* or *Destination*.
- Using or omitting **/b** when copying to a device

  When *Destination* is a device (for example, Com1 or Lpt1), **/b** copies data to the device in binary mode. In binary mode, **copy /b** copies all characters (that is, including special characters, such as CTRL+C, CTRL+S, CTRL+Z, and carriage return) to the device as data. However, if you omit **/b**, data is copied to the device in ASCII mode. In ASCII mode, special characters might cause Windows XP to combine files during the copying process. For more information, see "Combining files."
- Using the default destination file

  If you do not specify a destination file, a copy is created 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 displays the following error message:

  ```
  File cannot be copied onto itself

  0 File(s) copied
  ```

- Combining files

  If you specify more than one *Source*, separating entries with a plus sign (+), **copy** combines the files into a single file. If you use wildcard characters (that is, * or ?) in *Source*, but you specify a single file name in *Destination*, **copy** combines all files matching the file name in *Source* and creates a single file with the file name specified in *Destination*. In either case, **copy** assumes the combined files are ASCII files unless you use **/b**.
- 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 the following syntax:

  copy /b *Source*+,,

  The commas indicate the omission of the *Destination* parameter.
- Copying files in subdirectories

  To copy all of a directory's files and subdirectories, use the **xcopy** command. For information about **xcopy**, see Related Topics.
- The **copy** command, with different parameters, is available from the Recovery Console.

⇡Top of page

Examples

To copy a file called Memo.doc to Letter.doc in the current drive and ensure that an end-of-file character is at the end of the copied file, type:

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:

copy robin.typ c:\birds

If the Birds directory does not exist, the file Robin.typ is copied 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 several *Source* files, separate the file names with a plus sign

(+), and specify a *Destination* file that you want to contain the resulting combined file. For example, to combine Mar89.rpt, Apr89.rpt, and May89.rpt, which are located on the current drive and directory, and place them in a file named Report on the current drive and directory, type:

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

When you combine files, **copy** marks the destination file with the current date and time. If you omit *Destination*, the files are combined and stored under the name of the file specified first. For example, to combine all four files in Report when a file named Report already exists, type:

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

You can also combine several files into one file by using wildcard characters (that is, * or ?). For example, to combine all files in the current directory on the current drive that have the extension .txt into one file named Combin.doc, type:

copy *.txt combin.doc

If you want to combine several binary files into one file by using wildcard characters, include **/b**. This prevents Windows XP from treating CTRL+Z as an end-of-file character. For example, type:

copy /b *.exe combin.exe

Caution

- If you combine binary files, the resulting file might be unusable 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 file name but with a .doc extension. **Copy** combines File1.txt with File1.ref to form File1.doc, and then **copy** combines File2.txt with File2.ref to form File2.doc, and so on. For example, type:

copy *.txt + *.ref *.doc

To combine all files with the .txt extension, and then combine all files with the .ref extension into one file named Combin.doc, type:

copy *.txt + *.ref combin.doc

⇧Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Xcopy

Command-line reference A-Z

↑Top of page

Manage Your Profile |Legal |Contact Us

Cprofile

Cleans specified profiles of wasted space and, if user-specific file associations are disabled, removes these associations from the registry. Profiles that are currently in use are not modified.

Syntax

cprofile [**/l**] [**/i**] [**/v**] [*FileList*]

cprofile [**/i**] [**/v**] *FileList*

⇡Top of page

Parameters

/l : Cleans all local profiles. You can also specify a list of additional profiles in the *FileList* parameter.

/i : Interactively prompts the user with each profile.

/v : Displays information about the actions being performed.

***FileList* :** A list of files from which you want to remove user-specific file associations. Separate each file in the list with a space. File names can contain wildcard characters.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Only administrators can run **cprofile**.
- A terminal server uses file associations to determine which application to use to access files of various types. File types are registered using Windows Explorer.

- Per-user file associations allow each user to have a different application associated with a specific file type. For example, one user could have .doc files associated with Microsoft Word and another user could have .doc files associated with Windows WordPad.
- If user-specific file associations are enabled, **cprofile** only removes the unused space from the user profile. If user-specific file associations are disabled, **cprofile** also removes the corresponding registry entries.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Terminal Services commands

Top of page

Using the command-based script host (CScript.exe)

Cscript.exe is a command-line version of the Windows Script Host that provides command-line options for setting script properties.

With Cscript.exe, you can run scripts by typing the name of a script file at the command prompt. Like Microsoft Internet Explorer, Windows Script Host serves as a controller of Windows Script compliant scripting engines, but Windows Script Host has very low memory requirements. Windows Script Host is ideal for both interactive and non-interactive scripting needs, such as logon scripting and administrative scripting.

Windows Script Host supports scripts written in VBScript or JScript. When you start a script from your desktop or from the command prompt, the script host reads and passes the specified script file contents to the registered script engine. The script engine uses file extensions (that is, .vbs for VBScript and .js for JScript) to identify the script. As a result, you do not need to know the exact programmatic identifier (that is, the ProgID) of the script engine. The script host maintains a mapping of script extensions to programmatic identifiers, and uses the Windows XP association model to start the appropriate engine for a given script. For more information about Windows Script Host, see Windows Script Host For more information about CScript.exe, see To run scripts using the command-line-based script host (cscript.exe)

Top of page

Manage Your Profile |Legal |Contact Us

Date

Displays the current system date setting. Used without parameters, **date** displays the current system date setting and prompts you to type a new date.

Syntax

date [*mm-dd-yy*] [**/t**]

⇡Top of page

Parameters

***mm-dd-yy* :** Sets the date specified where *mm* is month, *dd* is day, and *yy* is year.

/t : Displays the current date without prompting you for a new date.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- You must separate values for *mm*, *dd*, and *yy* with periods (.), hyphens (-), or slash marks (/).
- Valid *mm* values are 1 through 12.
- Valid *dd* values are 1 through 31.
- Valid *yy* values are 80 through 99, or 1980 through 2099.
- Windows XP changes the month and year automatically, whether the month has 28, 29, 30, or 31 days.

⇡Top of page

Examples

To change the date to August 3, 2000, type any of the following:

08.03.00

08-03-00

08/03/00

To display the current system date, type:

date /t

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Time

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Defrag

Locates and consolidates fragmented boot files, data files, and folders on local volumes.

Syntax

defrag *volume*

defrag *volume* [/**a**]

defrag *volume* [/**a**] [/**v**]

defrag *volume* [/**v**]

defrag *volume* [/**f**]

Top of page

Parameters

***volume* :** The drive letter or a mount point of the volume to be defragmented.

/a : Analyzes the volume and displays a summary of the analysis report.

/v : Displays the complete analysis and defragmentation reports.

When used in combination with **/a**, displays only the analysis report. When used alone, displays both the analysis and defragmentation reports.

/f : Forces defragmentation of the volume regardless of whether it needs to be defragmented.

/? : Displays help at the command prompt.

Top of page

Remarks

- A volume must have at least 15% free space for **defrag** to completely and adequately defragment it. **Defrag** uses this space as a sorting area for file fragments. If a volume has less than 15% free space, **defrag** will only partially defragment it. To increase the free space on a volume, delete unneeded files or move them to another disk.
- You cannot defragment volumes that the file system has marked as dirty, which indicates possible corruption. You must run **chkdsk** on a dirty volume before you can defragment it. You can determine if a volume is dirty by using the **fsutil dirty query** command. For more information about **chkdsk** and **fsutil dirty**, see Related Topics.
- While **defrag** is analyzing and defragmenting a volume, it displays a blinking cursor. When **defrag** is finished analyzing and defragmenting the volume, it displays the analysis report, the defragmentation report, or both reports, and then exits to the command prompt.
- By default, **defrag** displays a summary of both the analysis and defragmentation reports if you do not specify the **/a** or **/v** parameters.
- You can send the reports to a text file by typing >*filename.txt*, where *filename.txt* is a file name you specify. For example:

  defrag *volume* **/v** >*filename.txt*
- To interrupt the defragmentation process, at the command line, press CTRL+C.
- Running the **defrag** command and Disk Defragmenter are mutually exclusive. If you are using Disk Defragmenter to defragment a volume and you run the **defrag** command at a command-line, the **defrag** command fails. Conversely, if you run the **defrag** command and open Disk Defragmenter, the defragmentation options in Disk Defragmenter are unavailable.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one

Courier font	Code or program output

Defrag

Command-line reference A-Z

Chkdsk

Chkdsk

Fsutil

Fsutil: dirty

Top of page

Manage Your Profile |Legal |Contact Us

Del (erase)

Deletes specified files.

Syntax

del [*Drive***:**][*Path*] *FileName* [...] [**/p**] [**/f**] [**/s**] [**/q**] [**/a**[**:***attributes*]]

erase [*Drive***:**][*Path*] *FileName* [...] [**/p**] [**/f**] [**/s**] [**/q**] [**/a**[**:***attributes*]]

Top of page

Parameters

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the file or set of files you want to delete. *Filename* is required. You can use multiple file names. Separate file names with 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. Displays the names of the files as they are being deleted.

/q : Specifies quiet mode. You are not prompted for delete confirmation.

/a : Deletes files based on specified attributes.

***attributes* :** Specifies any of the following file attributes:

Attribute	Description
r	Read-only
a	Archive

s	System
h	Hidden
-	Prefix meaning "not"

/? : Displays help at the command prompt.

Top of page

Remarks

- Using **/p**

  If you use **/p**, **del** displays the name of a file and sends the following message:

  filename, Delete (Y/N)?

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

  If you disable command extensions, the **/s** command-line option displays the names of any files that were not found, instead of displaying the names of files that are being deleted (that is, the behavior is reversed). For more information about disabling command extensions, see **cmd** in Related Topics.
- Deleting more than one file at a time

  You can delete all of the files in a folder by typing **del** followed by [*Drive*:]*Path*. For example, the following command deletes all files in the \Work folder:

  ```
  del \work
  ```

  You can also use wildcards (that is, * 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. For example, if you type the following command:

  ```
  del *.*
  ```

  The **del** command 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 folder, 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.

⚠ Warning

•After you delete a file from your disk using **del**, you cannot retrieve it.

•The **del** command, with different parameters, is available from the Recovery Console.

↑Top of page

Examples

To delete all the files in a folder named Test on drive C:\, type any of the following:

del c:\test

del c:\test\*.*

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one

Courier font	Code or program output

Cmd

Rmdir (rd)

Dir

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Dir

Displays a list of a directory's files and subdirectories. Used without parameters, **dir** displays the disk's volume label and serial number, followed by a list of directories and files on the disk, including their names and the date and time each was last modified. For files, **dir** displays the name extension and the size in bytes. **Dir** also displays the total number of files and directories listed, their cumulative size, and the free space (in bytes) remaining on the disk.

Syntax

dir [*Drive***:**][*Path*][*FileName*] [...] [**/p**] [**/q**] [**/w**] [**/d**] [**/a**[[**:**]*attributes*]][**/o**[[**:**]*SortOrder*]] [**/t**[[**:**] *TimeField*]] [**/s**] [**/b**] [**/l**] [**/n**] [**/x**] [**/c**] [**/4**]

Top of page

Parameters

[*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.

/p : Displays one screen of the listing at a time. To see the next screen, press any key on the keyboard.

/q : Displays file ownership information.

/w : Displays the listing in wide format, with as many as five file names or directory names on each line.

/d : Same as **/w** 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 **/a**, **dir** displays the names of all files except hidden and system files. If you use **/a** 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.

Value	Description
h	Hidden files
s	System files
d	Directories
a	Files ready for archiving
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 file names. If you omit **/o**, **dir** displays the names in the order in which they occur in the directory. If you use **/o** 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 white spaces.

Value	Description
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 (.ZZZ through .AAA)
-d	By date and time, latest first

-s	By size, largest first
-g	With directories grouped after files

/t [[:]*TimeField*] : Specifies which time field to display or use for sorting. The following list describes each of the values you can use for *TimeField*.

Value	Description
c	Creation
a	Last access
w	Last written

/s : Lists every occurrence, in the specified directory and all subdirectories, of the specified file name.

/b : Lists each directory name or file name, one per line, including the file name extension. **/b** does not display heading information or a summary. **/b** overrides **/w**.

/l : Displays unsorted directory names and file names in lowercase. **/l** does not convert extended characters to lowercase.

/n : Displays a long list format with file names on the far right of the screen.

/x : Displays the short names generated for files on NTFS and FAT volumes. The display is the same as the display for **/n**, but short names are displayed after the long name.

/c : Displays the thousand separator in file sizes.

/4 : Displays four-digit year format.

/? : Displays help at the command prompt.

Top of page

Remarks

- Using multiple *filenames*

  You can use multiple *filenames*. Separate file names with spaces, commas, or semicolons. You can use wildcard characters (that is, ? and *) in *FileName* to display a group of files.
- Using wildcards

  You can use wildcards (that is, ? and *) to display a list of a subset of files and subdirectories.
- Specifying file display attributes

  If you use **/a** with more than one value in *attributes*, **dir** displays the names of only those files with all the specified attributes. For example, if you use **/a** with **r** and **-h** 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 file name sorting

  If you specify more than one *SortOrder* value, **dir** sorts the file names by the first criterion first, then by the second criterion, and so on. For example, if you use **/o** 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 then displays the final result. The alphabetic sorting by extension causes file names with no extensions to appear first, then directory names, and then file names 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 **/a:-d** and **/b** to list the file names only. You can use *FileName* with **/b** and **/s** to specify that **dir** is to search the current directory and its subdirectories for all file names that match *FileName*. **Dir** lists only the drive letter, directory name, file name, and file name extension, one path per line, for each file name it finds. Before you use a pipe for redirection, you should set the TEMP environment variable in your Autoexec.nt file.
- Presetting **dir** parameters

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

  For example, to use the DIRCMD environment variable to set the wide display format (that is, **/w**) as the default format, type the following command in your Autoexec.nt file:

  set dircmd=/w

  For a single use of the **dir** command, you can override a parameter by using the DIRCMD environment variable. To do so, type the parameter that you want to override at the **dir** command prompt, preceding the parameter with a minus sign. For example:

dir /-w

To change the DIRCMD default settings, type:

set=*NewParameter*

The new default settings are effective for all subsequent **dir** commands until you use **set dircmd** again or until you restart your computer.

To clear all default settings, type:

set dircmd=

To view the current settings of the DIRCMD environment variable, type:

set

Set displays a list of environment variables and their settings. For more information about setting environment variables, see Related Topics.

- The **dir** command, with different parameters, is available from the Recovery Console.

Top of page

Examples

To display all directories, one after the other, in alphabetical order, in wide format and pausing after each screen, make sure that the root directory is the current directory, and then type:

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 file names in each subdirectory in the tree.

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

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

To print a directory listing, type:

```
dir > prn
```

When you specify **prn**, the directory list 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 file name. You can also type a path. For example, to direct **dir** output to the file Dir.doc in the Records directory, type:

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

If Dir.doc does not exist, **dir** creates it, unless the Records directory does not exist. In that case, the following message appears:

```
File creation error
```

To display a list of all the file names with the .txt extension in all directories on drive C, type:

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

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

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Attrib

Tree

Set

Using command redirection operators

Using wildcard characters

Top of page

Manage Your Profile |Legal |Contact Us

Diskcomp

Compares the contents of two floppy disks. Used without parameters, **diskcomp** uses the current drive for both disks that you want to compare.

Syntax

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

⇡Top of page

Parameters

drive1 **:** Specifies the drive containing one of the floppy disks.

drive2 **:** Specifies the drive containing the other floppy disk.

/? **:** Displays help at the command prompt.

⇡Top of page

Remarks

- Using disks

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 nonremovable
```

- Comparing disks

  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

  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 the following error message:

  Invalid drive specification
- Comparing an original disk with a copy

  When you use **diskcomp** with a disk that you made by using **copy**, **diskcomp** might 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 **copy** duplicates information, it does not necessarily place it in the same location on the destination disk. For more information about comparing individual files on two disks by using **Fc**, see Related Topics.
- Understanding **diskcomp** exit codes

  The following table lists each exit code and a brief description.

Exit code	Description
0	Disks are the same
1	Differences were found
3	Hard error occurred
4	Initialization error occurred

  To process exit codes returned by **diskcomp**, you can use *errorlevel* on the **if** command line in a batch program.

⇡Top of page

Examples

If your computer has only one floppy disk drive (for example, drive A), and you want to compare two disks, type:

diskcomp a: a:

Diskcomp prompts you to insert each disk, as needed.

The following example illustrates how to process a **diskcomp** exit code in a batch program that uses the *errorlevel* parameter on the **if** command line:

```
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
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Comp

If

Subst

Fc

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Diskcopy

Copies the contents of the floppy disk in the source drive to a formatted or unformatted floppy disk in the destination drive. Used without parameters, **diskcopy** uses the current drive for the source disk and the destination disk.

Syntax

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

⇡Top of page

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. This command-line option slows down the copying process.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Using disks

  Diskcopy 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 nonremovable
  ```

  The **diskcopy** command prompts you to insert the source and destination disks and waits for you to press any key on the keyboard 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.
- Using one drive for copying

  If *drive1* and *drive2* are the same, **diskcopy** prompts you to switch disks. If you omit both *Drive* parameters and the current disk drive is a floppy disk drive, **diskcopy** prompts you each time you need to 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. **Diskcopy** reads from the source disk, writes to the destination disk, and prompts you to insert the source disk again. This process continues until you have copied the entire disk.

- 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 process of finding, reading, or writing files. To avoid transferring fragmentation from one disk to another, use **copy** or **xcopy** to copy your disk. Because **copy** and **xcopy** copy files sequentially, the new disk is not fragmented.

  ⚠ Warning

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

- Understanding **diskcopy** exit codes

  The following table lists each exit code and a brief description.

Exit code	Description
0	Copy operation was successful
1	Nonfatal read/write error occurred
3	Fatal hard error occurred
4	Initialization error occurred

  To process exit codes returned by **diskcomp**, you can use the *errorlevel* on the **if** command line in a batch program. For an example of a batch program that processes exit codes, see **diskcomp** in Related Topics.

Top of page

Examples

To copy the disk in drive B to the disk in drive A, type:

diskcopy b: a:

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Copy

Xcopy

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

DiskPart

DiskPart.exe is a text-mode command interpreter that enables you to manage objects (disks, partitions, or volumes) by using scripts or direct input from a command prompt. Before you can use DiskPart.exe commands on a disk, partition, or volume, you must first list and then select the object to give it focus. When an object has focus, any DiskPart.exe commands that you type act on that object.

You can list the available objects and determine an object's number or drive letter by using the **list disk**, **list volume**, and **list partition** commands. The **list disk** and **list volume** commands display all disks and volumes on the computer. However, the **list partition** command only displays partitions on the disk that has focus. When you use the **list** commands, an asterisk (*) appears next to the object with focus. You select an object by its number or drive letter, such as disk 0, partition 1, volume 3, or volume C.

When you select an object, the focus remains on that object until you select a different object. For example, if the focus is set on disk 0, and you select volume 8 on disk 2, the focus shifts from disk 0 to disk 2, volume 8. Some commands automatically change the focus. For example, when you create a new partition, the focus automatically switches to the new partition.

You can only give focus to a partition on the selected disk. When a partition has focus, the related volume (if any) also has focus. When a volume has focus, the related disk and partition also have focus if the volume maps to a single specific partition. If this is not the case, then focus on the disk and partition is lost.

DiskPart commands

To view the command syntax, click a command:

active

On basic disks, marks the partition with focus as active. This informs the basic input/output system (BIOS) or Extensible Firmware Interface (EFI) that the partition or volume is a valid system partition or system volume

Only partitions can be marked as active.

Important

- **DiskPart** verifies only that the partition is capable of containing an operating system's startup files. **DiskPart** does not check the contents of the partition. If you mistakenly mark a partition as "active" and it does not contain the operating system's startup files, your computer might not start.

⇡Top of page

Syntax

active

add disk

Mirrors the simple volume with focus to the specified disk.

⇡Top of page

Syntax

add disk=*n* [**noerr**]

⇡Top of page

Parameters

***n* :** Specifies the disk to contain the mirror You can mirror only simple volumes. The specified disk must have unallocated space at least as large as the size of the simple volume you want to mirror.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

assign

Assigns a drive letter or mount point to the volume with focus. If no drive letter or mount point is specified, then the next available drive letter is assigned. If the drive letter or mount point is already in use, an error is generated.

By using the **assign** command, you can change the drive letter associated with a removable drive.

You cannot assign drive letters to system volumes, boot volumes, or volumes that contain the paging file. In addition, you cannot assign a drive letter to an Original Equipment Manufacturer (OEM) partition or any GUID Partition Table (GPT) partition other than a basic MSDATA partition.

↑Top of page

Syntax

assign [{**letter=***d*|**mount=***Path*}] [**noerr**]

↑Top of page

Parameters

letter=*d* : The drive letter you want to assign to the volume.

mount=*Path* : The mount point path you want to assign to the volume.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

break disk

Applies to dynamic disks only. Breaks the mirrored volume with focus into two simple volumes. One simple volume retains the drive letter and any mount points of the mirrored volume, while the other simple volume receives the focus so you can assign it a drive letter.

By default, the contents of both halves of the mirror are retained. Each half becomes a simple volume. By using the **nokeep** parameter, you retain only one half of the mirror as a simple volume, while the other half is deleted and converted to free space. Neither volume receives the focus.

↑Top of page

Syntax

break disk=*n* [**nokeep**] [**noerr**]

Top of page

Parameters

***n* :** Specifies the disk that contains the mirrored volume.

nokeep : Specifies that only one of the mirrored volumes is retained; the other simple volume is deleted and converted to free space. Neither the volume nor the free space receive the focus.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

clean

Removes any and all partition or volume formatting from the disk with focus. On master boot record (MBR) disks, only the MBR partitioning information and hidden sector information are overwritten. On GUID Partition Table (GPT) disks, the GPT partitioning information, including the Protective MBR, is overwritten. There is no hidden sector information.

Top of page

Syntax

clean [**all**]

Top of page

Parameter

all : Specifies that each and every sector on the disk is zeroed, which completely deletes all data contained on the disk.

convert basic

Converts an empty dynamic disk into a basic disk

Top of page

Syntax

convert basic [**noerr**]

Top of page

Parameter

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

convert dynamic

Converts a basic disk into a dynamic disk

Top of page

Syntax

convert dynamic [**noerr**]

Top of page

Parameter

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

convert gpt

On Itanium-based computers, converts an empty basic disk with the master boot record (MBR) partition style into a basic disk with the GUID partition table (GPT) partition style.

Important

- The disk must be empty to convert it to a GPT disk. Back up your data and then delete all partitions or volumes before converting the disk.

⇡Top of page

Syntax

convert gpt [**noerr**]

⇡Top of page

Parameter

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

convert mbr

On Itanium-based computers, converts an empty basic disk with the GUID Partition Table (GPT) partition style to a basic disk with the master boot record (MBR) partition style.

Important

- The disk must be empty to convert it to an MBR disk. Back up your data and then delete all partitions or volumes before converting the disk.

⇡Top of page

Syntax

convert mbr [**noerr**]

⇡Top of page

Parameter

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create partition efi

On Itanium-based computers, creates an Extensible Firmware Interface (EFI) system partition on a GUID Partition Table (GPT) disk. After the partition has been created, the focus is given to the new partition.

Top of page

Syntax

create partition efi [**size=***n*] [**offset**=*n*] [**noerr**]

Top of page

Parameters

size=*n* : The size of the partition in megabytes (MB). If no size is given, then the partition continues until there is no more free space in the current region.

offset=*n* : The byte offset at which to create the partition. If no offset is given, the partition is placed in the first disk extent that is large enough to hold it.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create partition extended

Creates an extended partition on the current drive. After the partition has been created, the focus automatically shifts to the new partition. Only one extended partition can be created per disk. This command fails if you attempt to create an extended partition within another extended partition. You must create an extended partition before you can create logical drives.

Top of page

Syntax

create partition extended [**size=***n*] [**offset=***n*] [**noerr**]

Top of page

Parameters

size=*n* : The size of the extended partition in megabytes (MB). If no size is given, then the partition continues until there is no more free space in the region. The size is cylinder snapped. The size is rounded to the closest cylinder boundary. For example, if you specify a size of 500 MB, the partition would be rounded up to 504 MB.

offset=*n* : Applies to master boot record (MBR) disks only. The byte offset at which to create the extended partition If no offset is given, the partition will start at the beginning of the first free space on the disk. The offset is cylinder snapped. The offset is rounded to the closest cylinder boundary. For example, if you specify an offset that is 27 MB and the cylinder size is 8 MB, the offset is rounded to the 24 MB boundary.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create partition logical

Creates a logical drive in the extended partition. After the partition has been created, the focus automatically shifts to the new logical drive.

Top of page

Syntax

create partition logical [**size=**_n_] [**offset=**_n_] [**noerr**]

Top of page

Parameters

size=*n* : The size of the logical drive in megabytes (MB). If no size is given, then the partition continues until there is no more free space in the current region.

offset=*n* : Applies to master boot record (MBR) disks only. The byte offset at which to create the logical drive. The offset is cylinder snapped (that is, the offset is rounded up to completely fill whatever cylinder size is being used). If no offset is given, then the partition is placed in the first disk extent that is large enough to hold it. The partition is at least as long in bytes as the number specified by **size=***n*. If you specify a size for the logical drive, it must be smaller than the extended partition.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create partition msr

On Itanium-based computers, creates a Microsoft Reserved (MSR) partition on a GUID Partition Table (GPT) disk.

Caution

- Be very careful when using the **create partition msr** command. Because GPT disks require a specific partition layout, creating Microsoft reserved partitions could cause the disk to become unreadable. On GPT disks that are used to start Windows XP 64-Bit Edition, the EFI System partition is the first partition on the disk, followed by the Microsoft Reserved partition. GPT disks used only for data storage do not have an EFI System partition; the Microsoft Reserved partition is the first partition.

  Windows XP 64-Bit Edition does not mount Microsoft reserved partitions. You cannot store data on them and you cannot delete them.

Top of page

Syntax

create partition msr [**size=***n*] [**offset=***n*] [**noerr**]

Top of page

Parameters

size=*n* : The size of the partition in megabytes (MB). The partition is at least as long in bytes as the number specified by **size=***n*. If no size is given, the partition continues until there is no more free space in the current region.

offset=*n* : The byte offset at which to create the partition. The partition starts at the byte offset specified by **offset=***n*. It is sector snapped; that is, the offset is rounded up to completely fill whatever sector size is being used. If no offset is given, then the partition is placed in the first disk extent that is large enough to hold it.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create partition primary

Creates a primary partition on the current basic disk After you create the partition, the focus automatically shifts to the new partition. The partition does not receive a drive letter. You must use the **assign** command to assign a drive letter to the partition.

⇡Top of page

Syntax

create partition primary [**size=***n*] [**offset=***n*] [**ID=**{*byte*|*GUID*}] [**noerr**]

⇡Top of page

Parameters

size=*n* The size of the partition in megabytes (MB). If no size is given, the partition continues until there is no more unallocated space in the current region. The size is cylinder snapped. The size is rounded to the closest cylinder boundary. For example, if you specify a size of 500 MB, the partition would be rounded up to 504 MB. **offset=***n* The byte offset at which to create the partition. If no offset is given, the partition will start at the beginning of the first free space on the disk. For master boot record (MBR) disks, the offset is cylinder snapped. The offset is rounded to the closest cylinder boundary. For example, if you specify an offset that is 27 MB and the cylinder size is 8 MB, the offset is rounded to the 24 MB boundary. **ID=**{*byte*|*GUID*} Intended for Original Equipment Manufacturer (OEM) use only.

Caution

- Creating partitions with this parameter might cause your computer to crash or be unable to start up. Unless you are an OEM or an IT professional experienced with GPT disks, do not create partitions on GPT disks using the **ID=***byte* | *GUID* parameter. Instead, always use the **create partition efi** command to create EFI System partitions, the **create partition msr** command to create Microsoft Reserved partitions, and the **create partition primary** command (without the **ID=***byte* | *GUID* parameter) to create primary partitions on GPT disks.

For MBR disks, you can specify a partition type byte, in hexadecimal form, for the partition. If no partition type byte is specified on an MBR disk, the **create partition primary** command creates a partition of type 0x6. Any partition type byte can be specified with the **ID=***byte* | *GUID* parameter. **DiskPart** does not check the partition type byte for validity, nor does it perform any other checking of the ID parameter.

For GPT disks you can specify a partition type GUID for the partition you want to create:

- EFI System partition: c12a7328-f81f-11d2-ba4b-00a0c93ec93b
- Microsoft reserved partition: e3c9e316-0b5c-4db8-817d-f92df00215ae
- MSDATA partition: ebd0a0a2-b9e5-4433-87c0-68b6b72699c7
- LDM Metadata partition on a dynamic disk: 5808c8aa-7e8f-42e0-85d2-e1e90434cfb3
- LDM Data partition on a dynamic disk: af9b60a0-1431-4f62-bc68-3311714a69ad

If no partition type GUID is specified, the **create partition primary** command creates an MSDATA partition. Any partition type can be specified with the **ID=**{*byte* | *GUID*} parameter. **DiskPart** does not check the partition GUID for validity, nor does it perform any other checking of the ID parameter.

noerr For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create volume raid

Creates a RAID-5 volume on the specified dynamic disks After you create the volume, the focus automatically shifts to the new volume.

⇡Top of page

Syntax

create volume raid [**size=***n*] [**disk=***n*[,[*n*,]] [**noerr**]

⇡Top of page

Parameters

size=*n* : The amount of disk space, in megabytes (MB), that the volume will occupy on each disk. If no size is given, the largest possible RAID-5 volume will be created. The disk with the smallest available contiguous free space determines the size for the RAID-5 volume and the same amount of space is allocated from each disk. The actual amount of usable disk space in the RAID-5 volume is less than the combined amount of disk space because some of the disk space is required for parity.

disk=*n* : The dynamic disks on which to create the volume. An amount of space equal to **size=***n* is allocated on each disk.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create volume simple

Creates a simple volume After you create the volume, the focus automatically shifts to the new volume.

⇡Top of page

Syntax

create volume simple [**size=***n*] [**disk=***n*] [**noerr**]

⇡Top of page

Parameters

size=*n* : The size of the volume in megabytes (MB). If no size is given, the new volume takes up the remaining free space on the disk.

disk=*n* : The dynamic disk on which to create the volume. If no disk is given, the current disk is used.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

create volume stripe

Creates a striped volume on the specified disks. After you create the volume, the focus automatically shifts to the new volume.

⇡Top of page

Syntax

create volume stripe [**size=***n*] [**disk=***n*[,[*n*,]] [**noerr**]

⇡Top of page

Parameters

size=*n* : The amount of disk space, in megabytes (MB), that the volume will occupy on each disk. If no size is given, the new volume takes up the remaining free space on the smallest disk and an equal amount of space on each subsequent disk.

disk=*n* : The dynamic disks on which to create the volume. An amount of space equal to **size=***n* is allocated on each disk.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

delete disk

Deletes a missing dynamic disk from the disk list.

⇡Top of page

Syntax

delete disk [**noerr**] [**override**]

⇡Top of page

Parameters

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

override : Enables **DiskPart** to delete all simple volumes on the disk. If the disk contains half of a mirrored volume, the half of the mirror on the disk is deleted. The **delete disk override** command fails if the disk is a member of a RAID-5 volume.

delete partition

On a basic disk, deletes the partition with focus. You cannot delete the system partition, boot partition, or any partition that contains the active paging file or crash dump (memory dump).

Caution

- Deleting a partition on a dynamic disk can delete all offline dynamic volumes on the disk, thus destroying any data and converting the disk to a basic disk. To delete a dynamic volume, always use the **delete volume** command instead.

Partitions can be deleted from dynamic disks, but they should not be created. For example, it is possible to delete an unrecognized GUID Partition Table (GPT) partition on a dynamic GPT disk. Deleting such a partition does not cause the resulting free space to become available. This command is particularly intended to allow reclamation of the space on a corrupted offline dynamic disk in an emergency situation where the **clean** command cannot be used.

⇡Top of page

Syntax

delete partition [**noerr**] [**override**]

⇡Top of page

Parameters

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

override : Enables **DiskPart** to delete any partition regardless of type. Typically, **DiskPart** only allows you to delete known data partitions.

delete volume

Deletes the selected volume. You cannot delete the system volume, boot volume, or any volume that contains the active paging file or crash dump (memory dump).

⇡Top of page

Syntax

delete volume [**noerr**]

⇡Top of page

Parameter

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

detail disk

Displays the properties of the selected disk and the volumes on that disk.

⇡Top of page

Syntax

detail disk

detail volume

Displays the disks on which the current volume resides.

Top of page

Syntax

detail volume

exit

Exits the **DiskPart** command interpreter.

Top of page

Syntax

exit

extend

Extends the volume with focus into next contiguous unallocated space. For basic volumes, the unallocated space must be on the same disk as, and must follow (be of higher sector offset than) the partition with focus. A dynamic simple or spanned volume can be extended to any empty space on any dynamic disk Using this command, you can extend an existing volume into newly created space.

If the partition was previously formatted with the NTFS file system, the file system is automatically extended to occupy the larger partition. No data loss occurs. If the partition was previously formatted with any file system format other than NTFS, the command fails with no change to the partition.

You cannot extend the current system or boot partitions.

Top of page

Syntax

extend [**size=***n*] [**disk=***n*] [**noerr**]

Top of page

Parameters

size=*n* : The amount of space, in megabytes (MB), to add to the current partition. If you do not specify a size, the disk is extended to take up all of the next contiguous unallocated space.

disk=*n* : The dynamic disk on which to extend the volume. An amount of space equal to **size=***n* is allocated on the disk. If no disk is specified, the volume is extended on the current disk.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

help

Displays a list of the available commands.

Top of page

Syntax

help

import

Imports a foreign disk group into the local computer's disk group. The **import** command imports every disk that is in the same group as the disk that has focus.

Top of page

Syntax

import [**noerr**]

Top of page

Parameter

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes

DiskPart to exit with an error code.

list disk

Displays a list of disks and information about them, such as their size, amount of available free space, whether the disk is a basic or dynamic disk, and whether the disk uses the master boot record (MBR) or GUID partition table (GPT) partition style. The disk marked with an asterisk (*) has focus.

⇡Top of page

Syntax

list disk

list partition

Displays the partitions listed in the partition table of the current disk. On dynamic disks, these partitions may not correspond to the dynamic volumes on the disk. This discrepancy occurs because dynamic disks contain entries in the partition table for the system volume or boot volume (if present on the disk) and a partition that occupies the remainder of the disk in order to reserve the space for use by dynamic volumes.

⇡Top of page

Syntax

list partition

list volume

Displays a list of basic and dynamic volumes on all disks.

⇡Top of page

Syntax

list volume

online

Brings an offline disk or volume with focus online.

⇡Top of page

Syntax

online [**noerr**]

⇡Top of page

Parameter

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

rem

Provides a way to add comments to a script.

⇡Top of page

Syntax

rem

⇡Top of page

Examples

rem These commands set up 3 drives.

create partition primary size=2048

assign d:

create partition extend

create partition logical size=2048

assign e:

create partition logical

assign f:

remove

Removes a drive letter or mount point from the volume with focus. If the **all** parameter is used, all current drive letters and mount points are removed. If no drive letter or mount point is specified, then **DiskPart** removes the first drive letter or mount point it encounters.

The **remove** command can be used to change the drive letter associated with a removable drive. You cannot remove the drive letters on system, boot, or paging volumes. In addition, you cannot remove the drive letter for an OEM partition, any GPT partition with an unrecognized GUID, or any of the special, non-data, GPT partitions such as the EFI system partition.

⇡Top of page

Syntax

remove [{**letter=***d*|**mount=***Path* [**all**]}] [**noerr**]

⇡Top of page

Parameters

letter=*d* : The drive letter to be removed.

mount=*Path* : The mount point path to be removed.

all : Removes all current drive letters and mount points.

noerr : For scripting only. When an error is encountered, specifies that **DiskPart** continue to process commands as if the error did not occur. Without the **noerr** parameter, an error causes **DiskPart** to exit with an error code.

rescan

Locates new disks that may have been added to the computer.

⇡Top of page

Syntax

rescan

retain

Prepares an existing dynamic simple volume to be used as a boot or system volume.

On an x86-based computer, creates a partition entry in the master boot record (MBR) on the dynamic simple volume with focus. To create an MBR partition, the dynamic simple volume must start at a cylinder aligned offset and be an integral number of cylinders in size.

On an Itanium-based computer, creates a partition entry in the GUID partition table (GPT) on the dynamic simple volume with focus.

Note

- The **retain** command is intended for use only during Unattended Setup or by Original Equipment Manufacturers (OEMs).

⇡Top of page

Syntax

retain

select disk

Selects the specified disk and shifts the focus to it.

⇡Top of page

Syntax

select disk=[*n*]

⇡Top of page

Parameters

***n* :** The disk number of the disk to receive focus. If no disk number is specified, the **select** command lists the disk that currently has the focus. You can view the numbers for all disks on the computer by using the **list disk** command.

select partition

Selects the specified partition and gives it focus. If no partition is specified, the **select** command lists the current partition with focus. You can view the numbers of all partitions on the current disk by using the **list partition** command.

⇡Top of page

Syntax

select partition=[{*n*|*d*}]

⇡Top of page

Parameters

***n* :** The number of the partition to receive the focus.

***d* :** The drive letter or mount point path of the partition to receive the focus.

select volume

Selects the specified volume and shifts the focus to it. If no volume is specified, the **select** command lists the current volume with focus. You can specify the volume by number, drive letter, or mount point path. On a basic disk, selecting a volume also gives the corresponding partition focus. You can view the numbers of all volumes on the computer by using the **list volume** command.

⇡Top of page

Syntax

select volume=[{*n*|*d*}]

Top of page

Parameters

***n* :** The number of the volume to receive the focus.

***d* :** The drive letter or mount point path of the volume to receive the focus.

Top of page

DiskPart scripting

Using **DiskPart**, you can create scripts to automate disk-related tasks, such as creating volumes or converting disks to dynamic. Scripting these tasks is useful if you are deploying Windows by using Unattended Setup or Sysprep, which do not support creating volumes other than the boot volume

To start a **DiskPart** script, at the command prompt, type:

DiskPart /S *scriptname.txt*

Where *scriptname.txt* is the name of the text file that contains your script.

To redirect **DiskPart**'s scripting output to a file, type:

DiskPart /S *scriptname.txt* > *logfile.txt*

Where *logfile.txt* is the name of the text file where **DiskPart** writes its output.

When **DiskPart** starts, the **DiskPart** version and computer name are displayed at the command prompt. By default, if **DiskPart** encounters an error while attempting to perform a scripted task, **DiskPart** stops processing the script and displays an error code (unless you specified the **noerr** parameter). However, **DiskPart** always returns errors when it encounters syntax errors, regardless of whether you used the **noerr** parameter. The **noerr** parameter enables you to perform useful tasks such as using a single script to delete all partitions on all disks regardless of the total number of disks.

The following table lists the **DiskPart** error codes:

Error	Description
0	No errors occurred. The entire script ran without failure.
1	A fatal exception occurred. There may be a serious problem.
2	The parameters specified for a **DiskPart** command were incorrect.
3	**DiskPart** was unable to open the specified script or output file.
4	One of the services **DiskPart** uses returned a failure.
5	A command syntax error occurred. The script failed because an object was improperly selected or was invalid for use with that command.

Command-line reference A-Z

Disk Management overview

Top of page

Manage Your Profile |Legal |Contact Us

Doskey

Calls Doskey.exe, which recalls Windows XP commands, edits command lines, and creates macros.

Syntax

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

⇡Top of page

Parameters

/reinstall : Installs a new copy of Doskey.exe. 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 **/macros** to redirect the list to a file. You can abbreviate **/macros** to **/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 **/history** to redirect the list to a file. You can abbreviate **/history** as **/h**.

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

/exename=*exename* : Specifies the program (that is, executable) in which the **doskey** macro runs.

/macrofile=*FileName* : Specifies a file containing macros that you want 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.

/? : Displays help at the command prompt.

Top of page

Remarks

- Using Doskey.exe

  Doskey.exe is always available for all character-based, interactive programs (such as program debuggers or file transfer programs), and maintains a command history buffer and macros for each program that it starts. You cannot use **doskey** command-line options from a program. You must run **doskey** command-line options before you start 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 starting Doskey.exe. If you use Doskey.exe within a program, that program's key assignments take precedence.

Key	Description
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 Doskey.exe, you can edit the current command line. If you use Doskey.exe within a program, that program's key assignments take precedence and some Doskey.exe editing keys might not work.

The following table lists **doskey** editing keys and their functions.

Key or key combination	Description
LEFT ARROW	Moves the insertion point back one character.
RIGHT ARROW	Moves the insertion point forward one character.
CTRL+LEFT ARROW	Moves the insertion point back one word.
CTRL+RIGHT ARROW	Moves the insertion point forward one word.
HOME	Moves the insertion point to the beginning of the line.
END	Moves the insertion point 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 Command Prompt window. (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.exe 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.exe begins copying characters from the position in the template that corresponds to the position indicated by the insertion point on the command line.
F4	Deletes characters from the current insertion point position up to a character you specify. To use this editing key, press F4 and type a character. Doskey.exe deletes the characters from the current insertion point position to the first letter specified.
F5	Copies the template into the current command line.
F6	Places an end-of-file character (that is, CTRL+Z) at the current insertion point position.

F7	Displays all commands for this program stored in memory in a pop-up box. Use the UP ARROW key and the DOWN ARROW key to select the command you want, and press ENTER to run the command. 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 program debuggers or file transfer programs (FTP), automatically use Doskey.exe. To use Doskey.exe, a program must be a console process and use buffered input. Program key assignments override **doskey** key assignments. For example, if the program uses the F7 key for some function, you cannot get a **doskey** command history in a pop-up window.

With Doskey.exe you can maintain a command history for each program you start, repeat and edit previous commands at the program's prompt, and start **doskey** macros created for the program. If you exit and then restart a program from the same Command Prompt window, the command history from the previous program session is available.

You must run Doskey.exe before you start a program. You cannot use **doskey** command-line options from a program's command prompt, even if the program has a shell command.

If you want to customize how Doskey.exe works with a program and create **doskey** macros for that program, you can create a batch program that modifies Doskey.exe 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, after you press ENTER, Doskey.exe returns your keyboard to replace mode. You must press INSERT again to return to insert mode.

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

  The insertion point changes shape when you use the INSERT key to change from one mode to the other.
- Creating a macro

  You can use Doskey.exe to create macros that carry out one or more commands. The following table lists special characters you can use to control command operations when you define a macro.

Character	Description
$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 (>).
GG or gg	Appends output to the end of a file. Use either of these special double characters to append output to an existing file instead of replacing 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. These special characters are 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 create macros or type commands on the **doskey** command line. These special characters are 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: everything you type on the command line after the macro name is substituted for the $* in the macro.

- 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 white space to separate parameters. You cannot run a **doskey** macro from a batch program.
- Creating a macro with the same name as a Windows XP command

  If you always use a particular command with specific command-line options, you can create a macro that has the same name as the command. To specify whether you want to run the macro or the 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 run the command, insert one or more white spaces between the command prompt and the command name.
- Deleting a macro

  To delete a macro, type:

  doskey *macroname***=**

Top of page

Examples

The **/macros** and **/history** command-line options are useful for creating batch programs to save macros and commands. For example, to store all current **doskey** macros, type:

doskey /macros > macinit

To use the macros stored in Macinit, type:

```
doskey /macrofile=macinit
```

To create a batch program named Tmp.bat that contains recently used commands, type:

```
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 in wide display format. You can use **$*** at the end of the macro to append other command-line options to **dir** when you run TX.

The following macro uses a batch parameter for a new directory name:

```
doskey mc=md $1$tcd $1
```

The macro creates a new directory and then changes to it from the current directory.

To use the preceding macro to create and change to a directory named Books, type:

```
mc books
```

To create a **doskey** macro for a program called Ftp.exe, include **/exename** as follows:

```
doskey /exename:ftp.exe go=open 10.127.1.100$tmget *.TXT c:\reports$bye
```

To use the preceding macro, start **ftp**. At the **ftp** prompt, type:

```
go
```

Ftp runs the **open**, **mget**, and **bye** commands.

To create a macro that performs a quick and unconditional format of a disk, type:

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

To format a disk in drive A quickly and unconditionally, type:

qf a:

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Format

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Driverquery

Displays a list of all installed device drivers and their properties.

Syntax

driverquery [**/s** *Computer*] [**/u** *Domain\User* **/p** *Password*] [**/fo** {**TABLE**|**LIST**|**CSV**}] [**/nh**] [**/v**] [**/si**]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fo {TABLE|LIST|CSV} : Specifies the format to display the driver information. Valid values are **TABLE**, **LIST**, and **CSV**. The default format for output is **TABLE**.

/nh : Omits the header row from the displayed driver information. Valid when the **/fo** parameter is set to **TABLE** or **CSV**.

/v : Specifies that detailed driver information be displayed.

/si : Displays digital signature information for both signed and unsigned device drivers.

/?: Displays help at the command prompt.

⇡Top of page

Examples

The following examples show how you can use the **driverquery** command:

driverquery
driverquery /fo csv
driverquery /nh
driverquery /s ipaddress
driverquery /s server1 /u hirpln /v driverquery /s server1 /u maindom\hirpln /p p@ssw3d /fo list

[Top of page](#)

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

[Top of page](#)

Manage Your Profile |Legal |Contact Us

Echo

Turns the command-echoing feature on or off, or displays a message. Used without parameters, **echo** displays the current echo setting.

Syntax

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

Top of page

Parameters

{on|off} : Specifies whether to turn the command-echoing feature on or off.

***message* :** Specifies text you want to display on the screen.

/? : Displays help at the command prompt.

Top of page

Remarks

- The **echo** *message* command is useful when echo is turned 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.
- If you use **echo off**, the command prompt does not appear on your screen. To display the command prompt, type **echo on**.
- To prevent echoing of a line, insert an at sign (@) in front of a command in a batch program.
- To echo a blank line on the screen, type:

  echo.
- To display a pipe (|) or redirection character (< or >) when you are using **echo**, use a caret character immediately before the pipe or redirection character (for example, ^>, ^<, or ^|). If you need to use the caret character (^), type two (^^).

Top of page

Examples

The following example is a batch program that includes a three-line message preceded by and then 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, type an at sign (@) before the command as follows:

@echo off

You can use the **if** and **echo** commands on the same command line. For example:

if exist *.rpt echo The report has arrived.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Endlocal

Ends localization of environment changes in a batch file, restoring environment variables to their values before the matching **setlocal** command.

Syntax

endlocal

⇡Top of page

Parameters

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- You must use **endlocal** in a script or batch file. If you use **endlocal** outside of a script or batch file, it has no effect.
- There is an implicit **endlocal** command at the end of a batch file.
- With command extensions enabled (that is, the default), the **endlocal** command restores the state of command extensions (that is, enabled or disabled) to what it was before the matching **setlocal** command was executed. For more information about enabling and disabling command extensions, see **cmd** in Related Topics.

⇡Top of page

Examples

You can localize environment variables in a batch file. For example:

```
@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
```

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Command-line reference A-Z

⇡Top of page

Eventcreate

Enables an administrator to create a custom event in a specified event log.

Syntax

eventcreate [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]] {[**/l** {**APPLICATION**|**SYSTEM**}]|[**/so** *SrcName*]} **/t** {**ERROR**|**WARNING**|**INFORMATION**|**SUCCESSAUDIT**|**FAILUREAUDIT**} **/id** *EventID* **/d** *Description*

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/l {APPLICATION|SYSTEM} : Specifies the name of the event log where the event will be created. The valid log names are APPLICATION and SYSTEM.

/so *SrcName* : Specifies the source to use for the event. A valid source can be any string and should represent the application or component that is generating the event.

/t {ERROR|WARNING|INFORMATION|SUCCESSAUDIT|FAILUREAUDIT} : Specifies the type of event to create. The valid types are ERROR, WARNING, INFORMATION, SUCCESSAUDIT, and FAILUREAUDIT.

/id *EventID* : Specifies the event ID for the event. A valid ID is any number from 1 to 65535.

/d *Description* : Specifies the description to use for the newly created event.

/?: Displays help at the command prompt.

Top of page

Remarks

- Custom events cannot be written to the security log.

Top of page

Examples

The following examples show how you can use the **eventcreate** command:

```
eventcreate /t error /id 100 /l application /d "Create event in application log"
eventcreate /t information /id 1000 /so winmgmt /d "Create event in WinMgmt source"
eventcreate /t error /id 2001 /so winword /l application /d "new src Winword in application log"
eventcreate /s server /t error /id 100 /l application /d "Remote machine without user credentials"
eventcreate /s server /u user /p password /id 100 /t error /l application /d "Remote machine with user credentials"
eventcreate /s server1 /s server2 /u user /p password /id 100 /t error /so winmgmt /d "Creating events on Multiple remote machines"
eventcreate /s server /u user /id 100 /t warning /so winmgmt /d "Remote machine with partial user credentials"
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command-line reference

Top of page

Manage Your Profile |Legal |Contact Us

Eventquery.vbs

Lists the events and event properties from one or more event logs.

Syntax

eventquery[**.vbs**] [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] [**/fi** *FilterName*] [**/fo** {**TABLE**|**LIST**|**CSV**}] [**/r** *EventRange* [**/nh**] [**/v**] [**/l** [**APPLICATION**] [**SYSTEM**] [**SECURITY**] [**"DNS server"**] [*UserDefinedLog*] [*DirectoryLogName*] [*****]]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the script with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fi *FilterName* : Specifies the types of events to include in or exclude from the query. To find events with either value, **Type** and **ID** can be used together in a single syntax statement by using the or operator.The following are valid filter names, operators, and values.

Name	Operator	Value
Datetime	**eq, ne, ge, le, gt, lt**	mm/dd/yy(yyyy), hh:mm:ssAM(/PM)
Type	**eq, ne, or**	{**ERROR**|**INFORMATION**|**WARNING**|**SUCCESSAUDIT**|**FAILUREAUDIT**}
ID	**eq, ne, or, ge, le, gt, lt**	Any valid positive integer.
User	**eq, ne**	Any valid string.

Computer	**eq, ne**	Any valid string.
Source	**eq, ne**	Any valid string.
Category	**eq, ne**	Any valid string

/fo {TABLE|LIST|CSV} : Specifies the format to use for the output. Valid values are **table**, **list**, and **csv**.

/r *EventRange* : Specifies the range of events to list.

Value	Description
N	Lists *N* most recent events.
-*N*	Lists *N* oldest events.
N1-*N2*	Lists the events from *N1* to *N2*.

/nh : Suppresses column headers in the output. Valid only for **table** and **csv** formats.

/v : Specifies that verbose event information be displayed in the output.

/l [APPLICATION] [SYSTEM] [SECURITY] ["DNS server"] [*UserDefinedLog*] [*DirectoryLogName*] [*]] : Specifies the log(s) to monitor. Valid values are **Application**, **System**, **Security**, **"DNS server"**, a user-defined log, and Directory log. **"DNS server"** can be used only if the DNS service is running on the computer specified by the **/s** parameter. To specify more than one log to monitor, reuse the **/l** parameter. The wildcard (*) can be used and is the default.

/?: Displays help at the command prompt.

Top of page

Remarks

- To run this script, you must be running CScript. If you have not already set the default Windows Script Host to CScript, type:

```
cscript //h:cscript //s //nologo
```

Top of page

Examples

The following examples show how you can use the **eventquery** command:

```
eventquery /l system
eventquery /l mylog
eventquery /l application /l system
eventquery /s srvmain /u maindom\hiropln /p p@ssW23 /v /l *
eventquery /r 10 /l application /nh
eventquery /r -10 /fo LIST /l security
eventquery /r 5-10 /l "DNS server"
eventquery /fi "Type eq Error" /l application
eventquery /fi "Datetime eq 06/25/00,03:15:00AM/06/25/00,03:15:00PM" /l application
eventquery /fi "Datetime gt 08/03/00,06:20:00PM" /fi "id gt 700" /fi "Type eq warning" /l
system eventquery /fi "ID eq 1000 OR ID ge 4500" eventquery /fi "Type eq error OR Type
eq INFORMATION" eventquery /fi "ID eq 250 OR Type eq ERROR"
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Using the command-based script host (CScript.exe)

Eventtriggers

Displays and configures event triggers on local or remote machines.

To view the command syntax, click a command:

eventtriggers create

Creates a new event trigger that monitors and acts upon the occurrence of log events of given criteria.

Syntax

eventtriggers[**.exe**] **/create** [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] **/tr** *TriggerName* [**/l** [**APPLICATION**] [**SYSTEM**] [**SECURITY**] ["**DNS Server**"] [**LOG**] [*DirectoryLogName*] [*****]] {[**/eid** *ID*]|[**/t** {**ERROR**|**INFORMATION**|**WARNING**|**SUCCESSAUDIT**|**FAILUREAUDIT**}]|[**/so** *Source*]} [**/d** *Description*] **/tk** *TaskName* [**/ru** {[*Domain*]*User* | **"System"**} [**/rp** *Password*]]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/tr *TriggerName* : Specifies a friendly name to associate with the event trigger.

/l [APPLICATION] [SYSTEM] [SECURITY] ["DNS Server"] [LOG] [*DirectoryLogName*] [*]] : Specifies the event log(s) to monitor. Valid types include: Application, System, Security, DNS server, Log, and Directory log. The wildcard (*) can be used and is the default value.

/eid *ID* : Specifies a specific event ID for which the event trigger should monitor. Valid values are any valid integer. Cannot be used in conjunction with the **/type** or **/so** parameters.

/t {ERROR|INFORMATION|WARNING|SUCCESSAUDIT|FAILUREAUDIT} : Specifies an event type for which the event trigger should monitor. Valid values include: ERROR, INFORMATION, WARNING, SUCCESSAUDIT, and FAILUREAUDIT. Cannot be used in conjunction with the **/id** or **/so** parameters.

/so *Source* : Specifies an event source for which the event trigger should monitor. Valid values are any string. Cannot be used in conjunction with the **/id** or **/type** parameters.

/d *Description* : Specifies a detailed description of the event trigger. Valid values are any string.

/tk *TaskName* : Specifies the task/command/line to execute when the event trigger conditions are met.

/ru {[*Domain*\]*User* | "System"} : Runs the tasks with the permission of the specified user account. By default, the task runs with the permissions of the user logged on to the computer running SchTasks.

Value	Description
[*Domain*\]*User*	Specifies a user account.
"System" or **""**	Specifies the NT Authority\System account, which is used by the operating system.

/rp *Password* : Specifies the password of the user account that is specified in the **/ru** parameter. If you omit this parameter when specifying a user account, SchTasks.exe prompts you for the password and obscures the text you type. Tasks run with with permissions of the NT Authority\System account do not require a password and SchTasks.exe does not prompt for one.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **eventtriggers /create** command:

eventtriggers /create /tr "Disk Cleanup" /l system /t error /tk c:\windows\system32\cleanmgr.exe
eventtriggers /create /s srvmain /u maindom\hiropln /p p@ssW23 /tr "Low Disk Space" /eid 4133 /t warning /tk \\server\share\diskcleanup.cmd
eventtriggers /create /s srvmain /user maindom\hiropln /p p@ssW23 /tr "Disk Backup" /eid 4133 /l system /t error /tk \\server\share\ntbackup.exe

eventtriggers delete

Deletes an event trigger from a system by event trigger ID.

Top of page

Syntax

eventtriggers[**.exe**] **/delete** [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] **/tid** {*ID*|*****}

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/tid {*ID*|*} : Specifies the event trigger(s) to be deleted by "Event Trigger ID". The (*) wildcard can be used.

/?: Displays help at the command prompt.

⇧Top of page

Examples

The following examples show how you can use the **eventtriggers /delete** command:

eventtriggers /delete /tid 1 /tid 2 /tid 4 /tid 6
eventtriggers /delete /s srvmain /u maindom\hiropln /p p@ssW23 /tid *
eventtriggers /delete /s srvmain /u maindom\hiropln /p p@ssW23 /tid 1

eventtriggers query

Queries and displays a system's event trigger properties and settings.

⇧Top of page

Syntax

eventtriggers[**.exe**] **/query** [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] [**/fo** {**TABLE**|**LIST**|**CSV**}] [**/nh**] [**/v**]

⇧Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fo {TABLE|LIST|CSV} : Specifies the format to use for the query output. Valid values are **TABLE**, **LIST**, and **CSV**. The default format for output is **TABLE**.

/nh : Suppresses column header in the output. Valid when the **/fo** parameter is set to **TABLE** or **CSV**.

/v : Specifies that detailed information be displayed in the output.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **eventtriggers /query** command:

eventtriggers /query
eventtriggers /query /s srvmain
eventtriggers /query /s srvmain /u maindom\hiropln /p p@ssW23 /fo list

Top of page

Remarks

- When specified without an operation, **eventtriggers** returns a list of event triggers. To see a list of event triggers, type:

  eventtriggers

  Output similar to the following appears:

```
Trigger ID Event Trigger Name         Task
========== ========================== ==============================================
         1 Disk Cleanup               c:\windows\system32\cleanmgr.exe
```

- In the case that an event fails to execute, **eventtriggers** creates a log file called TriggerConsumer.log in the \windows\system32\wbem\logs directory containing a message that the event failed to trigger.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command-line reference

Evntcmd

Configures the translation of events to traps, trap destinations, or both based on information in a configuration file.

Syntax

evntcmd [**/s** *ComputerName*] [**/v** *VerbosityLevel*] [**/n**] *FileName*

⇡Top of page

Parameters

/s *ComputerName* : Specifies, by name, the computer on which you want to configure the translation of events to traps, trap destinations, or both. If you do not specify a computer, the configuration occurs on the local computer.

/v *VerbosityLevel* : Specifies which types of status messages appear as traps and trap destinations are configured. This parameter must be an integer between 0 and 10. If you specify **10**, all types of messages appear, including tracing messages and warnings about whether trap configuration was successful. If you specify **0**, no messages appear.

/n : Specifies that the SNMP service should not be restarted if this computer receives trap configuration changes.

***FileName* :** Specifies, by name, the configuration file that contains information about the translation of events to traps and trap destinations you want to configure.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- If you want to configure traps but not trap destinations, you can create a valid configuration file by using Event to Trap Translator, which is a graphical utility. If you have the SNMP service installed, you can start Event to Trap Translator by typing **evntwin** at a command prompt. After you have defined the traps you want, click **Export** to create a file suitable for use with **evntcmd**. You can use Event to Trap Translator to easily create a configuration file and then use the configuration file with **evntcmd** at the command prompt to quickly configure traps on multiple computers.
- The syntax for configuring a trap is as follows:

  #pragma ADD *EventLogFile EventSource EventID* [*Count* [*Period*]]

  - The text **#pragma** must appear at the beginning of every entry in the file.
  - The parameter **ADD** specifies that you want to add an event to trap configuration.
  - The parameters *EventLogFile*, *EventSource*, and *EventID* are required. The parameter *EventLogFile* specifies the file in which the event is recorded. The parameter *EventSource* specifies the application that generates the event. The *EventID* parameter specifies the unique number that identifies each event. To find out what values correspond to particular events, start Event to Trap Translator by typing **evntwin** at a command prompt. Click **Custom**, and then click **Edit**. Under **Event Sources**, browse the folders until you locate the event you want to configure, click it, and then click **Add**. Information about the event source, the event log file, and the event ID appear under **Source**, **Log**, and **Trap specific ID**, respectively.
  - The *Count* parameter is optional, and it specifies how many times the event must occur before a trap message is sent. If you do not use the *Count* parameter, the trap message is sent after the event occurs once.
  - The *Period* parameter is optional, but it requires you to use the *Count* parameter. The *Period* parameter specifies a length of time (in seconds) during which the event must occur the number of times specified with the *Count* parameter before a trap message is sent. If you do not use the *Period* parameter, a trap message is sent after the event occurs the number of times specified with the *Count* parameter, no matter how much time elapses between occurrences.
- The syntax for removing a trap is as follows:

  #pragma DELETE *EventLogFile EventSource EventID*

  - The text **#pragma** must appear at the beginning of every entry in the file.
  - The parameter **DELETE** specifies that you want to remove an event to trap configuration.
  - The parameters *EventLogFile*, *EventSource*, and *EventID* are required. The parameter *EventLogFile* specifies the log in which the event is recorded. The parameter *EventSource* specifies the application that generates the event. The *EventID* parameter specifies the unique number that identifies each event.

- The syntax for configuring a trap destination is as follows:

  #pragma ADD_TRAP_DEST *CommunityName HostID*

  - The text **#pragma** must appear at the beginning of every entry in the file.
  - The parameter **ADD_TRAP_DEST** specifies that you want trap messages to be sent to a specified host within a community.
  - The parameter **CommunityName** specifies, by name, the community in which trap messages are sent.
  - The parameter *HostID* specifies, by name or IP address, the host to which you want trap messages to be sent.
- The syntax for removing a trap destination is as follows:

  #pragma DELETE_TRAP_DEST *CommunityName HostID*

  - The text **#pragma** must appear at the beginning of every entry in the file.
  - The parameter **DELETE_TRAP_DEST** specifies that you do not want trap messages to be sent to a specified host within a community.
  - The parameter *CommunityName* specifies, by name, the community in which trap messages are sent.
  - The parameter *HostID* specifies, by name or IP address, the host to which you do not want trap messages to be sent.

Top of page

Examples

The following examples illustrate entries in the configuration file for the **evntcmd** command. They are not designed to be typed at a command prompt.

To send a trap message if the Event Log service is restarted, type:

#pragma ADD System "Eventlog" 2147489653

To send a trap message if the Event Log service is restarted twice in three minutes, type:

#pragma ADD System "Eventlog" 2147489653 2 180

To stop sending a trap message whenever the Event Log service is restarted, type:

#pragma DELETE System "Eventlog" 2147489653

To send trap messages within the community named Public to the host with the IP address 192.168.100.100, type:

#pragma ADD_TRAP_DEST public 192.168.100.100

To send trap messages within the community named Private to the host named Host1, type:

#pragma ADD_TRAP_DEST private Host1

To stop sending trap messages within the community named Private to the same computer on which you are configuring trap destinations, type:

#pragma DELETE_TRAP_DEST private localhost

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Exit

Exits the current batch script or the Cmd.exe program (that is, the command interpreter) and returns to the program that started Cmd.exe or to the Program Manager.

Syntax

exit [**/b**] [*ExitCode*]

Top of page

Parameters

/b : Exits the current batch script.

***ExitCode* :** Specifies a numeric number.

/? : Displays help at the command prompt.

Top of page

Remarks

- If you use **/b** outside of a batch script, it will exit Cmd.exe.
- If you use **/b**, Cmd.exe sets the ERRORLEVEL to the specified *ExitCode*. If you exit Cmd.exe, Cmd.exe sets the process exit code with the specified *ExitCode*.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Expand

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

Syntax

expand [**-r**] *Source* [*Destination*]

expand -d *source.cab* [**-f:***files*]

expand *source.cab* **-f:***files Destination*

⇧Top of page

Parameters

-r : Renames expanded files.

-d : Displays a list of files in the source location. Does not expand or extract the files.

-f:*files* : Specifies the files in a cabinet (.cab) file you want to expand. You can use wildcards (* and ?).

***Source* :** Specifies the files to expand. *Source* can consist of a drive letter and colon, a directory name, a file name, or a combination. You can use wildcards (* and ?).

***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 file name, or a combination.

/?: Displays help at the command prompt.

⇧Top of page

Remarks

•Using **expand** at the Recovery Console

The **expand** command, with different parameters, is available from the Recovery Console.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Fc

Compares two files and displays the differences between them.

Syntax

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

Top of page

Parameters

/a : Abbreviates the output of an ASCII comparison. Instead of displaying all of 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 the following file extensions: .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, except files with the following file extensions: .exe, .com, .sys, .obj, .lib, or .bin.

/lb*n* : Sets the *n* number of lines for the internal line buffer. The default length of the line buffer is 100 lines. If the files that you are comparing have more than this number of consecutive differing lines, **fc** cancels the comparison.

/n : Displays the line numbers during an ASCII comparison.

/t : Prevents **fc** from converting 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 (that is, tabs and spaces) during the comparison. If a line contains many consecutive spaces or tabs, **/w** treats these characters as a single space. When used with the **/w** command-line option, **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 *nnnn*, **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. *Filename1* is required.

[*drive2*:][*path2*]*filename2* : Specifies the location and name of the second file you want to compare. *Filename2* is required.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•Reporting differences between files for an ASCII comparison

When you use **fc** for an ASCII comparison, **fc** displays differences between two files in the following order:

•Name of the first file
•Lines from *filename1* that differ between the files
•First line to match in both files
•Name of the second file
•Lines from *filename2* that differ
•First line to match

- Using **/b** for binary comparisons

  /b displays mismatches found during a binary comparison as follows:

  xxxxxxxx: yy zz

  The value of *xxxxxxxx* 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 (that is, * and ?) in *filename1* and *filename2*. 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*.
- Working with 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.

Top of page

Examples

To make an ASCII comparison of two text files that are named Monthly.rpt and Sales.rpt and display the results in abbreviated format, type:

fc /a monthly.rpt sales.rpt

To make a binary comparison of two batch files named Profits.bat and Earnings.bat, type:

fc /b profits.bat earnings.bat

The results similar to the following appear:

```
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 are identical, **fc** displays the following message:

FC: no differences encountered

To compare every .bat file in the current directory with the file New.bat, type:

fc *.bat new.bat

To compare the file New.bat on drive C with the file New.bat on drive D, type:

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:

fc c:*.bat d:*.bat

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Using filters

Used in conjunction with the command redirection pipe character (|), a command filter is a command within a command that reads the command's input, transforms the input, and then writes the output. Filter commands help you sort, view, and select parts of a command output.

Filter commands divide, rearrange, or extract portions of the information that passes through them. The following table lists filter commands that are available in Windows XP.

Command	Description
more	Displays the contents of a file or the output of a command in one Command Prompt window at a time.
find	Searches through files and command output for the characters you specify.
sort	Alphabetizes files and command output.

To send input from a file to a filter command, use a less than sign (<). If you want the filter command to get input from another command, use a pipe (|).

Using the more command

The **more** command displays the contents of a file or the output of a command in one Command Prompt window at a time. For example, to display the contents of a file called List.txt in one Command Prompt window at a time, type:

more < list.txt

One Command Prompt window of information appears, and then the -- More -- prompt appears at the bottom of the Command Prompt window. To continue to the next Command Prompt window, press any key on the keyboard except PAUSE. To stop the command without viewing more information, press CTRL+C.

You can use the **more** command when you work with a command that produces more than one Command Prompt window of output. For example, suppose you want to view a directory tree on your hard disk. If you have more directories than can be displayed in the Command Prompt window, you can use the **tree** command with a pipe (|) and the **more** command as follows:

```
tree c:\ | more
```

The first Command Prompt window of output from the **tree** command appears, followed by the -- More -- prompt. Output pauses until you press any key on the keyboard, except PAUSE.

⇡Top of page

Using the find command

The **find** command searches files for the string or text that you specify. Cmd.exe displays every line that matches the string or text that you specify in the Command Prompt window. You can use the **find** command either as a filter command or a standard Windows XP command. For more information about using **find** as a standard command, see Find

To use **find** as a filter command, you must include a less than sign (<) and the string or text on which you want to search. By default, **find** searches are case-sensitive. For example, the following command finds occurrences of the string "Pacific Rim" in the file Trade.txt:

```
find "Pacific Rim" < trade.txt
```

The output does not include any occurrences of "pacific rim." It includes occurrences of the capitalized "Pacific Rim" only.

To save the output of the **find** command rather than display it in the Command Prompt window, type a greater than sign (>) and the name of the file where you want to store the output. For example, the following command finds occurrences of "Pacific Rim" in the Trade.txt file and saves them in Nwtrade.txt:

```
find "Pacific Rim" < trade.txt > nwtrade.txt
```

⇡Top of page

Using the sort command

The **sort** command alphabetizes a text file or the output of a command. For example, the following command sorts the contents of a file named List.txt and displays the results in the Command Prompt window:

```
sort < list.txt
```

In this example, the **sort** command sorts the lines of the List.txt file into an alphabetical list and displays the results without changing the file. To save the output of the **sort** command rather than display it, type a greater than sign (>) and a file name. For example, the following command alphabetizes the lines of the List.txt file and stores the results in the Alphlist.txt file:

```
sort < list.txt > alphlist.txt
```

To sort the output of a command, type the command, type a pipe (|), and then type **sort** (that is, *command* **| sort**). For example, the following command sorts the lines that include the string "Jones" (that is, the **find** command output) in alphabetical order:

```
find "Jones" maillst.txt | sort
```

Top of page

More

Displays one screen of output at a time.

Syntax

command **| more** [**/c**] [**/p**] [**/s**] [**/t***n*] [**+***n*]

more [[**/c**] [**/p**] [**/s**] [**/t***n*] [**+***n*]] < [*Drive***:**] [*Path*] *FileName*

more [**/c**] [**/p**] [**/s**] [**/t***n*] [**+***n*] [*files*]

⇡Top of page

Parameters

[*Drive*:] [*Path*] *FileName* : Specifies the file to display.

***command* :** Specifies a command for which you want to display the output.

/c : Clears screen before displaying page.

/p : Expands form-feed characters.

/s : Changes multiple blank lines to one blank line.

/t*n* : Changes tabs to the number of spaces specified by *n*.

+*n* : Displays first file beginning at the line specified by *n*.

***files* :** Specifies list of files to display. Separate file names with a space.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Using **more** at the Recovery Console

  The **more** command, with different parameters, is available from the Recovery Console.
- Using **more** subcommands

  The following commands are accepted at the **more** prompt (-- More --).

Key	Action
SPACEBAR	Display next page
ENTER	Display next line
f	Display next file
q	Quit
?	Show available commands
=	Show line number
p *n*	Display next *n* lines
s *n*	Skip next *n* lines

- Sources of data

  When using the redirection character (<), you must specify a file name as the source. When using the pipe (|), you can use such commands as **dir**, **sort**, and **type**.

Top of page

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 the following prompt appears:

-- More --

You can then press the SPACEBAR 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

The **more** command displays the first screen of information from Clients.new, and then the following prompt appears:

-- More --

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, type **f**.

To quit **more**, type **q**.

Add commands to the **more** prompt as shown in this example:

-- More [Options: psfq=<space><ret>] --

To display the current line number, type **=**. The current line number is added to the **more** prompt as shown in this example:

-- More [Line: 24] --

To display a specific number of lines, type **p**. **More** prompts you for the number of lines to display as follows:

-- 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, type **s**. **More** prompts you for the number of lines to skip as follows:

-- More -- Lines:

Type the number of lines to skip, and then press ENTER. **More** skips the specified number of lines and displays the next screen of information.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Dir

Sort

Type

Using filters

Using command redirection operators

Command-line reference A-Z

Find

Searches for a specific string of text in a file or files. After searching the specified file or files, **find** displays any lines of text that contain the specified string.

Syntax

find [**/v**] [**/c**] [**/n**] [**/i**] **"***string***"** [[*Drive***:**][*Path*]*FileName*[...]]

⇡Top of page

Parameters

/v : Displays all lines that do not contain the specified *string*.

/c : Counts the lines that contain the specified *string* and displays the total.

/n : Precedes each line with the file's line number.

/i : Specifies that the search is not case-sensitive.

"*string*" : Required. Specifies the group of characters that you want to search for. You must enclose *string* in quotation marks (that is, "*string*").

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the file in which to search for the specified string.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Specifying a string

  If you do not use **/i**, **find** searches for exactly what you specify for *string*. For example, the **find** command treats the characters "a" and "A" differently. If you use **/i**, however, **find** is not case-sensitive and treats "a" and "A" as 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 (that is, **"***StringContaining***""***QuotationMarks***"**).
- Using **find** as a filter

  If you omit a file name, **find** acts as a filter, taking input from the standard input source (usually the keyboard, a pipe, or a redirected file) and then displaying any lines that contain *string*.
- Ordering command syntax

  You can type parameters and command-line options for the **find** command in any order.
- Using wildcards

  You cannot use wildcards (that is, * and ?) in file names or extensions that you specify with the **find** command. To search for a string in a set of files that you specify with wildcards, you can use the **find** command in a **for** command.
- Using **/v** or **/n** with **/c**

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

⇡Top of page

Examples

To display all lines from Pencil.ad that contain the string "Pencil Sharpener", type:

find "Pencil Sharpener" pencil.ad

To find a string that contains text within quotation marks, you must first enclose the entire string in quotation marks. Second, you must use two quotation marks for each quotation mark contained within the string. To find "The scientists labeled their paper "for discussion only." It is not a final report." in Report.doc, type:

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. To search the current directory for files that have the extension .bat and that contain the string "PROMPT," type:

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

To search your hard disk to find and display the file names on drive C that contain the string "CPU," use the pipe (|) to direct the results of a **dir** command to **find** as follows:

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

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

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Using filters

Using command redirection operators

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Sort

Reads input, sorts data, and writes the results to the screen, to a file, or to another device

Syntax

sort [**/r**] [/+*n*] [**/m** *kilobytes*] [**/l** *locale*] [**/rec** *characters*] [[*drive1:*][*path1*]*filename1*] [**/t** [*drive2:*][*path2*]] [**/o** [*drive3:*][*path3*]*filename3*]

[*command* |] **sort** [**/r**] [/+*n*] [**/m** *kilobytes*] [**/l** *locale*] [**/rec** *characters*] [[*drive1:*][*path1*]*filename1*] [**/t** [*drive2:*][*path2*]] [**/o** [*drive3:*][*path3*]*filename3*]

⇡Top of page

Parameters

/r : Reverses the sort order (that is, sorts from Z to A, and then from 9 to 0).

/+*n* : Specifies the character position number, *n*, at which **sort** begins each comparison.

/m *kilobytes* : Specifies the amount of main memory to use for the sort, in kilobytes (KB).

/l *locale* : Overrides the sort order of characters defined by the system default locale (that is, the language and Country/Region selected during installation).

/rec *characters* : Specifies the maximum number of characters in a record, or a line of the input file (the default is 4,096, and the maximum is 65,535).

[*drive1:*][*path1*]*filename1* : Specifies the file to be sorted. If no file name is specified, the standard input is sorted. Specifying the input file is faster than redirecting the same file as standard input.

/t [*drive2:*][*path2*] : Specifies the path of the directory to hold the **sort** command's working storage, in case the data does not fit in main memory. The default is to use the system temporary directory.

/o [*drive3:*][*path3*]*filename3* : Specifies the file where the sorted input is to be stored. If not specified, the data is written to the standard output. Specifying the output file is faster than redirecting standard output to the same file.

/? : Displays help at the command prompt.

Top of page

Remarks

•Using the **/+***n* command-line option

In using the **/+***n* command-line option, for example, /+3 indicates that each comparison should begin at the third character in each line. Lines with fewer than *n* characters collate before other lines. By default, comparisons start at the first character in each line.

•Using the **/m** *kilobytes* command-line option

The memory used is always a minimum of 160 KB. If the memory size is specified, the exact specified amount (but at least 160 KB) is used for the sort, regardless of how much main memory is available.

The default maximum memory size when no size is specified is 90 percent of available main memory if both the input and output are files, and 45 percent of main memory otherwise. The default setting usually gives the best performance.

•Using the **/l** *locale* command-line option

Currently, the only alternative to the default locale is the "C" locale, which is faster than natural language sorting and sorts characters according to their binary encodings.

•Specifying a source

Unless you specify the *command* or *FileName* parameter, **sort** acts as a filter and takes input from the standard input (that is, usually from the keyboard, from a pipe, or from a file).

•Using redirection symbols with sort

You can use the pipe (|) symbol to direct data through the **sort** command from another command, or to direct the sort output to another command (for example, to the **more** command to display information one screen at a time). Using the less than symbol (<) or greater than symbol (>) to specify the input file or output file may not be very efficient; instead, specify the input file directly (as defined in the command syntax) and specify the output file using the **/o** parameter. This can be much faster, particularly with large files.

- Uppercase vs. lowercase

  The **sort** command does not distinguish between uppercase and lowercase letters.
- Limits on file size

  The **sort** command has no limit on file size.
- Collating sequence

  The sort program uses the collating-sequence table that corresponds to the Country/Region 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.
- Memory usage

  If the sort fits in memory (that is, either the default maximum memory size or as specified by the **/m** parameter), the sort is performed in one pass. Otherwise, the sort is performed in two passes, such that the amounts of memory used for both the sort and merge passes are equal. When two passes are performed, the partially sorted data is stored in a temporary file on disk. If there is not enough memory to perform the sort in two passes, a run-time error is issued. If the **/m** command-line option is used to specify more memory than is truly available, performance degradation or a run-time error can occur.

Top of page

Examples

Sorting a file

The following command reads the file Expenses.txt, sorts it in reverse order, and displays it on your screen:

```
sort /r expenses.txt
```

Sorting the output from a command

To search a large file named Maillist.txt for the text "Jones," and to sort the results of the search, use the pipe (|) to direct the output of a **find** command to the **sort** command, as follows:

```
find "Jones" maillist.txt | sort
```

The command produces a sorted list of lines that contain the specified text.

Sorting keyboard input

To sort keyboard input and display the results alphabetically on the screen, you can first use the **sort** command with no parameters, as follows:

```
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 can also redirect sorted keyboard input to a file. For more information, click **redirect** in the Related Topics list.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Country

Using command redirection operators

Find

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Findstr

Searches for patterns of text in files using regular expressions.

Syntax

findstr [**/b**] [**/e**] [**/l**] [**/r**] [**/s**] [**/i**] [**/x**] [**/v**] [**/n**] [**/m**] [**/o**] [**/p**] [**/offline**] [**/g:***file*] [**/f:***file*] [**/c:***string*] [**/d:** *dirlist*] [**/a:***ColorAttribute*] [*strings*] [[*Drive***:**][*Path*] *FileName* [...]]

Top of page

Parameters

/b : Matches the pattern if at the beginning of a line.

/e : Matches the pattern if at the end of a line.

/l : Uses search strings literally.

/r : Uses search strings as regular expressions. **Findstr** interprets all metacharacters as regular expressions unless you use **/l**.

/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 file name if a file contains a match.

/o : Prints seek offset before each matching line.

/p : Skips files with non-printable characters.

/offline : Processes files with offline attribute set.

/f:*file* : Reads file list from the specified file.

/c:*string* : Uses specified text as a literal search string.

/g:*file* : Gets search strings from the specified file.

/d:*dirlist* : Searches a comma-delimited list of directories.

/a:*ColorAttribute* : Specifies color attributes with two hexadecimal digits.

***strings* :** Specified text to be searched for in *FileName*.

[*Drive*:][*Path*] *FileName* [...] : Specifies a file or files to search.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Using regular expressions with **findstr**

Findstr is capable of finding the exact text you are looking for in any ASCII file or files. However, sometimes 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.

The following table lists the metacharacters that **findstr** accepts.

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

The special characters in regular expression syntax are most powerful when you use them together. For example, the following combination of the wildcard character (.) and repeat (*) character match any string of characters:

.*

Use the following expression as part of a larger expression that matches any string beginning with "b" and ending with "ing":

b.*ing

Top of page

Examples

Use spaces to separate multiple search strings unless the argument is prefixed with **/c**. To search for "hello" or "there" in file x.y, type:

findstr "hello there" x.y

To search for "hello there" in file x.y, type:

findstr /c:"hello there" x.y

To find all occurrences of the word "Windows" (with an initial capital W) in the file Proposal.txt, type the following:

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 the following:

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 the following:

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 criterion 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, search the files listed in Filelist.txt, and then store the results in the file Results.out, type the following:

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 the following:

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 the following:

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

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Finger

Displays information about a user or users on a specified remote computer (typically a computer running UNIX) that is running the Finger service or daemon. The remote computer specifies the format and output of the user information display. Used without parameters, **finger** displays help.

Syntax

finger [**-l**] [*User*] [**@***host*] [**...**]

⇡Top of page

Parameters

-l : Displays user information in long list format.

***User* :** Specifies the user about which you want information. If you omit the *User* parameter, **finger** displays information about all users on the specified computer.

***@host* :** Specifies the remote computer running the Finger service where you are looking for user information. You can specify a computer name or IP address.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Multiple *User@host* parameters can be specified.
- You must prefix **finger** parameters with a hyphen (**-**) rather than a slash (**/**).
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections
- Windows 2000 and Windows XP do not provide a finger service.

⇡Top of page

Examples

To display information for user1 on the computer users.microsoft.com, type the following command:

finger user1@users.microsoft.com

To display information for all users on the computer users.microsoft.com, type the following command:

finger @users.microsoft.com

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Flattemp

Enables or disables flat temporary folders.

Syntax

flattemp {**/query** | **/enable** | **/disable**}

⇡Top of page

Parameters

/query : Queries the current setting.

/enable : Enables flat temporary directories.

/disable : Disables flat temporary directories.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Once each user has a unique temporary directory, use **flattemp /enable** to enable flat temporary directories.
- The default method for creating temporary folders for multiple users (usually pointed to by the TEMP and TMP environment variables) is to create subfolders in the \Temp folder, using the *logonID* as the subfolder name. For example, if the TEMP environment variable points to C:\Temp, the temporary folder assigned to the user *logonID* 4 is C:\Temp\4. Using **flattemp**, you can point directly to the \Temp folder and prevent subfolders from forming. This is useful when you want the user temporary folders to be contained in home directories, whether on a Terminal server local drive or on a network share drive. You should use this command only when each user has a separate temporary folder. For instructions on using temporary folders per session, see To use separate temporary folders for each session
- You must have administrative privileges to run **flattemp**.

- You might encounter application errors if the user's temporary directory is on a network drive. This occurs when the network share becomes momentarily inaccessible on the network. Because the temporary files of the application are either inaccessible or out of synchronization, it responds as if the disk has stopped. Moving the temporary directory to a network drive is not recommended. The default is to keep temporary directories on the local hard disk. If you experience unexpected behavior or disk-corruption errors with certain applications, stabilize your network or move the temporary directories back to the local hard disk.
- **Flattemp** settings are ignored if you disable using separate temporary folders per-session. This option is set in Terminal Services Configuration.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Terminal Services commands

Top of page

For

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

Syntax

for {**%***variable*|**%%***variable*} **in (***set***) do** *command* [*CommandLineOptions*]

⇡Top of page

Parameters

{%*variable*|%%*variable*} : Required. Represents a replaceable parameter. Use **%***variable* to carry out **for** from the command prompt. Use **%%***variable* to carry out the **for** command within a batch file. Variables are case-sensitive and must be represented with an alpha value, such as %A, %B, or %C.

(*set*) : Required. Specifies one or more files, directories, range of values, or text strings that you want to process with the specified command. The parentheses are required.

command* :** Required. Specifies the command that you want to carry out on each file, directory, range of values, or text string included in the specified **(set***)**.

***CommandLineOptions* :** Specifies any command-line options that you want to use with the specified command.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

•Using **for**

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

- Using batch parameters

  The following attributes apply to the **for** command:

  - The **for** command replaces **%***variable* or **%%***variable* with each text string in the specified *set* until the *command* processes all of the files.
  - **For** *variable* names are case-sensitive, global, and no more than 52 total can be active at any one time.
  - 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 files, a single character such as **%%f** works.
  - You can use multiple values for *variable* in complex batch files 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 (that is, * 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 then the specified command processes this value. This continues until all of the files (or groups of files) that correspond to the *set* value are processed.

- Using the **in** and **do** keywords

  In and **do** are not parameters, but you must use them with **for**. If you omit either of these keywords, an error message appears.

- Using additional forms of **for**

If command extensions are enabled (that is, the default), the following additional forms of **for** are supported:

- Directories only

If *set* contains wildcards (* and ?), the specified *command* executes for each directory (instead of a set of files in a specified directory) that matches *set*. The syntax is:

for /D {**%%** | **%**}*variable* **in (***set***) do** *command* [*CommandLineOptions*]

- Recursive

Walks the directory tree rooted at [*Drive***:**]*Path*, executing the **for** statement in each directory of the tree. If no directory is specified after **/R**, the current directory is assumed. If *set* is just a single period (.), it only enumerates the directory tree. The syntax is:

for /R [[*Drive* **:**]*Path*] {**%%** | **%**}*variable* **in (***set***) do** *command* [*CommandLineOptions*]

- Iterating a range of values

Use an iterative variable to set the starting value (*start#*) and then step through a set range of values until the value exceeds the set ending value (*end#*). **/L** will execute the iterative by comparing *start#* with *end#*. If *start#* is less than *end#* the command will execute. When the iterative variable exceeds *end#* the command shell exists the loop. You can also use a negative *step#* to step through a range in decreasing values. For example, (1,1,5) generates the sequence 1 2 3 4 5 and (5,-1,1) generates the sequence (5 4 3 2 1). The syntax is:

for /L {**%%** | **%**}*variable* **in (***start#,step#,end#***) do** *command* [*CommandLineOptions*]

- Iterating and file parsing

Use file parsing to process command output, strings and file content. Use iterative variables to define the content or strings you want to examine and use the various *ParsingKeywords* options to further modify the parsing. Use the *ParsingKeywords* token option to specify which tokens should be passed as iterator variables. Note that when used without the token option, **/F** will only examine the first token.

File parsing consists of reading the output, string or file content, breaking it up into individual lines of text and then parsing each line into zero or more tokens. The **for** loop is then called with the iterator variable value set to the token. By default, **/F** passes the first blank separated token from each line of each file. Blank lines are skipped. The different syntaxes are:

for /F ["*ParsingKeywords*"] {**%%** | **%**}*variable* **in (***filenameset***) do** *command*

[*CommandLineOptions*]

for /F [**"***ParsingKeywords***"**] {**%%** | **%**}*variable* **in ("***LiteralString***") do** *command* [*CommandLineOptions*]

for /F [**"***ParsingKeywords***"**] {**%%** | **%**}*variable* **in ('***command***') do** *command* [*CommandLineOptions*]

The *filenameset* argument specifies one or more file names. Each file is opened, read and processed before going on to the next file in *filenameset*. To override the default parsing behavior, specify **"***ParsingKeywords***"**. This is a quoted string that contains one or more keywords to specify different parsing options.

If you use the usebackq option, use one of the following syntaxes:

for /F [**"usebackq***ParsingKeywords***"**] {**%%** | **%**}*variable* **in (**"*filenameset*"**) do** *command* [*CommandLineOptions*]

for /F [**"usebackq***ParsingKeywords***"**] {**%%** | **%**}*variable* **in ('***LiteralString***') do** *command* [*CommandLineOptions*]

for /F [**"usebackq***ParsingKeywords***"**] {**%%** | **%**}*variable* **in (`***command***`) do** *command* [*CommandLineOptions*]

The following table lists the parsing keywords that you can use for *ParsingKeywords*.

Keyword	Description
eol=c	Specifies an end of line character (just one character).
skip=n	Specifies the number of lines to skip at the beginning of the file.
delims=xxx	Specifies a delimiter set. This replaces the default delimiter set of space and tab.
tokens=x,y,m-n	Specifies which tokens from each line are to be passed to the **for** body for each iteration. As a result, additional variable names are allocated. The *m-n* form is a range, specifying the *m*th through the *n*th tokens. If the last character in the **tokens=** string is an asterisk (*), an additional variable is allocated and receives the remaining text on the line after the last token that is parsed.

usebackq	Specifies that you can use quotation marks to quote file names in *filenameset*, a back quoted string is executed as a command, and a single quoted string is a literal string command.

- Variable substitution

Substitution modifiers for **for** variable references have been enhanced. The following table lists optional syntax (for any variable **I**).

Variable with modifier	Description
%~I	Expands %I which removes any surrounding quotation marks ("").
%~fI	Expands %I to a fully qualified path name.
%~dI	Expands %I to a drive letter only.
%~pI	Expands %I to a path only.
%~nI	Expands %I to a file name only.
%~xI	Expands %I to a file extension only.
%~sI	Expands path to contain short names only.
%~aI	Expands %I to the file attributes of file.
%~tI	Expands %I to the date and time of file.
%~zI	Expands %I to the size of file.
%~$PATH:I	Searches the directories listed in the PATH environment variable and expands %I to the fully qualified name of the first one found. If the environment variable name is not defined or the file is not found by the search, this modifier expands to the empty string.

The following table lists modifier combinations that you can use to get compound results.

Variable with combined modifiers	Description
%~dpI	Expands %I to a drive letter and path only.
%~nxI	Expands %I to a file name and extension only.
%~fsI	Expands %I to a full path name with short names only.

%~dp$PATH:I	Searches the directories listed in the PATH environment variable for %I and expands to the drive letter and path of the first one found.
%~ftzaI	Expands %I to an output line that is like **dir**.

In the above examples, you can replace *%I* and PATH by other valid values. A valid **for** variable name terminates the %~ syntax.

By use uppercase variable names such as *%I*, you can make your code more readable and avoid confusion with the modifiers, which are not case-sensitive.

- Parsing a string

  You can use the **for /F** parsing logic on an immediate string, by wrapping the *filenameset* between the parentheses in single quotation marks (that is, '*filenameset*'). *Filenameset* is treated as a single line of input from a file, and then it is parsed.

- Parsing output

  You can use the **for /F** command to parse the output of a command by making the *filenameset* between the parenthesis a back quoted string. It is treated as a command line, which is passed to a child Cmd.exe and the output is captured into memory and parsed as if it were a file.

Top of page

Examples

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

for %%*variable* **in (***set***) do** *command* [*CommandLineOptions*]

To display the contents of all the files in the current directory that have the extension .doc or .txt using the replaceable variable *%f*, type:

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

In the preceding 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, replace every occurrence of *%f* with *%%f*. Otherwise, the variable is ignored and an error message is displayed.

To parse a file, ignoring commented lines, type:

for /F "eol=; tokens=2,3* delims=," %i in (myfile.txt) do @echo %i %j %k

This command parses each line in Myfile.txt, ignoring lines that begin with a semicolon and passing the second and third token from each line to the **FOR** body (tokens are delimited by commas or spaces). The body of the **FOR** statement references *%i* to get the second token, *%j* to get the third token, and *%k* to get all of the remaining tokens. If the file names that you supply contain spaces, use quotation marks around the text (for example, "*File Name*"). To use quotation marks, you must use **usebackq**. Otherwise, the quotation marks are interpreted as defining a literal string to parse.

%i is explicitly declared in the **FOR** statement, and *%j* and *%k* are implicitly declared by using **tokens=**. You can specify up to 26 tokens using **tokens=**, provided that it does not cause an attempt to declare a variable higher than the letter 'z' or 'Z'.

To parse the output of a command by placing *filenameset* between the parentheses, type:

for /F "usebackq delims==" %i IN (`set`) DO @echo %i

This example enumerates the environment variable names in the current environment.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Format

Formats the disk in the specified volume to accept Windows files.

Syntax

format *volume* [**/fs:***file-system*] [**/v:***label*] [**/q**] [**/a:***UnitSize*] [**/c**] [**/x**]

format *volume* [**/v:***label*] [**/q**] [**/f:***size*]

format *volume* [**/v:***label*] [**/q**] [**/t:***tracks* **/n:***sectors*]

format *volume* [**/v:***label*] [**/q**]

format *volume* [**/q**]

⇡Top of page

Parameters

***volume* :** Specifies the mount point, volume name, or drive letter of the drive you want to format. If you do not specify any of the following command-line options, **format** uses the volume type to determine the default format for the disk.

/fs:*file-system* : Specifies the file system to use FAT, FAT32, or NTFS Floppy disks can use only the FAT file system.

/v:*label* : Specifies the volume label. If you omit the **/v** command-line option or use it without specifying a volume label, **format** 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. For more information about disk volume labels, click **Dir**, **Label**, and **Vol** in the Related Topics list.

/a:*UnitSize* : Specifies the allocation unit size to use on FAT, FAT32, or NTFS volumes. If you do not specify *UnitSize*, it is chosen based on volume size. The following table lists valid values for *UnitSize*.

Value	Description
512	Creates 512 bytes per cluster.
1024	Creates 1024 bytes per cluster.
2048	Creates 2048 bytes per cluster.
4096	Creates 4096 bytes per cluster.
8192	Creates 8192 bytes per cluster.
16K	Creates 16 kilobytes per cluster.
32K	Creates 32 kilobytes per cluster.
64K	Creates 64 kilobytes per cluster.

/q : Performs a quick format. Deletes the file table and the root directory of a previously formatted volume but does not perform a sector by sector scan for bad areas. You should use the **/q** command-line option to format only previously formatted volumes that you know are in good condition.

/f:*size* : Specifies the size of the floppy disk to format. When possible, use this command-line option instead of the **/t** and **/n** command-line options. Windows accepts the following value for size:

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

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

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

/c : NTFS only. Files created on the new volume will be compressed by default.

/x : Causes the volume to dismount, if necessary, before it is formatted. Any open handles to the volume will no longer be valid.

/? : Displays help at the command prompt.

⇧Top of page

Remarks

- Using **format** at the Recovery Console

  The **format** command, with different parameters, is available from the Recovery Console.
- Administrative privileges

  You must be a member of the Administrators group to format a hard drive.
- 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. 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, click **label** in the Related Topics list.
- Formatting a hard disk

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

  ```
  WARNING, ALL DATA ON nonremovable 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.

- Unit size

  FAT file systems restrict the number of clusters to no more than 65526. FAT32 file systems restrict the number of clusters to between 65527 and 4177917.

  NTFS compression is not supported for allocation unit sizes above 4096.
- **Format** messages

  When formatting is complete, **format** 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** command-line option. Use this command-line option 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 table lists each exit code and a brief description of its meaning.

Exit code	Description
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.

Top of page

Examples

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

format a:

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

format a: /q

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

format a: /v:DATA

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command shell overview

Dir

If

Label

Vol

Subst

Fsutil

Fsutil is a command-line utility that you can use to perform many FAT and NTFS file system related tasks, such as managing reparse points, managing sparse files, dismounting a volume, or extending a volume. Because **fsutil** is quite powerful, it should only be used by advanced users who have a thorough knowledge of Windows XP. In addition, you must be logged on as an administrator or a member of the Administrators group in order to use **fsutil**.

The following table lists the **fsutil** subcommands.

Command	Description
behavior	Queries, changes, enables, or disables the settings for generating 8.3 character-length file names, accepting extended characters in 8.3 character-length file names on NTFS volumes, updating the last access timestamp on NTFS volumes, how often quota events are written to the system log, and the amount of disk space reserved of the MFT Zone.
dirty	Queries whether volume's dirty bit is set. Sets a volume's dirty bit. When a volume's dirty bit is set, **autochk** automatically checks the volume for errors the next time the computer is restarted.
file	Typically used by support professionals. Finds a file by its security identifier, queries allocated ranges for a file, sets a file's short name, sets a file's valid data length, or sets zero data for a file.
fsinfo	Typically used by support professionals. Lists all drives, queries the drive type, queries volume information, queries NTFS-specific volume information, or queries file system statistics.
hardlink	Creates a hard link. A hard link is a directory entry for a file. Every file can be considered to have at least one hard link. On NTFS volumes, each file can have multiple hard links, and thus a single file can appear in many directories (or even in the same directory with different names). Because all of the links reference the same file, programs can open any of the links and modify the file. A file is deleted from the file system only after all links to it have been deleted. After you create a hard link, programs can use it like any other file name.

objectid	Typically used by support professionals. Manages object identifiers, which are used by Windows XP to track objects such as files and directories.
quota	Manages disk quotas on NTFS volumes in order to provide more precise control of network-based storage. Disk quotas are implemented on a per-volume basis and enable both hard- and soft-storage limits to be implemented on a per-user basis.
reparsepoint	Typically used by support professionals. Queries or deletes reparse points, which are NTFS file system objects that have a definable attribute containing user-controlled data, and are used to extend functionality in the input/output (I/O) subsystem. Reparse points are used for directory junction points and volume mount points. They are also used by file system filter drivers to mark certain files as special to that driver.
sparse	Manages sparse files A sparse file is a file with one or more regions of unallocated data in it. A program will see these unallocated regions as containing bytes with the value zero, but there is actually no disk space used to represent these zeros. In other words, all meaningful or nonzero data is allocated, whereas all non-meaningful data (large strings of data composed of zeros) is not allocated. When a sparse file is read, allocated data is returned as stored and unallocated data is returned, by default, as zeros, in accordance with the C2 security requirement specification. Sparse file support allows data to be deallocated from anywhere in the file.
usn	Typically used by support professionals. Manages the update sequence number (USN) change journal, which provides a persistent log of all changes made to files on the volume.
volume	Manages a volume. Dismounts a volume or queries to see how much free space is available on a disk.

Note

- To view help for the available subcommands at the command prompt, type **fsutil**, type the subcommand, and then type **help** (that is, **fsutil** *subcommand* **help**).

Fsutil: behavior

Fsutil: dirty

Fsutil: file

Fsutil: fsinfo

Fsutil: hardlink

Fsutil: objectid

Fsutil: quota

Fsutil: reparsepoint

Fsutil: sparse

Fsutil: usn

Fsutil: volume

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp

Transfers files to and from a computer running a File Transfer Protocol (FTP) server service such as Internet Information Services. **Ftp** can be used interactively or in batch mode by processing ASCII text files.

Syntax

ftp [**-v**] [**-d**] [**-i**] [**-n**] [**-g**] [**-s:***FileName*] [**-a**] [**-w:***WindowSize*] [**-A**] [*Host*]

⇡Top of page

Parameters

-v : Suppresses the display of FTP server responses.

-d : Enables debugging, displaying all commands passed between the FTP client and FTP server.

-i : Disables interactive prompting during multiple file transfers.

-n : Suppresses the ability to log on automatically when the initial connection is made.

-g : Disables file name globbing. **Glob** permits the use of the asterisk (*) and question mark (?) as wildcard characters in local file and path names. For more information, see Ftp: Glob

-s:*FileName* : Specifies a text file that contains **ftp** commands. These commands run automatically after **ftp** starts. This parameter allows no spaces. Use this parameter instead of redirection (<).

-a : Specifies that any local interface can be used when binding the FTP data connection.

-w:*WindowSize* : Specifies the size of the transfer buffer. The default window size is 4096 bytes.

-A : Logs onto the FTP server as anonymous.

***Host* :** Specifies the computer name, IP address, or IPv6 address of the FTP server to which to connect. The host name or address, if specified, must be the last parameter on the line.

/? : Displays help at the command prompt.

⇧Top of page

Remarks

- You must prefix **ftp** parameters with a hyphen (**-**) rather than a slash (**/**).
- **Ftp** command-line parameters are case-sensitive.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections
- **Ftp** can be used interactively. After it is started, **ftp** creates a subenvironment in which you can use **ftp** commands. You can return to the command prompt by typing the **quit** command. When the **ftp** subenvironment is running, it is indicated by the **ftp >** command prompt.
- For more information about **ftp** subcommands, see Related Topics.
- Ftp supports the use of IPv6 when the IPv6 protocol is installed. For more information, see IP version 6 and IPv6 applications

⇧Top of page

Examples

To log on to the FTP server named ftp.example.microsoft.com, type the following command:

```
ftp ftp.example.microsoft.com
```

To anonymously log on to the FTP server named ftp.example.microsoft.com, type the following command:

```
ftp -A ftp.example.microsoft.com
```

To log on to the FTP server named ftp.example.microsoft.com and run the **ftp** commands contained in a file named Resynch.txt, type the following command:

```
ftp -s:resynch.txt ftp.example.microsoft.com
```

⇧Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp subcommands

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp subcommands

For more information, click a command:

- Ftp: !
- Ftp: ?
- Ftp: append
- Ftp: ascii
- Ftp: bell
- Ftp: binary
- Ftp: Bye
- Ftp: cd
- Ftp: close
- Ftp: debug
- Ftp: delete
- Ftp: dir
- Ftp: disconnect
- Ftp: get
- Ftp: glob
- Ftp: hash
- Ftp: help
- Ftp: lcd
- Ftp: literal
- Ftp: ls
- Ftp: mdelete
- Ftp: mdir
- Ftp: mget
- Ftp: mkdir
- Ftp: mls
- Ftp: mput
- Ftp: open
- Ftp: prompt
- Ftp: put
- Ftp: pwd

- Ftp: quit
- Ftp: quote
- Ftp: recv
- Ftp: remotehelp
- Ftp: rename
- Ftp: rmdir
- Ftp: send
- Ftp: status
- Ftp: trace
- Ftp: type
- Ftp: user
- Ftp: verbose

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: !

Exits the ftp subsystem to a command prompt temporarily so that Windows XP commands can be run. To return to the ftp subsystem, type **exit** at the command prompt.

Syntax

!

↑Top of page

Parameters

? ! : Displays help for the **!** command.

↑Top of page

Remarks

- Use **!** to execute Windows XP commands without quitting ftp.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: help

Displays descriptions for **ftp** subcommands.

Syntax

help [*Command*]

? [*Command*]

Top of page

Parameters

***Command* :** Specifies the name of the command for which you want a description. If *Command* is not specified, a list of all commands is displayed.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp: append

Appends a local file to a file on the remote computer using the current file type setting.

Syntax

append *LocalFile* [*RemoteFile*]

⇡Top of page

Parameters

***LocalFile* :** Specifies the local file to add.

***RemoteFile* :** Specifies the file on the remote computer to which *LocalFile* is added.

? append : Displays help for the **append** command.

⇡Top of page

Remarks

- If *RemoteFile* is omitted, the *LocalFile* name is used in place of the remote file name.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: ascii

Sets the file transfer type to ASCII.

Syntax

ascii

⇡Top of page

Parameters

? ascii : Displays help for the **ascii** command.

⇡Top of page

Remarks

- The file transfer type is ASCII by default.
- 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** supports both ASCII and binary image file transfer types. You should use ASCII when transferring text files. For more information about binary file transfer, see **Ftp: binary** in Related Topics.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: binary

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: bell

Toggles an audible sound to occur after each file transfer command is completed.

Syntax

bell

Top of page

Parameters

? bell : Displays help for the **bell** command.

Top of page

Remarks

- By default, **bell** is off.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: binary

Sets the file transfer type to binary.

Syntax

binary

↑Top of page

Parameters

? binary : Displays help for the **binary** command.

↑Top of page

Remarks

- **Ftp** supports both ASCII and binary image file transfer types. You should use binary when transferring executable files. In binary mode, files are transferred in one-byte units. For more information about ASCII file transfer, see **Ftp: Ascii** in Related Topics.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: ascii

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: bye

Ends the FTP session with the remote computer and exits **ftp**.

Syntax

bye

⇡Top of page

Parameters

? bye : Displays help for the **bye** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Ftp: cd

Changes the working directory on the remote computer.

Syntax

cd *RemoteDirectory*

↑Top of page

Parameters

RemoteDirectory **:** Specifies the directory on the remote computer to which you want to change.

? cd : Displays help for the **cd** command.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: close

Ends the FTP session with the remote server and remains at the **ftp>** prompt.

Syntax

close

Top of page

Parameters

? close : Displays help for the **close** command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Ftp: debug

Toggles debugging mode.

Syntax

debug

Top of page

Parameters

? debug : Displays help for the **debug** command.

Top of page

Remarks

- By default, debugging is off.
- When debugging is on, each command that is sent to the remote computer is displayed, preceded by the greater than (>) character.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: delete

Deletes files on remote computers.

Syntax

delete *RemoteFile*

Top of page

Parameters

***RemoteFile* :** Specifies the file to delete.

? delete : Displays help for the **delete** command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: dir

Displays a list of directory files and subdirectories on a remote computer.

Syntax

dir [*RemoteDirectory*] [*LocalFile*]

⇡Top of page

Parameters

***RemoteDirectory* :** 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.

***LocalFile* :** Specifies a local file in which to store the listing. If a local file is not specified, results are displayed on the screen.

? dir : Displays help for the **dir** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: disconnect

Disconnects from the remote computer and retains the **ftp>** prompt.

Syntax

disconnect

[Top of page](#)

Parameters

? disconnect : Displays help for the **disconnect** command.

[Top of page](#)

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

[Ftp](#)

[Command-line reference A-Z](#)

Ftp: get

Copies a remote file to the local computer using the current file transfer type.

Syntax

get *RemoteFile* [*LocalFile*]

⇡Top of page

Parameters

RemoteFile **:** Specifies the remote file to copy.

LocalFile **:** Specifies the name of the file to use on the local computer.

? get : Displays help for the **get** command.

⇡Top of page

Remarks

- If *LocalFile* is not specified, the file is given the *RemoteFile* name.
- The **get** command is identical to the **recv** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Ftp: glob

Toggles wildcard expansion of local file names.

Syntax

glob

↑Top of page

Parameters

? glob : Displays help for the **glob** command.

↑Top of page

Remarks

- By default, globbing is on.
- Globbing permits the use of the asterisk (*) and question mark (?) as wildcard characters in local file or path names.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

↑Top of page

Manage Your Profile |Legal |Contact Us

Ftp: hash

Toggles number sign (#) printing for each data block that is transferred.

Syntax

hash

Top of page

Parameters

? hash : Displays help for the **hash** command.

Top of page

Remarks

- The size of a data block is 2048 bytes.
- By default, **hash** is off.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: lcd

Changes the working directory on the local computer. By default, the working directory is the directory in which **ftp** was started.

Syntax

lcd [*Directory*]

Top of page

Parameters

***Directory* :** Specifies the directory on the local computer to which to change. If *Directory* is not specified, the current working directory on the local computer is displayed.

? lcd : Displays help for the **lcd** command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: literal

Sends verbatim arguments to the remote FTP server. A single FTP reply code is returned.

Syntax

literal *Argument* [**...**]

⇡Top of page

Parameters

Argument **:** Specifies the argument to send to the FTP server.

? literal : Displays help for the **literal** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: ls

Displays an abbreviated list of files and subdirectories in a remote directory.

Syntax

ls [*RemoteDirectory*] [*LocalFile*]

⇡Top of page

Parameters

***RemoteDirectory* :** 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.

***LocalFile* :** Specifies a local file in which to store the listing. If a local file is not specified, results are displayed on the screen.

? ls : Displays help for the **ls** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: mdelete

Deletes files on remote computers.

Syntax

mdelete *RemoteFiles* [**...**]

⇡Top of page

Parameters

RemoteFiles **:** Specifies the remote files to delete.

? mdelete : Displays help for the **mdelete** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: mdir

Displays a list of files and subdirectories in a remote directory.

Syntax

mdir *RemoteFiles* [**...**] *LocalFile*

⇡Top of page

Parameters

***RemoteFiles* :** Specifies the directory for which you want to see a listing.

***LocalFile* :** Specifies a local file in which to store the listing.

? mdir : Displays help for the **mdir** command.

⇡Top of page

Remarks

•You can use **mdir** to specify multiple files.
•Specifying *RemoteFiles*

Type a hyphen (**-**) to use the current working directory on the remote computer.
•Specifying a *LocalFile*

Type a hyphen (**-**) to display the listing on the screen.

⇡Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: mget

Copies remote files to the local computer using the current file transfer type.

Syntax

mget *RemoteFiles* [**...**]

↑Top of page

Parameters

***RemoteFiles* :** Specifies the remote files to copy to the local computer.

? mget : Displays help for the **mget** command.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: mkdir

Creates a directory on the remote computer.

Syntax

mkdir *Directory*

↑Top of page

Parameters

***Directory* :** Specifies the name of the new remote directory.

? mkdir : Displays help for the **mkdir** command.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: mls

Displays an abbreviated list of files and subdirectories in a remote directory.

Syntax

mls *RemoteFiles* [...] *LocalFile*

⇡Top of page

Parameters

RemoteFiles **:** Specifies the files for which you want to see a listing.

LocalFile **:** Specifies a local file in which to store the listing.

? mls : Displays help for the **mls** command.

⇡Top of page

Remarks

- Specifying *RemoteFiles*

  Type a hyphen (-) to use the current working directory on the remote computer.
- Specifying *LocalFile*

  Type a hyphen (-) to display the listing on the screen.

⇡Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: mput

Copies local files to the remote computer using the current file transfer type.

Syntax

mput *LocalFiles* [...]

Top of page

Parameters

***LocalFiles* :** Specifies the local files to copy to the remote computer.

? mput : Displays help for the **mput** command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

tml; charset=iso-8859-1 Content-Length: 8827

Ftp: open

Connects to the specified FTP server.

Syntax

open *Computer* [*Port*]

⇡Top of page

Parameters

***Computer* :** Required. Specifies the remote computer to which you are trying to connect.

***Port* :** Specifies a TCP port number to use to contact an FTP server. By default, TCP port 21 is used.

? open : Displays help for the **open** command.

⇡Top of page

Remarks

- You can use an IP address or computer name (in which case a DNS server or Hosts file must be available) to specify *Computer*. If auto-login is on, **ftp** attempts to automatically log the user onto the FTP server. For information about disabling auto-login by using **ftp**, see Related Topics.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp subcommands

Command-line reference A-Z

Top of page

Ftp: prompt

Toggles between **prompt** mode on and off.

Syntax

prompt

Top of page

Parameters

? prompt : Displays help for the **prompt** command.

Top of page

Remarks

- By default, **prompt** is on.
- **Ftp** prompts during multiple file transfers to allow you to selectively retrieve or store files. **Mget** and **mput** transfer all files if **prompt** is off.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: put

Copies a local file to the remote computer using the current file transfer type.

Syntax

put *LocalFile* [*RemoteFile*]

Top of page

Parameters

***LocalFile* :** Specifies the local file to copy.

***RemoteFile* :** Specifies the name to use on the remote computer.

? put : Displays help for the **put** command.

Top of page

Remarks

- If *RemoteFile* is not specified, the file is given the *LocalFile* name.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: pwd

Prints the current directory on the remote computer.

Syntax

pwd

⇡Top of page

Parameters

? pwd : Displays help for the **pwd** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Ftp: quit

Ends the FTP session with the remote computer and exits **ftp** to the command prompt.

Syntax

quit

Top of page

Parameters

? quit : Displays help for the **quit** command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Ftp: quote

Sends verbatim arguments to the remote FTP server. A single FTP reply code is returned.

Syntax

quote *Argument* [**...**]

⇡Top of page

Parameters

***Argument* :** Specifies the argument to send to the FTP server.

? quote : Displays help for the **quote** command.

⇡Top of page

Remarks

- The **quote** command is identical to the **literal** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: recv

Copies a remote file to the local computer using the current file transfer type.

Syntax

recv *RemoteFile* [*LocalFile*]

⇡Top of page

Parameters

***RemoteFile* :** Specifies the remote file to copy.

***LocalFile* :** Specifies the name to use on the local computer.

? recv : Displays help for the **recv** command.

⇡Top of page

Remarks

- The **recv** command is identical to the **get** command.
- If *LocalFile* is not specified, the file is given the *RemoteFile* name.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: get

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: remotehelp

Displays help for remote commands.

Syntax

remotehelp [*Command*]

⇡Top of page

Parameters

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.

? remotehelp : Displays help for the **remotehelp** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp: rename

Renames remote files.

Syntax

rename *FileName NewFileName*

⇡Top of page

Parameters

***FileName* :** Specifies the file that you want to rename.

***NewFileName* :** Specifies the new file name.

? rename : Displays help for the **rename** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp: rmdir

Deletes a remote directory.

Syntax

rmdir *Directory*

⇡Top of page

Parameters

***Directory* :** Specifies the name of the remote directory to delete.

? rmdir : Displays help for the **rmdir** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: send

Copies a local file to the remote computer using the current file transfer type.

Syntax

send *LocalFile* [*RemoteFile*]

Top of page

Parameters

***LocalFile* :** Specifies the local file to copy.

***RemoteFile* :** Specifies the name to use on the remote computer.

? send : Displays help for the **send** command.

Top of page

Remarks

- The **send** command is identical to the **put** command.
- If *RemoteFile* is not specified, the file is given the *LocalFile* name.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: put

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: status

Displays the current status of FTP connections.

Syntax

status

Top of page

Parameters

? status : Displays help for the **status** command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: trace

Toggles packet tracing.

Syntax

trace

↑Top of page

Parameters

? trace : Displays help for the **trace** command.

↑Top of page

Remarks

- **Trace** displays the series of internal FTP function calls when running an **ftp** command.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: type

Sets or displays the file transfer type.

Syntax

type [*TypeName*]

⇡Top of page

Parameters

TypeName **:** Specifies the file transfer type.

? type : Displays help for the **type** command.

⇡Top of page

Remarks

- If *TypeName* is not specified, the current type is displayed.
- **Ftp** supports two file transfer types, ASCII and binary.

  The default file transfer type is ASCII. The **ascii** command 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 required, based on the operating system at the destination.

  The **binary** command should be used when transferring executable files. In binary mode, the file is moved in one-byte units.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Ftp: ascii

Ftp: binary

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: user

Specifies a user to the remote computer.

Syntax

user *UserName* [*Password*] [*Account*]

⇡Top of page

Parameters

***UserName* :** Specifies a user name with which to log on to the remote computer.

***Password* :** Specifies the password for *UserName*. If a password is not specified but is required, **ftp** prompts for the password.

***Account* :** Specifies an account with which to log on to the remote computer. If an *Account* is not specified but is required, **ftp** prompts for the account.

? user : Displays help for the **user** command.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftp: verbose

Toggles **verbose** mode.

Syntax

verbose

v

⇡Top of page

Parameters

? verbose : Displays help for the **verbose** command.

⇡Top of page

Remarks

- By default, **verbose** is on. When **verbose** is on, all **ftp** responses are displayed. When a file transfer is completed, statistics regarding the efficiency of the transfer are also displayed.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Ftp

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ftype

Displays or modifies file types used in file name extension associations. Used without parameters, **ftype** displays the file types that have open command strings defined.

Syntax

Ftype [*FileType*[**=**[*OpenCommandString*]]]

⇡Top of page

Parameters

***FileType* :** Specifies the file type you want to display or change.

***OpenCommandString* :** Specifies the open command to use when opening files of this type.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•Within an *OpenCommandString*, **ftype** substitutes the following variables:

- •*%0* or *%1* are replaced with the file name that you want to open.
- •*%** is replaced with all of the parameters.
- •*%~n* is replaced with all of the remaining parameters, starting with the *n*th parameter, where *n* can be any number from 2 to 9.
- •*%2* is replaced with the first parameter, *%3* with the second, and so on.

⇡Top of page

Examples

To display the current file types that have open command strings defined, type:

ftype

To display the current open command string for a specific file type, type:

ftype *FileType*

To delete the open command string for a specific file type, type:

ftype *FileType***=**

Type:

ASSOC .pl=PerlScript FTYPE PerlScript=perl.exe %1 %*

To invoke the Perl script, type:

script.pl 1 2 3

To eliminate the need to type the extensions, type:

set PATHEXT=.pl;%PATHEXT%

To invoke the Perl script, type:

script 1 2 3

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Assoc

Start

Command-line reference A-Z

Top of page

Getmac

Returns the media access control (MAC) address and list of network protocols associated with each address for all network cards in each computer, either locally or across a network.

Syntax

getmac[**.exe**] [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] [**/fo** {**TABLE**|**LIST**|**CSV**}] [**/nh**] [**/v**]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fo {TABLE|LIST|CSV} : Specifies the format to use for the query output. Valid values are **TABLE**, **LIST**, and **CSV**. The default format for output is **TABLE**.

/nh : Suppresses column header in output. Valid when the **/fo** parameter is set to **TABLE** or **CSV**.

/v : Specifies that the output display verbose information.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- **Getmac** can be useful either when you want to enter the MAC address into a network analyzer or when you need to know what protocols are currently in use on each network adapter in a computer.

Examples

The following examples show how you can use the **getmac** command:

getmac /fo table /nh /v
getmac /s srvmain
getmac /s srvmain /u maindom\hiropln
getmac /s srvmain /u maindom\hiropln /p p@ssW23
getmac /s srvmain /u maindom\hiropln /p p@ssW23 /fo list /v
getmac /s srvmain /u maindom\hiropln /p p@ssW23 /fo table /nh

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command-line reference

Top of page

Goto

Within a batch program, directs Windows XP to a line identified by a label. When the label is found, it processes the commands that begin on the next line.

Syntax

goto *label*

⇡Top of page

Parameters

label **:** Specifies the line in a batch program that you want to go to.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Working with command extensions

  If command extensions are enabled (that is, the default) and you use the **goto** command with a target label of **:EOF**, you transfer control to the end of the current batch script file and exit the batch script file without defining a label. When you use **goto** with the **:EOF** label, you must insert a colon before the label. For example:

  goto :EOF

  For a description of extensions to the **call** command that make this feature useful, see **cmd** in Related Topics.

- Using valid *label* values

  You can use spaces in the *label* parameter, but you cannot include other separators (for example, semicolons or equal signs). The **goto** command uses only the first eight characters of a label. For example, the following labels are equivalent and resolve to **:hithere0**:

  :hithere0

  :hithere01

  :hithere02
- Matching *label* with the label in the batch program

  The *label* value you specify must match a label in the batch program. The label within the batch program must begin with a colon (:). Windows XP 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, any commands on that line are ignored. If your batch program does not contain the label that you specify, the batch program stops and displays the following message:

  Label not found
- Using **goto** for conditional operations

  You can use **goto** with other commands to perform conditional operations. For more information about using **goto** for conditional operations, see **if** in Related Topics.

⇡Top of page

Examples

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

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

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Call

Cmd

If

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Gpresult

Displays Group Policy settings and Resultant Set of Policy (RSOP) for a user or a computer.

Syntax

gpresult [**/s** *Computer* [**/u** *Domain\User* **/p** *Password*]] [**/user** *TargetUserName*] [**/scope** {**user**|**computer**}] [**/v**] [**/z**]

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer. (Do not use backslashes.) The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user that is specified by *User* or *Domain\User*. The default is the permissions of the current logged-on user on the computer that issues the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/user *TargetUserName* : Specifies the user name of the user whose RSOP data is to be displayed.

/scope {user|computer} : Displays either **user** or **computer** results. Valid values for the **/scope** parameter are **user** or **computer**. If you omit the **/scope** parameter, **gpresult** displays both **user** and **computer** settings.

/v : Specifies that the output display verbose policy information.

/z : Specifies that the output display all available information about Group Policy. Because this parameter produces more information than the **/v** parameter, redirect output to a text file when you use this parameter (for example, **gpresult /z >policy.txt**).

/?: Displays help at the command prompt.

⇧Top of page

Remarks

- Group Policy is the primary administrative tool for defining and controlling how programs, network resources, and the operating system operate for users and computers in an organization. In an Active Directory environment, Group Policy is applied to users or computers on the basis of their membership in sites, domains, or organizational units.
- Because you can apply overlapping levels of policies to any computer or user, the Group Policy feature generates a resulting set of policies at logon. **Gpresult** displays the resulting set of policies that were enforced on the computer for the specified user at logon.

Top of page

Examples

The following examples show how you can use the **gpresult** command:

```
gpresult /user targetusername /scope computer
gpresult /s srvmain /u maindom\hiropln /p p@ssW23 /user targetusername /scope USER
gpresult /s srvmain /u maindom\hiropln /p p@ssW23 /user targetusername /z >policy.txt
gpresult /s srvmain /u maindom\hiropln /p p@ssW23
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Using command redirection operators

Top of page

Manage Your Profile |Legal |Contact Us

Gpupdate

Refreshes local and Active Directory-based Group Policy settings, including security settings. This command supersedes the now obsolete **/refreshpolicy** option for the **secedit** command.

Syntax

gpupdate [**/target:**{**computer**|**user**}] [**/force**] [**/wait:***value*] [**/logoff**] [**/boot**]

⇡Top of page

Parameters

/target:{computer|user} : Processes only the *Computer* settings or the current *User* settings. By default, both the computer settings and the user settings are processed.

/force : Ignores all processing optimizations and reapplies all settings.

/wait:*value* : Number of seconds that policy processing waits to finish. The default is 600 seconds. *0* means "no wait"; *-1* means "wait indefinitely."

/logoff : Logs off after the refresh has completed. This is required for those Group Policy client-side extensions that do not process on a background refresh cycle but that do process when the user logs on, such as user Software Installation and Folder Redirection. This option has no effect if there are no extensions called that require the user to log off.

/boot : Restarts the computer after the refresh has completed. This is required for those Group Policy client-side extensions that do not process on a background refresh cycle but that do process when the computer starts up, such as computer Software Installation. This option has no effect if there are no extensions called that require the computer to be restarted.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- If you make a syntax error, a syntax summary appears that looks similar to this Help topic.

⇡Top of page

Examples

The following examples show how you can use the **gpupdate** command:

gpupdate

gpupdate /target:computer

gpupdate /force /wait:100

gpupdate /boot

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Group Policy

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Graftabl

Enables the ability to display an extended character set in graphics mode.

Syntax

graftabl [*xxx*] [**/status**]

↑Top of page

Parameters

xxx **:** Specifies the code page for which you want the appearance of extended characters in graphics mode defined. The following table lists each valid code page identification number and its Country/Region or language.

Value	Country/Region
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 that **graftabl** is using.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- For information about using code pages, see **chcp** or **mode** (set device code pages) in Related Topics.
- **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.
- Understanding **graftabl** exit codes

  The following table lists each exit code and a brief description of it.

Exit code	Description
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 **graftabl**.

⇡Top of page

Examples

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

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Chcp

Mode

If

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Help

Provides online information about system commands (that is, non-network commands). Used without parameters, **help** lists and briefly describes every system command.

Syntax

{**help** [*command*]|[*command*]**/?**}

↑Top of page

Parameters

***command* :** Specifies the name of the command about which you want information.

↑Top of page

Remarks

- There are two ways to get online Help for a command. You can type **help** *command*, or you can type *command* **/?**. **/?** is slightly faster.
- For information about network commands, click **net help** in Related Topics.

↑Top of page

Examples

To see information about the **xcopy** command, type either of the following:

help xcopy
xcopy /?

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net help

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Helpctr

Starts Help and Support Center. Used without parameters, **helpctr** displays the Help and Support Center home page.

Syntax

helpctr [**/url** [*URL*]] [**/mode** [*URL*]] [**/hidden**] [**/fromstarthelp**]

⇡Top of page

Parameters

/url [*URL*] : Specifies the Uniform Resource Locator (URL) that you want to display within Help and Support Center.

/mode [*URL*] : Specifies an Extensible Markup Language (XML) Definition file that complies with the Launch_Description.dtd schema, which controls the context, layout and content of Help and Support Center.

/hidden : Starts Help and Support Center without displaying the user interface. This command can load a topic. You use this command for remotely administered script execution.

/fromstarthelp : Starts a new instance of Help and Support Center.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•Using *URL*

When you use **/url** or **/mode**, you can use quotation marks around the *URL* (that is, **"***URL***"**). If a space exists within *URL*, replace it by typing **%20** (that is, **"***URL***%20***Address***"**).

Top of page

Examples

To start a Remote Assistance Session with a .msrincident file, type:

```
HelpCtr /mode "hcp://CN=Microsoft%20Corporation,L=Redmond,S=Washington,C=US/Remote%20Assistance/RAClientLayout.xml" /url "hcp://CN=Microsoft%20Corporation,L=Redmond,S=Washington,C=US/Remote%20Assistance/Interaction/Client/rctoolScreen1.htm?IncidentFile=rahelp.msrincident"
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Set

Dir

Path

Top of page

Manage Your Profile |Legal |Contact Us

Hostname

Displays the host name portion of the full computer name of the computer.

Syntax

hostname

↑Top of page

Parameters

/? : Displays help at the command prompt.

↑Top of page

Remarks

- For more information about specifying the full computer name, see Related Topics.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

↑Top of page

Examples

To display the name of the computer, type:

hostname

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Specify your computer and workgroup names

Top of page

Manage Your Profile |Legal |Contact Us

If

Performs conditional processing in batch programs.

Syntax

if [**not**] **errorlevel** *number command* [**else** *expression*]

if [**not**] *string1*==*string2 command* [**else** *expression*]

if [**not**] **exist** *FileName command* [**else** *expression*]

If command extensions are enabled, use the following syntax:

if [**/i**] *string1 CompareOp string2 command* [**else** *expression*]

if cmdextversion *number command* [**else** *expression*]

if defined *variable command* [**else** *expression*]

Top of page

Parameters

not : Specifies that the command should be carried out 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 should be carried 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 (for example, *%1*). You do not need to use quotation marks around literal strings.

exist *FileName* : Specifies a true condition if *FileName* exists.

***CompareOp* :** Specifies a three-letter comparison operator. The following table lists valid values for *CompareOp*.

Operator	Description
EQU	equal to
NEQ	not equal to
LSS	less than
LEQ	less than or equal to
GTR	greater than
GEQ	greater than or equal to

/i : Forces string comparisons to ignore case. You can use **/i** on the *string1***==***string2* form of **if**. These comparisons are generic, in that if both *string1* and *string2* are both comprised of all numeric digits, the strings are converted to numbers and a numeric comparison is performed.

cmdextversion *number* : Specifies a true condition only if the internal version number associated with the Command Extensions feature of Cmd.exe is equal to or greater than *number*. The first version is 1. It is incremented by one when significant enhancements are added to the command extensions. The **cmdextversion** conditional is never true when command extensions are disabled (by default, command extensions are enabled).

defined *variable* : Specifies a true condition if *variable* is defined.

***expression* :** Specifies a command-line command and any parameters to be passed to the command in an *else* clause.

/? : Displays help at the command prompt.

Top of page

Remarks

- If the condition specified in an **if** command is true, the command that follows the condition is carried out. If the condition is false, the command in the **if** clause is ignored, and executes any command in the **else** clause, if one has been specified.

- When a program stops, it returns an exit code. You can use exit codes as conditions by using the *errorlevel* parameter.
- Using **defined** *variable*

If you use **defined** *variable*, the following three variables are added: **%errorlevel%**, **%cmdcmdline%**, and **%cmdextversion%**.

%errorlevel% expands into a string representation of the current value of *errorlevel*, provided that there is not already an environment variable with the name ERRORLEVEL, in which case you get the ERRORLEVEL value instead. The following example illustrates how you can use **errorlevel** after running a batch program:

```
goto answer%errorlevel%
:answer0
echo Program had return code 0
:answer1
echo Program had return code 1
goto end
:end
echo done!
```

You can also use the *CompareOp* comparison operators as follows:

```
if %errorlevel% LEQ 1 goto okay
```

%cmdcmdline% expands into the original command line passed to Cmd.exe prior to any processing by Cmd.exe, provided that there is not already an environment variable with the name **cmdcmdline**, in which case you get the **cmdcmdline** value instead.

%cmdextversion% expands into the a string representation of the current value of **cmdextversion**, provided that there is not already an environment variable with the name CMDEXTVERSION, in which case you get the CMDEXTVERSION value instead.

- Using the **else** clause

You must use the **else** clause on the same line as the command after the **if**. For example:

```
IF EXIST filename. (
del filename.
) ELSE (
echo filename. missing.
)
```

The following code does not work because you must terminate the **del** command by a new line:

```
IF EXIST filename. del filename. ELSE echo filename. missing
```

The following code does not work because you must use the **else** clause on the same line as the end of the **if** command:

```
IF EXIST filename. del filename.
ELSE echo filename. missing
```

If you want to format it all on a single line, use the following form of the original statement:

```
IF EXIST filename. (del filename.) ELSE echo filename. missing
```

Top of page

Examples

If the file Product.dat cannot be found, the following message appears:

if not exist product.dat echo Can't find data file

If an error occurs during the formatting of the disk in drive A, the following example displays an error message:

```
:begin
@echo off
format a: /s
if not errorlevel 1 goto end
echo An error occurred during formatting.
:end
```

```
echo End of batch program.
```

If no error occurs, the error message does not appear.

You cannot use the **if** command to test directly for a directory, but the null (NUL) device does exist in every directory. As a result, you can test for the null device to determine whether a directory exists. The following example tests for the existence of a directory:

```
if exist c:mydir\nul goto process
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Batch files

Cmd

Command-line reference A-Z

Top of page

Ipconfig

Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. Used without parameters, **ipconfig** displays the IP address, subnet mask, and default gateway for all adapters.

Syntax

ipconfig [**/all**] [**/renew** [*Adapter*]] [**/release** [*Adapter*]] [**/flushdns**] [**/displaydns**] [**/registerdns**] [**/showclassid** *Adapter*] [**/setclassid** *Adapter* [*ClassID*]]

Top of page

Parameters

/all : Displays the full TCP/IP configuration for all adapters. Without this parameter, **ipconfig** displays only the IP address, subnet mask, and default gateway values for each adapter. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connections.

/renew [*Adapter*] : Renews DHCP configuration for all adapters (if an adapter is not specified) or for a specific adapter if the *Adapter* parameter is included. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use **ipconfig** without parameters.

/release [*Adapter*] : Sends a DHCPRELEASE message to the DHCP server to release the current DHCP configuration and discard the IP address configuration for either all adapters (if an adapter is not specified) or for a specific adapter if the *Adapter* parameter is included. This parameter disables TCP/IP for adapters configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use **ipconfig** without parameters.

/flushdns : Flushes and resets the contents of the DNS client resolver cache. During DNS troubleshooting, you can use this procedure to discard negative cache entries from the cache, as well as any other entries that have been added dynamically.

/displaydns : Displays the contents of the DNS client resolver cache, which includes both entries preloaded from the local Hosts file and any recently obtained resource records for name queries resolved by the computer. The DNS Client service uses this information to resolve frequently queried names quickly, before querying its configured DNS servers.

/registerdns : Initiates manual dynamic registration for the DNS names and IP addresses that are configured at a computer. You can use this parameter to troubleshoot a failed DNS name registration or resolve a dynamic update problem between a client and the DNS server without rebooting the client computer. The DNS settings in the advanced properties of the TCP/IP protocol determine which names are registered in DNS.

/showclassid *Adapter* : Displays the DHCP class ID for a specified adapter. To see the DHCP class ID for all adapters, use the asterisk (*****) wildcard character in place of *Adapter*. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically.

/setclassid *Adapter* [*ClassID*] : Configures the DHCP class ID for a specified adapter. To set the DHCP class ID for all adapters, use the asterisk (*****) wildcard character in place of *Adapter*. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. If a DHCP class ID is not specified, the current class ID is removed.

/?: Displays help at the command prompt.

Top of page

Remarks

- The **ipconfig** command is the command-line equivalent to the **winipcfg** command, which is available in Windows Millennium Edition, Windows 98, and Windows 95. Although Windows XP does not include a graphical equivalent to the **winipcfg** command, you can use Network Connections to view and renew an IP address. To do this, open Network Connections, right-click a network connection, click **Status**, and then click the **Support** tab.
- This command is most useful on computers that are configured to obtain an IP address automatically. This enables users to determine which TCP/IP configuration values have been configured by DHCP, Automatic Private IP Addressing (APIPA), or an alternate configuration.
- If the *Adapter* name contains any spaces, use quotation marks around the adapter name (that is, "*Adapter Name*").
- For adapter names, **ipconfig** supports the use of the asterisk (*) wildcard character to specify either adapters with names that begin with a specified string or adapters with names that contain a specified string. For example, **Local*** matches all adapters that start with the string Local and ***Con*** matches all adapters that contain the string Con.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

⇡Top of page

Examples

To display the basic TCP/IP configuration for all adapters, type:

ipconfig

To display the full TCP/IP configuration for all adapters, type:

ipconfig /all

To renew a DHCP-assigned IP address configuration for only the **Local Area Connection** adapter, type:

ipconfig /renew "Local Area Connection"

To flush the DNS resolver cache when troubleshooting DNS name resolution problems, type:

ipconfig /flushdns

To display the DHCP class ID for all adapters with names that start with *Local*, type:

ipconfig /showclassid Local*

To set the DHCP class ID for the **Local Area Connection** adapter to *TEST*, type:

ipconfig /setclassid "Local Area Connection" TEST

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

TCP/IP utilities and services

Ping

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ipseccmd

Configures Internet Protocol Security (IPSec) policies in a directory service or in a local or remote registry. Ipseccmd is a command-line alternative to the IP Security Policies Microsoft Management Console (MMC) snap-in. Ipseccmd has three modes: dynamic mode, static mode, and query mode.

To view the command syntax, click a command:

ipseccmd dynamic mode

You can use Ipseccmd dynamic mode to add anonymous rules to the existing IPSec policy by adding them to the IPSec security policies database. The rules added will be present even after the IPSEC Services service is restarted. The benefit of using dynamic mode is that the rules you add coexist with domain-based IPSec policy. Dynamic mode is the default mode for Ipseccmd.

Syntax

- To add a rule, use the following syntax:

  ipseccmd [\\ComputerName] **-f** *FilterList* [**-n** *NegotiationPolicyList*] [**-t** *TunnelAddr*] [**-a** *AuthMethodList*] [**-1s** *SecurityMethodList*] [**-1k** *MainModeRekeySettings*] [**-1p**] [**-1f** *MMFilterList*] [**-1e** *SoftSAExpirationTime*] [**-soft**] [**-confirm**] [{**-dialup** | **-lan**}]

- To delete all dynamic policies, use the following syntax:

  ipseccmd -u

Top of page

Parameters

\\ComputerName **:** Specifies the computer name of a remote computer to which you want to add a rule.

-f *FilterList* : Required for first syntax. Specifies one or more filter specifications, separated by spaces, for quick mode security associations (SAs). Each filter specification defines a set of network traffic affected by this rule.

-n *NegotiationPolicyList* : Specifies one or more security methods, separated by spaces, for securing traffic defined by the filter list.

-t *TunnelAddr* : Specifies the tunnel endpoint for tunnel mode as either an IP address or a DNS domain name.

-a *AuthMethodList* : Specifies one or more authentication methods, separated by spaces.

-1s *SecurityMethodList* : Specifies one or more key exchange security methods, separated by spaces.

-1k *MainModeRekeySettings* : Specifies main mode SA rekey settings.

-1p : Enables master key perfect forward secrecy.

-1f *MMFilterList* : Specifies one or more filter specifications for main mode SAs, separated by spaces.

-1e *SoftSAExpirationTime* : Specifies the expiration time for soft SAs in seconds.

-soft : Enables soft SAs.

-confirm : Specifies that a confirmation prompt appears before the rule or policy is added.

{-dialup | -lan} : Specifies whether the rule applies only to remote access or dial-up connections or whether the rule applies only to local area network (LAN) connections.

-u : Required for the second syntax. Specifies that all dynamic rules are deleted.

/?: Displays help at the command prompt.

Top of page

Remarks

- Ipseccmd cannot be used to configure rules on computers running Windows 2000.
- If you do not specify the *ComputerName* parameter, the rule is added to the local computer.
- If you use the *ComputerName* parameter, you must use it before all other parameters, and you must have administrator permissions on the computer to which you want to add the rule.

- For the **-f** parameter, a filter specification is one or more filters that are separated by spaces and defined by the format:

  SourceAddress/*SourceMask*:*SourcePort*=*DestAddress*/*DestMask*:*DestPort*:*Protocol*

- *SourceMask*, *SourcePort*, *DestMask*, and *DestPort* are optional. If you omit them, the mask of 255.255.255.255 and all ports are used for the filter.
- *Protocol* is optional. If you omit it, all protocols are used for the filter. If you specify a protocol, you must specify the port or precede the protocol with two colons (::). (See the first example for dynamic mode.) The protocol must be the last item in the filter. You can use the following protocol symbols: ICMP, UDP, RAW, or TCP.
- You can create mirrored filters by replacing the equals sign (=) with a plus sign (+).
- You can replace *SourceAddress*/*SourceMask* or *DestAddress*/*DestMask* with the values in the following table.

Value	Description
0	My address or addresses
*****	Any address
DNSName	DNS domain name. If the DNS name resolves to multiple addresses, it is ignored.
GUID	A globally unique identifier (GUID) of a local network interface in the form {12345678-1234-1234-1234-123456789ABC}. Specifying a GUID is not supported when the **-n** parameter is used in static mode.

- You can enable the default response rule by specifying the filter specification of **default**.
- You can specify a permit filter by surrounding the filter specification with parentheses. You can specify a blocking filter by surrounding the filter specification with brackets ([ ]).
- If you are using Internet address class-based subnet masks (the subnet masks are defined along octet boundaries), you can use wildcard notation to specify subnet masks. For example, 10.*.*.* is the same as 10.0.0.0/255.0.0.0 and 10.92.*.* is the same as 10.92.0.0/255.255.0.0.

Filter examples

To create mirrored filters to filter TCP traffic between Computer1 and Computer2, type:

Computer1+Computer2::TCP

To create a filter for all TCP traffic from the subnet 172.31.0.0/255.255.0.0, port 80, to the subnet 10.0.0.0/255.0.0.0, port 80, type:

172.31.0.0/255.255.0.0:80=10.0.0.0/255.0.0.0:80:TCP

To create a mirrored filter that permits traffic between the local IP address and the IP address 10.2.1.1, type:

(0+10.2.1.1)

- For the **-n** parameter, one or more negotiation policies are separated by spaces and follow one of the following forms:

  - **esp[***EncrypAlg***,***AuthAlg***]***Rekey***PFS[***Group***]**
  - **ah[***HashAlg***]**
  - **ah[***HashAlg***]+esp[***EncrypAlg***,***AuthAlg***]**

  where *EncrypAlg* can be **none**, **des**, or **3des**, *AuthAlg* can be **none**, **md5**, or **sha**, and *HashAlg* can be **md5** or **sha**.

  - The configuration **esp[none,none]** is not supported.
  - The **sha** parameter refers to the SHA1 hash algorithm.
  - The *Rekey* parameter is optional, and it specifies the number of kilobytes (indicated by placing a **K** after the number) or the number of seconds (indicated by placing an **S** after the number) that precede a rekeying of the quick mode SA. To specify both rekey parameters, separate the two numbers with a slash (/). For example, to rekey the quick mode SA every hour and after every 5 megabytes of data, type:

    3600S/5000K
  - The **PFS** parameter is optional, and it enables session key perfect forward secrecy. By default, session key perfect forward secrecy is disabled.
  - The *Group* parameter is optional, and it specifies the Diffie-Hellman group for session key perfect forward secrecy. For the Low(1) Diffie-Hellman group, specify **PFS1** or **P1**. For the Medium(2) Diffie-Hellman group, specify **PFS2** or **P2**. By default, the group value for session key perfect forward secrecy is taken from the current main mode settings.
  - If you do not specify negotiation policies, the default negotiation policies are the following:

    - esp[3des,sha]
    - esp[3des,
      md5]
    - esp[des,sha]
    - esp[des,md5]
- If you omit the **-t** parameter, IPSec transport mode is used.

- For the **-a** parameter, one or more authentication methods are separated by spaces and are in one of the following forms:

  - **preshare:"***PresharedKeyString***"**
  - **kerberos**
  - **cert:"***CAInfo***"**

  The *PresharedKeyString* parameter specifies the string of characters of the preshared key. The *CAInfo* parameter specifies the distinguished name of the certificate as displayed in the IP Security Policies snap-in when the certificate is selected as an authentication method for a rule. The *PresharedKeyString* and *CAInfo* parameters are case-sensitive. You can abbreviate the method by using the first letter: **p**, **k**, or **c**. If you omit the **-a** parameter, the default authentication method is Kerberos.
- For the **-1s** parameter, one or more key exchange security methods are separated by spaces and defined by the following format:

  EncrypAlg-*HashAlg*-*GroupNum*

  where *EncrypAlg* can be **des** or **3des**, *HashAlg* can be **md5** or **sha**, and *GroupNum* can be **1** for the Low(1) Diffie-Hellman group or **2** for the Medium(2) Diffie-Hellman group. If you omit the **-1s** parameter, the default key exchange security methods are 3des-sha-2, 3des-md5-2, des-sha-1, and des-md5-1.
- For the **-1k** parameter, you can specify the number of quick mode SAs (indicated by placing a **Q** after the number) or the number of seconds (indicated by placing an **S** after the number) to rekey the main mode SA. To specify both rekey parameters, you must separate the two numbers with a slash (/). For example, to rekey the main mode SA after every 10 quick mode SAs and every hour, type:

  10Q/3600S

  If you omit the **-1k** parameter, the default values for main mode rekey are an unlimited number of quick mode SAs and 480 minutes.
- Master key perfect forward secrecy is disabled by default.
- For the **-1f** parameter, the syntax for specifying main mode filter specifications is the same as for the **-f** parameter, except that you cannot specify permit filters, blocking filters, ports, or protocols. If you omit the **-1f** parameter, the main mode filters are automatically created based on the quick mode filters.
- If you omit the **-1e** parameter, the expiration time for soft SAs is 300 seconds. However, soft SAs are disabled unless you include the **-soft** parameter.
- Confirmation is available for dynamic mode only.
- If you specify neither the **-dialup** parameter nor the **-lan** parameter, the rule applies to all adapters.

⇡Top of page

Examples

To create a rule that uses the Authentication Header (AH) with MD5 hashing for all traffic to and from the local computer, type:

ipseccmd -f 0+* -n ah[md5]

To create a tunnel rule for traffic from 10.2.1.1 and 10.2.1.13 using the tunnel endpoint 10.2.1.13, with AH tunnel mode using the SHA1 hash algorithm, with master key perfect forward secrecy enabled, and with a confirmation prompt for the rule before it is created, type:

ipseccmd -f 10.2.1.1=10.2.1.13 -t 10.2.1.13 -n ah[sha] -1p -c

To create a rule on the computer named corpsrv1 for all traffic between the computers named corpsrv1 and corpsrv2, using the combination of both AH and Encapsulating Security Payload (ESP), with preshared key authentication, type:

ipseccmd \\corpsrv1 -f corpsrv2+corpsrv1 -n ah[md5]+esp[des,sha] -a p:"corpauth"

ipseccmd static mode

You can use Ipseccmd static mode to create named policies and named rules. You can also use static mode to modify existing policies and rules, provided they were originally created with Ipseccmd. The syntax for static mode combines the syntax for dynamic mode with parameters that enable it to work at a policy level.

⇡Top of page

Syntax

ipseccmd *DynamicModeParameters* **-w** *Type*[**:***Location*] **-p** *PolicyName*[**:***PollInterval*] **-r** *RuleName* [{**-x** | **-y**}] [**-o**]

⇡Top of page

Parameters

***DynamicModeParameters* :** Required. Specifies a set of dynamic mode parameters for an IPSec rule as described earlier.

-w *Type*[:*Location*] : Required. Specifies that the policies and rules are written to the local registry, to the registry of a remote computer, or to an Active Directory domain.

-p *PolicyName*[:*PollInterval*] : Required. Specifies the name of the policy and how often, in minutes, the policy is checked for changes. If *PolicyName* contains any spaces, use quotation marks around the text (that is, "*Policy Name*").

-r *RuleName* : Required. Specifies the name of the rule. If *RuleName* contains any spaces, use quotation marks around the text (that is, "*Rule Name*").

[{-x | -y}] : Specifies whether the local registry policy is assigned. The **-x** parameter specifies that the local registry policy is assigned. The **-y** parameter specifies that the local registry policy is unassigned.

-o : Specifies that the rule or policy should be deleted.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- For the **-w** parameter, the *Type* is either **reg** to specify the registry of the local computer or a remote computer or **ds** to specify Active Directory.

  - If you specify **reg** for the *Type* parameter but you do not use the *Location* parameter, the rule is created for the registry of the local computer.
  - If you specify **reg** for the *Type* parameter and you specify the name of a remote computer for the *Location* parameter, the rule is created for the registry of the remote computer you specify.
  - If you specify **ds** for the *Type* parameter but you do not use the *Location* parameter, the rule is created for the Active Directory domain of which the local computer is a member.
  - If you specify **ds** for the *Type* parameter and you specify an Active Directory domain for the *Location* parameter, the rule is created for the specified domain.
- For the **-p** parameter, if a policy with this name already exists, the rule you specify is added to the policy. Otherwise a policy is created with the name you specify. If you specify an integer for the *PollInterval* parameter, the polling interval for the policy is set to that number of minutes.

- For the **-r** parameter, if a rule with that name already exists, the rule is modified to reflect the parameters you specify in the command. For example, if you include the **-f** parameter for an existing rule, only the filters of that rule are replaced. If no rule exists with the name you specify, a rule with that name is created.
- For the **-o** parameter, all aspects of the specified policy are deleted. Do not use this parameter if you have other policies that point to the objects in the policy you want to delete.
- Static mode usage differs from dynamic mode usage in one respect. Using dynamic mode, you indicate permit and blocking filters in *FilterList*, which you identify using the **-f** parameter. Using static mode, you indicate permit and blocking filters in *NegotiationPolicyList*, which you identify using the **-n** parameter. In addition to the parameters described for *NegotiationPolicyList* under dynamic mode, you can also use the **block**, **pass**, or **inpass** parameters in static mode. The following table list these parameters and a description of their behavior.

Parameter	Description
block	The rest of the policies in *NegotiationPolicyList* are ignored, and all of the filters are made blocking filters.
pass	The rest of the policies in *NegotiationPolicyList* are ignored, and all of the filters are made permit filters.
inpass	Inbound filters will allow initial communication to be unsecured, but responses will be secured using IPSec.

Top of page

Examples

To create a policy named Default Domain Policy with a 30-minute polling interval in the Active Directory domain of which the local computer is a member, with a rule named Secured Servers for traffic between the local computer and computers named SecuredServer1 and SecuredServer2, using Kerberos and preshared key authentication methods, type:

ipseccmd -f 0+SecuredServer1 0+SecuredServer2 -a k p:"corpauth" -w ds -p "Default Domain Policy":30 -r "Secured Servers"

To create and assign a local policy named Me to Anyone, with a rule named Secure My Traffic, using a mirrored filter for any traffic to the local computer, using a preshared key as the authentication method, type:

ipseccmd -f 0+* -a p:"localauth" -w reg -p "Me to Anyone" -r "Secure My Traffic" -x

ipseccmd query mode

You can use Ipseccmd query mode to display data from the IPSec security policies database.

Top of page

Syntax

ipseccmd [\\ComputerName] **show** {{[**filters**] | [**policies**] | [**auth**] | [**stats**] | [**sas**]} | **all**}

Top of page

Parameters

\\ComputerName **:** Specifies, by name, the remote computer for which you want to display data.

show : Required. Indicates that Ipseccmd should run in query mode.

filters : Displays main mode and quick mode filters.

policies : Displays main mode and quick mode policies.

auth : Displays main mode authentication methods.

stats : Displays statistics about Internet Key Exchange (IKE) and IPSec.

sas : Displays main mode and quick mode security associations (SAs).

all : Displays all of the above types of data.

/?: Displays help at the command prompt.

Top of page

Remarks

- Ipseccmd cannot be used to display IPSec data for computers running Windows 2000.
- If you do not use the *ComputerName* parameter, information about the local computer is displayed.

- If you use the *ComputerName* parameter, you must use it before all other parameters, and you must have administrator permissions on the computer for which you want to display information.

Top of page

Examples

To display the main mode and quick mode filters and policies for the local computer, type:

ipseccmd show filters policies

To display all IPSec information for the remote computer Server1, type the following command:

ipseccmd \\Server1 show all

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Start the IP Security Policies snap-in

Troubleshooting tools

Command-line reference A-Z

Internet Protocol security overview

Top of page

Manage Your Profile |Legal |Contact Us

Ipxroute

Displays and modifies information about the routing tables used by the IPX protocol. Used without parameters, **ipxroute** displays the default settings for packets that are sent to unknown, broadcast, and multicast addresses.

Syntax

ipxroute servers [**/type=***x*]

ipxroute ripout *network*

ipxroute resolve {**guid** | **name**} {*guid* | *AdapterName*}

ipxroute board=*n* [**def**] [**gbr**] [**mbr**] [**remove=***xxxxxxxxxxxx*]

ipxroute config

⇡Top of page

Parameters

servers [/type=*x*] : Displays the Service Access Point (SAP) table for the specified server type. *x* must be an integer. For example, **/type=4** displays all file servers. If you do not specify **/type**, **ipxroute servers** displays all types of servers, listing them by server name.

ripout *network* : Discovers if *network* is reachable by consulting the IPX stack's route table and sending out a rip request if necessary. *Network* is the IPX network segment number.

resolve {guid | name} {*guid* | *AdapterName*} : Resolves the name of the guid to its friendly name, or the friendly name to its guid.

board=*n* : Specifies the network adapter for which to query or set parameters.

def : Sends packets to the ALL ROUTES broadcast. If a packet is transmitted to a unique Media Access Card (MAC) address that is not in the source routing table, **ipxroute** sends the packet to the SINGLE ROUTES broadcast by default.

gbr : Sends packets to the ALL ROUTES broadcast. If a packet is transmitted to the broadcast address (FFFFFFFFFFFF), **ipxroute** sends the packet to the SINGLE ROUTES broadcast by default.

mbr : Sends packets to the ALL ROUTES broadcast. If a packet is transmitted to a multicast address (C000xxxxxxxx), **ipxroute** sends the packet to the SINGLE ROUTES broadcast by default.

remove=*xxxxxxxxxxxx* : Removes the given node address from the source routing table.

config : Displays information about all of the bindings for which IPX is configured.

/? : Displays help at the command prompt.

Top of page

Examples

To display the network segments that the workstation is attached to, the workstation node address, and frame type being used, type the following command:

ipxroute config

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

TCP/IP utilities and services

Top of page

Manage Your Profile |Legal |Contact Us

Irftp

Sends files over an infrared link. Used without parameters or used with **/s**, **irftp** opens the **Wireless Link** dialog box, where you can select the files that you want to send without using the command line.

Syntax

irftp [*Drive***:**\] [[*Path*] *FileName*] [**/h**]

irftp /s

⇡Top of page

Parameters

***Drive*:\ :** Specifies the drive that contains the files that you want to send over an infrared link.

[*Path*] *FileName* : Specifies the location and name of the file or set of files that you want to send over an infrared link. If you specify a set of files, you must specify the full path for each file.

/h : Specifies hidden mode. When hidden mode is used, the files are sent without displaying the **Wireless Link** dialog box.

/s : Opens the **Wireless Link** dialog box, so that you can select the file or set of files that you want to send without using the command line to specify the drive, path, and file names.

⇡Top of page

Remarks

- Before using this command, verify that the devices that you want to communicate over an infrared link have infrared functionality enabled and working correctly, and that an infrared link is established between the devices.

⇡Top of page

Examples

To send a Word document named MyWorkDoc, located in a folder named MyWorkFiles (on drive C of the source computer), to the target computer, type:

irftp C:\MyWorkFiles\MyWorkDoc.doc /h

To send two Word documents, MyWorkDoc1 and MyWorkDoc2, both located in a folder named MyWorkFiles (on drive C of the source computer), to the target computer, type:

irftp C:\MyWorkFiles\MyWordDoc1.doc C:\MyWorkFiles\MyWordDoc2.doc /h

Hidden mode is used, so that the **Wireless Link** dialog box is not displayed.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Send files using the command line

Verify infrared support on a computer

Establish an infrared link

Label

Creates, changes, or deletes the volume label (that is, the name) of a disk. Used without parameters, **label** changes the current volume label or deletes the existing label.

Syntax

label [*Drive***:**][*label*]

label [**/MP**][*volume*][*label*]

⇡Top of page

Parameters

***Drive*: :** Specifies the drive letter (followed by a colon) of the disk you want to name.

***label* :** Specifies the name for the volume.

/MP : Specifies that the volume should be treated as a mount point or volume name.

***volume* :** Specifies the drive letter (followed by a colon), mount point, or volume name. If a volume name is specified, the **/MP** parameter is unnecessary.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Windows displays the volume label as part of the directory listing. If a volume serial number exists, Windows displays this number as well.
- Label command messages

  If you do not specify a label when you use the **label** command, **label** displays a message in the following format:

  ```
  Volume in drive A is xxxxxxxxxxx
  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, **label** 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.

FAT volume labels cannot contain any of the following characters:

* ? / \ | . , ; : + = [ ] < > "

This limitation does not apply to NTFS volumes.

For FAT volumes, volume labels are stored as uppercase regardless of whether they contain lowercase letters. NTFS volume labels retain and display the case used when the label was created.

⇡Top of page

Examples

To label a disk in drive A that contains sales information for July, type:

label a:sales-july

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Dir

Vol

Lodctr

Registers new Performance counter names and Explain text for a service or device driver, and saves and restores counter settings and Explain text.

Syntax

lodctr [\\ComputerName] *FileName* [**/s:***FileName*] [**/r:***FileName*]

⇡Top of page

Parameters

[\\ComputerName] *FileName* Registers the Performance counter name settings and Explain text provided in initialization file *FileName*. If you do not specify the *ComputerName*, **lodctr** uses the local computer by default. **/s:***FileName* Saves Performance counter registry settings and Explain text to file *FileName*. **/r:***FileName* Restores Performance counter registry settings and Explain text from file *FileName*.

Caution

- If you use the **lodctr /r** command, you will overwrite all Performance counter registry settings and Explain text, replacing them with the configuration defined in the file specified.

/?Displays help at the command prompt.

⇡Top of page

Remarks

- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Examples

To save the current Performance registry settings and counter Explain text to file perf backup1.txt, type:

lodctr /s:"perf backup1.txt"

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Add counters to System Monitor

Troubleshooting

Command-line reference A-Z

Top of page

Logman

Manages and schedules performance counter and event trace log collections on local and remote systems.

Syntax Verbs

Logman [**create** {*counter* | *trace*} *collection_name*] [**start** *collection_name*]
[**stop** *collection_name*] [**delete** *collection_name*] [**query** {*collection_name*|**providers**}]
[**update** *collection_name*]

⇡Top of page

Parameters

create {*counter* | *trace*} *collection_name* : Creates collection queries for either counter or trace collections. You can use command line options to specify settings.

start *collection_name* : Starts the data collection query *collection_name*. Use this option to change from scheduled collections to manual ones. Use the **update** parameter in the command line with begin-time (-b), end-time (-e), or repeat-time (-rt) to reschedule collections.

stop *collection_name* : Stops the data collection query *collection_name*. Use this option to change from scheduled collections to manual ones. Use the **update** parameter in the command line with begin-time (-b), end-time (-e), or repeat-time (-rt) to reschedule collections.

delete *collection_name* : Deletes the data collection query *collection_name*. If the *collection_name* does not exist, you will receive an error.

query {*collection_name*|providers} : If no *collection_name* or **providers** are given, the status of all existing collection queries are displayed. Use *collection_name* to display the properties of a specific collection. To display the properties on remote computers, use the **-s** *remote computer* option in the command line. Use **providers** as your keyword in place of *collection_name* to display the registered providers installed on your local system. To list registered providers installed on the remote system, use the **-s** option in the command line.

update *collection_name* : Updates collection queries for counter and trace collections. For counter collections, modifications to the query will stop, and then restart the collections. For trace collections, use the following parameters in the command line to query without stopping the collection: **-p** *provider* [**(***flags*[**,***flags* ...]**)**] *Level*, **- max** *n*, **- o** *PathName*, **-ft** *mm:ss*, or **-fd**.

Top of page

Syntax Options

[**-s** *computer_name*] [**-config** *FileName*] [**-b** *M/d/yyyy h:mm:ss*[*AM* | *PM*]] [**-e** *M/d/yyyy h:mm:ss* [*AM* | *PM*]] [**-m** [*start*] [*stop*]] [-[-]**r**] [**-o** {*Path* | *DSN!counter_log*}] [**-f** {*bin* | *bincirc* | *csv* | *tsv* | *SQL*}] [-[-]**a**] [-[-]**v** [{*nnnnn* | *mmddhhmm*}]] [-[-] **rc** [*FileName*]] [-[-] **max** {*value*}] [-[-] **cnf** *[[hh:]]mm:]ss*] [**-c** {*Path* [*path* ...] | **-cf** *FileName*}] [**-si** *[[hh:]mm:]ss*] [**-ln** *logger_name*] [**-ets**] [-[-] **rt**] [**-p** {*GUID* | *provider* [(*flags* [,*flags*...]}] *Level* | **-pf** *FileName*}] [-[-] **ul**] [**-bs** *value*] [**-ft** *[[hh:]mm:]ss*] [**-nb** *min max*] [**-fd** *logger name*] [-[-]**u** *user password*] [**-rf** *[[hh:]mm:]ss*] [**-y**] [**-mode** {*trace_mode* [*trace_mode* ...]}]

Top of page

Parameters

-s *computer_name* : Specifies that create, start, stop, delete, query, or update commands will be performed on the remote system. By default, the local system is used for commands.

-config *FileName* : Specifies the pathname of the settings file that contains command line parameters.

-b *M/d/yyyy h:mm:ss*[*AM* | *PM*] : Specifies begin-time for collections in a 24-hour format. You can also specify begin-time for collections in a 12-hour format by adding AM or PM in the command line. By default, the current day and time is used unless otherwise specified. Use the manual start option to start the collection immediately.

-e *M/d/yyyy h:mm:ss*[*AM* | *PM*] : Specifies end-time for collections in a 24-hour format. You can also specify end-time for collections in a 12-hour format by adding AM or PM in the command line. By default, the current day and time is used unless otherwise specified. Use the manual stop and then the repeat option to specify a stop time before the actual current time, or you will receive an error message.

-m start stop : Specifies that collections start and stop manually by using the **start** and **stop** parameters in the command line. You cannot use the **-m start** and **-b**, or the **-m stop** and **-e**, or **-rf** parameters together in your command line for the same query.

-r : Repeats the collection every day at the time periods specified by the **-b** and **-rf** options, or the **-b** and **-e** options. This command is only valid for begin- and end-times specified on the same day, month, and year.

--r : Turns off the repeat option.

-o {*Path* | *DSN!counter_log*} : Specifies the pathname of the output file that collects performance counter and trace data, or the location of the SQL database and dataset. To specify SQL using the DSN!counter_log format, use the **-f** option in the command line. By default, the collection log file name is the collection query name suffixed by either .blg for performance counters, or .etl for trace data.

-f {bin | bincirc | csv | tsv | SQL} : Specifies the file format used for collecting performance counter and trace data. You can use binary, circular binary, comma and tab separated, or SQL database formats when collecting performance counters. You must use the **-o** option in the command line with the DNS!counter_log option. For SQL database formats, the Database System Name (DSN) must be predefined, and privileges granted to write to the database. The dataset *counter_log* is created in the database, and is specified by the DSN. Defaults to binary.

-a : Use this option to append the file.

--a : Turns off the append command option, and reverts to the overwrite mode.

-v {nnnnn | mmddhhmm} : Attaches the version control information to the end of the output file and path name. Use numeric *nnnnn* format, or date format *mmddhhmm* (month, day, 24-hour, minute) for version control.

--v : Turns off the version option.

-rc *FileName* : Specifies to run this command after the file is closed either at the end, or during the collection period. Use the **-rf** option in conjunction with **-cnf** to close the files during the collection periods. Using the **-stop** option will not turn off this command. Commands always run in the foreground.

--rc : Turns off the run this command option.

-max *value* : Specifies the maximum size of the collected log file in megabytes. If the log file exceeds the maximum size, the collection will stop. For a SQL database, the maximum size is the number of records to be written.

--max : Turns off the maximum size limit option. This is the default option.

-cnf *[[hh:]mm:]ss* : Creates a new file when output files exceed a maximum size, or when the time specified elapses. You must include the **-v** option when executing this command. By default, only one log file is created during each collection.

--cnf : Turns off the create-new-file option.

-c {*Path* [*path* ...] | -cf *FileName*} : Specifies the performance counter path to log, or specifies the pathname of the log file that lists these counters. To list multiple counter paths, separate the command line by a space, or use the **-cf** option to list counter paths in an input file, one per line.

The general format for counter paths is as follows: [\\Computer]*object*[parent/*instance#index*]*counter*] where:

The parent, instance, index, and counter components of the format may contain either a valid name or a wildcard character. The computer, parent, instance, and index components are not necessary for all counters.

You determine the counter paths to use based on the counter itself. For example, the *LogicalDisk* object has an instance *index*, so you must provide the *#index* or a wildcard. Therefore, you could use the following format:

\LogicalDisk(*/*#*)*

In comparison, the *Process* object does not require an instance *index*. Therefore, you could use the following format:

\Process(*)\ID Process

The following is a list of the possible formats:

•\\machine\object(parent/instance#index)\counter
•\\machine\object(parent/instance)\counter
•\\machine\object(instance#index)\counter
•\\machine\object(instance)\counter
•\\machine\object\counter
•\object(parent/instance#index)\counter
•\object(parent/instance)\counter
•\object(instance#index)\counter
•\object(instance)\counter
•\object\counter

If a wildcard character is specified in the parent name, all instances of the specified object that match the specified instance and counter fields will be returned.

If a wildcard character is specified in the instance name, all instances of the specified object and parent object will be returned if all instance names corresponding to the specified index match the wildcard character.

If a wildcard character is specified in the counter name, all counters of the specified object are returned.

Partial counter path string matches (for example, pro*) are not supported.

-si ***[[hh:]mm:]ss*** **:** Specifies sample intervals for performance counter collection in hours, minutes, and seconds. Default is 15-seconds.

-ln ***logger_name*** **:** Specifies a user-defined name for the event trace logging session. By default, the collection name is used as the logger name.

-ets : Creates and starts an event trace session with the options specified on the command line. You can use this optional parameter with the **create trace**, **update**, **query**, and **delete** parameters.

-rt : Specifies that the event trace session run in real-time mode, and not log to a file. By default, the data logs to a file.

--rt : Turns off the real-time logging option.

-p {*GUID* | *provider* [(*flags*[,*flags* ...])] *Level*] | -pf {*FileName*} : Specifies the providers (trace data collectors) to use for trace data collection. Use logman query providers to find the *pname* (named providers) from the registered provider list. Use the **-pf** option to list multiple providers. The **-pf** option identifies the input file containing the provider names. The provider names are enclosed by quotation marks (""), or with GUIDs enclosed by braces, flag masks, and integers (enable level). The flags are either in hexadecimal (OXFFFF) or (flag, flag) format.

-ul : Specifies that the event trace session is run in user mode. If you use the **-ul** option, only one provider can be enabled for the event trace session.

--ul : Specifies that the user mode is turned off, and the event trace session is run in kernel mode.

-bs ***value*** **:** Specifies the buffer size in *n* kilobytes for trace data collections.

-ft *[[hh:]mm:]ss* : Specifies the flush timer interval in minutes and seconds for trace data collections.

-nb *min max* : Specifies the minimum and maximum number of buffers for trace data collection. Minimum default is the number of processors on the system plus two. Maximum default is at 25.

-fd *logger_name* : Flushes all the active buffers of an existing event trace session to a disk. Use this command in conjunction with the **-ln** option.

-u *user password* : Specifies the account name and password the collection query uses on local or remote systems. To start collecting data for collection queries, log Performance Logs and Alerts to the remote system. You can use ***** as your password in the command line to produce a prompt for the password.

--u : Resets the account name to the Performance Logs and Alerts service account.

-rf *[[hh:]mm:]ss* : Specifies that collections run for a set period of time.

-y : Overwrites the settings for collection name, and then applies new ones without querying the end user.

-mode {*trace_mode* [*trace_mode* ...]} : Specifies advanced options for trace sessions only where *trace_mode* can be either **globalsequence**, **localsequence** or **pagedmemory**. **Globalsequence** specifies that the event tracer add a sequence number to every event it receives irrespective of which trace session received the event. **Localsequence** specifies that the event tracer add sequence numbers for events received at a specific trace session. When the **localsequence** option is used, duplicate sequence numbers can exist across all sessions but will be unique within each trace session. **Pagedmemory** specifies that the event tracer use paged memory rather than the default non-paged memory pool for its internal buffer allocations.

/? : Displays help at the command prompt.

Top of page

Remarks

•Valid options for the command-line verbs **create update**, **start**, **stop**, **delete**, and **query** are:

-s *remote machine name*

-[-]**u** *Domain/username password*

- Valid options for **create** and **update**, and common options for *counter* and *trace* are:

  -y

  -b *M/d/yyyy h:mm:ss [AM|PM]*

  -e *M/d/yyyy h:mm:ss [AM|PM]*

  -rf *hh:mm:ss*

  -m *start stop*

  -f *bin|bincirc*

  -[-]r

  -o *PathName*

  -[-]a

  -[-]v *nnnnn|mmddhhmm*

  -[-]rc *command pathname*

  -[-]max *n*

  -[-]cnf *hh:mm:ss*
- Common options for *counters* only are:

  -f *bin|bincirc|csv|tsv|SQL*

  -o *PathName | DSN!counter_log*
- Common options for **create** *counter* are:

  -c *counterpath* | **-cf** *input file*

  -si [[*hh:*]*mm:*]*ss*

  These options update the counter, and will stop and start collections.

- Options for **update** for trace collection are:

  - max *n*

  - o *PathName*

  -ft *mm:ss*

  -fd

  These options will query trace collections without stopping the collections.
- Valid options with *counters* only are:

  -f *bin|bincirc|csv|tsv|SQL*

  -o *PathName* | *DSN!counter_log*
- Valid options with **create** *counter* commands are:

  counterpath | **-cf** *input file*

  -si *hh:mm:ss*
- Valid options for **create** *trace* commands are:

  -ln *logger name*

  -[-]rt

  -p *Guid* | *provider (flags[,flags ...]) level* | **-pf** *FileName*

  -[-]ul

  -bs *n*

  -ft *mm:ss*

  -nb *min max*

  -fd *logger name*

  -ets

- Using the **-config** option

  The contents of the setting file used with the **-config** option should have the following format:

  [*command_option*]

  value

  where *command_option* is a command line option and *value* specifies its value. For example:

  [create]

  counter logx

  [-s]

  mysystem

  [-u]

  mypassword
- Using the **-mode** option

  You should only use this option if you are an advanced user.
- Managing Performance monitor

  You can only use **Logman** to manage systems running Windows 2000 or Windows XP.
- For more information about incorporating **Logman** into your Windows Management Instrumentation (WMI) scripts, see Scripting Logs and Monitoring at the Microsoft Windows Resource Kits Web site.

Top of page

Examples

Date formats in the following examples are for US local time only.

To create daily counter collection queries with begin and end times, repeat collections, version control numbers, counter paths and sample intervals, type:

Logman create counter daily_perf_log -b 7/27/2000 13:00:00 -e 7/27/2000 15:00:00 -r -v

```
mmddhhmm -c "\Processor(_Total)\% Processor Time" "\Memory\Available bytes" -si 00:15 -o "c:\perflogs\daily_log"
```

To create daily collection queries with begin and end times, repeat collections, output file collections, version control numbers, counter paths and sample intervals, type:

```
Logman create counter daily_perf_log -b 7/27/2000 13:00:00 -e 7/27/2000 15:00:00 -r -o "c:\perflogs\daily_log" -v mmddhhmm -c "\Processor(_Total)\% Processor Time" "\Memory\Available bytes" -si 00:15
```

To create daily collection queries using the config file, instead of the command line file, type:

```
Logman -config file daily_perf.txt
```

To create daily trace collection queries with begin and end times, repeat collections, version control numbers, provider names, input and output file collections, type:

```
Logman create trace daily_kernel_trace_log -b 7/27/2000 13:00:00 -e 7/27/2000 15:00:00 -r -v mmddhhmm -p "Windows Kernel Trace" 0xFFFFFFFF -rf 100 -o "c:\perflogs\daily_nt_trace"
```

To create daily trace collection queries with begin and end times, repeat collections, version control numbers, Guid and logger names, input and output file collections, type:

```
Logman create trace daily_lsass_trace_log -b 7/27/2000 13:00:00 -e 7/27/2000 15:00:00 -r -v mmddhhmm -p "Local Security Authority(LSA) " 0x00000001 -rf 30:00 -o " c:\perflogs\daily_lsass_trace"
```

To create daily counter collection queries with begin and end times, repeat collections, version control numbers, counter paths, sample intervals, SQL formats and output file collections, type:

```
Logman create counter daily_perf_log -b 7/27/2000 13:00:00 -e 7/27/2000 15:00:00 -r -v mmddhhmm -c "\Processor(_Total)\% Processor Time" "\Memory\Available bytes" -si 00:15 -f sql -o perfdb!daily_log
```

To start daily collections with sample intervals, account names and passwords, type:

```
Logman start daily_perf_log -s \\%computer_name% -u admin "adminpassword"
```

To start manual data collections, type:

Logman start daily_perf_log

To stop data collections, type:

Logman stop daily_perf_log

To delete data collections, type:

Logman delete daily_perf_log

To display the status of collection queries, type the following commands:

Logman query

Logman query daily_perf_log

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Start or stop a counter log, trace log, or alert manually

Define start or stop parameters for a log or alert

Performance Logs and Alerts overview

Relog

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Lpq

Displays the status of a print queue on a computer running Line Printer Daemon (LPD). Used without parameters, **lpq** displays command-line help for the **lpq** command.

Syntax

lpq **-S** *ServerName* **-P** *PrinterName* [**-l**]

↑Top of page

Parameters

-S ***ServerName*** **:** Required. Specifies, by name, the computer that hosts the print queue whose status you want to display.

-P ***PrinterName*** **:** Required. Specifies, by name, the printer for the print queue whose status you want to display.

-l : Specifies that you want to display details about the status of the print queue.

/? : Displays help at the command prompt.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Lpr

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Lpr

Sends a file to a computer running Line Printer Daemon (LPD) in preparation for printing. Used without parameters, **lpr** displays command-line help for the **lpr** command.

Syntax

lpr [**-S** *ServerID*] **-P** *PrinterName* [**-C** *BannerContent*] [**-J** *JobName*] [{**-o** | **-o l**}] [**-d**] [**-x**] *FileName*

⇡Top of page

Parameters

-S ***ServerID*** **:** Specifies, by name or IP address, the computer that hosts the printer on which you want to print the file. You do not need to provide this parameter if the printer is attached to the local computer.

-P ***PrinterName*** **:** Required. Specifies, by name, the printer on which you want to print the file.

-C ***BannerContent*** **:** Specifies the content to print on the banner page of the print job. If you do not include this parameter, the name of the computer from which the print job was sent appears on the banner page.

-J ***JobName*** **:** Specifies the print job name that will be printed on the banner page. If you do not include this parameter, the name of the file being printed appears on the banner page.

{-o | -o l} : Specifies the type of file that you want to print. The parameter **-o** specifies that you want to print a text file. The parameter **-o l** specifies that you want to print a binary file (for example, a PostScript file).

-d : Specifies that the data file must be sent before the control file. Use this parameter if your printer requires the data file to be sent first. For more information, see your printer documentation.

-x : Specifies that the **lpr** command must be compatible with the Sun Microsystems operating system referred to as SunOS for releases up to and including 4.1.4_u1.

FileName **:** Required. Specifies, by name, the file to be printed.

/? : Displays help at the command prompt.

Top of page

Remarks

- To find the name of the printer, open the Printers and Faxes folder. To open Printers and Faxes, click **Start**, click **Control Panel**, click **Printers and Other Hardware**, and then click **Printers and Faxes**.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Lpq

Command-line reference A-Z

Top of page

Macfile

Use the **macfile** commands at the command prompt to manage File Server for Macintosh servers, volumes, directories, and files. You can automate administrative tasks by including a series of commands in batch files and starting them manually or at predetermined times.

To modify directories in Macintosh-accessible volumes

Syntax

macfile directory [**/server:**\\ComputerName] **/path:***Directory* [**/owner:***OwnerName*] [**/group:***GroupName*] [**/permissions:***Permissions*]

⇡Top of page

Parameters

/server:\\*ComputerName* : Specifies the server on which to change a directory. If omitted, the operation is performed on the local computer.

/path:*Directory* : Required. Specifies the path to the directory that you want to change. The directory must exist. **Macfile directory** does not create directories.

/owner:*OwnerName* : Changes the owner of the directory. If omitted, the owner remains unchanged.

/group:*GroupName* : Specifies or changes the Macintosh primary group that is associated with the directory. If omitted, the primary group remains unchanged.

/permissions:*Permissions* : Sets permissions on the directory for the owner, primary group, and world (everyone). An 11-digit number is used to set permissions. The number 1 grants permission and 0 revokes permission (for example, 11111011000). The position of the digit determines which permission is set, as described in the following table. If omitted, permissions remain unchanged.

Position	Sets Permission for

First	OwnerSeeFiles
Second	OwnerSeeFolders
Third	OwnerMakeChanges
Fourth	GroupSeeFiles
Fifth	GroupSeeFolders
Sixth	GroupMakeChanges
Seventh	WorldSeeFiles
Eighth	WorldSeeFolders
Ninth	WorldMakeChanges
Tenth	The directory cannot be renamed, moved, or deleted.
Eleventh	The changes apply to the current directory and all subdirectories.

/?: Displays help at the command prompt.

Top of page

Remarks

- If the information that you supply contains spaces or special characters, use quotation marks around the text (for example, "*Computer Name*").
- Use **macfile directory** to make an existing directory in a Macintosh-accessible volume available to Macintosh users. The **macfile directory** command does not create directories. Use File Manager, the command prompt, or the **macintosh new folder** command to create a directory in a Macintosh-accessible volume before you use the **macfile directory** command.

Top of page

Examples

The following example changes the permissions of the subdirectory May sales, in the Macintosh-accessible volume Statistics, on the E drive of the local server. The example assigns See Files, See Folders, and Make Changes permissions to the owner and See Files and See Folders permissions to all other users, while preventing the directory from being renamed, moved, or deleted.

macfile directory /path:"e:\statistics\may sales" /permissions:11111011000

To join a Macintosh file's data and resource forks

⇡Top of page

Syntax

macfile forkize [**/server:**\\ComputerName] [**/creator:***CreatorName*] [**/type:***TypeName*] [**/ datafork:***FilePath*] [**/resourcefork:***FilePath*] **/targetfile:***FilePath*

⇡Top of page

Parameters

/server:\\*ComputerName* : Specifies the server on which to join files. If omitted, the operation is performed on the local computer.

/creator:*CreatorName* : Specifies the creator of the file. The Macintosh Finder uses the **/creator** command-line option to determine the application that created the file.

/type:*TypeName* : Specifies the type of file. The Macintosh Finder uses the **/type** command-line option to determine the file type within the application that created the file.

/datafork:*FilePath* : Specifies the location of the data fork that is to be joined. You can specify a remote path.

/resourcefork:*FilePath* : Specifies the location of the resource fork that is to be joined. You can specify a remote path.

/targetfile:*FilePath* : Required. Specifies the location of the file that is created by joining a data fork and a resource fork, or specifies the location of the file whose type or creator you are changing. The file must be on the specified server.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- If the information that you supply contains spaces or special characters, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Examples

To create the file Treeapp on the Macintosh-accessible volume D:\Release, using the resource fork C:\Cross\Mac\Appcode, and to make this new file appear to Macintosh clients as an application (Macintosh applications use the type APPL) with the creator (signature) set to MAGNOLIA, type:

macfile forkize /resourcefork:c:\cross\mac\appcode /type:APPL /creator:MAGNOLIA /targetfile:D:\Release\Treeapp

To change the file creator to Microsoft Word 5.1, for the file WORD.txt in the directory D:\Word documents\Group files, on the server \\SERVERA, type:

macfile forkize /server:\\servera /creator:MSWD /type:TEXT /targetfile:"d:\Word documents\Group files\Word.txt"

To change the logon message and limit sessions

⇡Top of page

Syntax

macfile server [**/server:**\\ComputerName] [**/maxsessions:**{*Number* | **unlimited**}] [**/loginmessage:** *Message*]

⇡Top of page

Parameters

/server:\\*ComputerName* : Specifies the server on which to change parameters. If omitted, the operation is performed on the local computer.

/maxsessions:{*Number* | unlimited} : Specifies the maximum number of users who can simultaneously use File and Print Servers for Macintosh. If omitted, the **maxsessions** setting for the server remains unchanged.

/loginmessage:*Message* : Changes the message Macintosh users see when logging on to the File Server for Macintosh server. The maximum number of characters for the logon message is 199. If omitted, the **loginmessage** message for the server remains unchanged. To remove an existing logon message, include the **/loginmessage** parameter, but leave the *Message* variable blank

/?: Displays help at the command prompt.

Top of page

Remarks

- If the information that you supply contains spaces or special characters, use quotation marks around the text (for example, "*Computer Name*").

Top of page

Examples

To change the number of File and Print Server for Macintosh sessions that are permitted on the local server from the current setting to five sessions, and to add the logon message "Log off from Server for Macintosh when you are finished.", type the following:

macfile server /maxsessions:5 /loginmessage:"Log off from Server for Macintosh when you are finished."

To add, change, or remove Macintosh-accessible volumes

Top of page

Syntax

macfile volume {**/add** | **/set**} [**/server:**\\ComputerName] **/name:***VolumeName* **/path:***Directory* [**/readonly:**{**true** | **false**}] [**/guestsallowed:**{**true** | **false**}] [**/password:***Password*] [**/maxusers:**{*Number* | **unlimited**}]

macfile volume /remove [**/server:**\\ComputerName] **/name:***VolumeName*

Top of page

Parameters

{/add | /set} : Required when you are adding or changing a Macintosh-accesible volume. Adds or changes the specified volume.

/server:\\*ComputerName* : Specifies the server on which to add, change, or remove a volume. If omitted, the operation is performed on the local computer.

/name:*VolumeName* : Required. Specifies the volume name to be added, changed, or removed.

/path:*Directory* : Required and valid only when you are adding a volume. Specifies the path to the root directory of the volume to be added.

/readonly:{true | false} : Specifies whether users can change files in the volume. Type **true** to specify that users cannot change files in the volume. Type **false** to specify that users can change files in the volume. If omitted when adding a volume, changes to files are allowed. If omitted when changing a volume, the readonly setting for the volume remains unchanged.

/guestsallowed:{true | false} : Specifies whether users who log on as guests can use the volume. Type **true** to specify that guests can use the volume. Type **false** to specify that guests cannot use the volume. If omitted when adding a volume, guests can use the volume. If omitted when changing a volume, the **guestsallowed** setting for the volume remains unchanged.

/password:*Password* : Specifies a password that will be required to access the volume. If omitted when adding a volume, no password is created. If omitted when changing a volume, the password remains unchanged.

/maxusers:{*Number* | unlimited} : Specifies the maximum number of users who can simultaneously use the files on the volume. If omitted when adding a volume, an unlimited number of users can use the volume. If omitted when changing a volume, the **maxusers** value remains unchanged.

/remove : Required when you are removing a Macintosh-accesible volume. Removes the specified volume.

/?: Displays help at the command prompt.

Top of page

Remarks

- If the information that you supply contains spaces or special characters, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Examples

To create a volume called US Marketing Statistics on the local server, using the Stats directory in the E drive, and to specify that the volume cannot be accessed by guests, type:

macfile volume /add /name:"US Marketing Statistics" /guestsallowed:false /path:e:\Stats

To change the volume created above to be read-only and to require a password, and to set the number of maximum users to five, type:

macfile volume /set /name:"US Marketing Statistics" /readonly:true /password:saturn /maxusers:5

To add a volume called Landscape Design, on the server \\Magnolia, using the Trees directory in the E drive, and to specify that the volume can be accessed by guests, type:

macfile volume /add /server:\\Magnolia /name:"Landscape Design" /path:e:\Trees

To remove the volume called Sales Reports on the local server, type:

macfile volume /remove /name:"Sales Reports"

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Mkdir

Creates a directory or subdirectory.

Syntax

mkdir [*Drive***:**]*Path*

md [*Drive***:**]*Path*

Top of page

Parameters

Drive**: :** Specifies the drive on which you want to create the new directory.

***Path* :** Required. Specifies the name and location of the new directory. The maximum length of any single path is determined by the file system.

/? : Displays help at the command prompt.

Top of page

Remarks

- When you enable command extensions (that is, the default), you can use a single **mkdir** command to create intermediate directories in a specified path. For more information about enabling and disabling command extensions, see **cmd** in Related Topics.

Top of page

Examples

To create a directory named Taxes with a subdirectory named Property, which contains a subdirectory named Current, type:

```
mkdir \Taxes\Property\Current
```

This is the same as typing the following sequence of commands with command extensions disabled:

```
mkdir \Taxes
chdir \Taxes
mkdir Property
chdir Property
mkdir Current
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Chdir

Command-line reference A-Z

Top of page

Mmc

Opens Microsoft Management Console (MMC). Using the **mmc** command-line options, you can open a specific MMC console, open MMC in author mode, or specify that the 32-bit or 64-bit version of MMC is opened..

Syntax

mmc *Path\filename.msc* [**/a**] [**/64**] [**/32**]

⇡Top of page

Parameters

Path\filename.msc **:** Starts MMC and opens a saved console. You need to specify the complete path and file name for the saved console file. If you do not specify a console file, MMC opens a new console.

/a : Opens a saved console in author mode. Used to make changes to saved consoles.

/64 : Opens the 64-bit version of MMC (MMC64). Use this option only if you are running Windows XP 64-Bit Edition.

/32 : Opens the 32-bit version of MMC (MMC32). When running Windows XP 64-Bit Edition, you can run 32-bit snap-ins by opening MMC with this command-line option.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Using the *Path\filename.msc* command-line option

  You can use environment variables to create command lines or shortcuts that do not depend on the explicit location of console files. For instance, if the path to a console file is in the system folder (for example, **mmc c:\winnt\system32\console_name.msc**), you can use the expandable data string %systemroot% to specify the location (**mmc %systemroot%\system32 \console_name.msc**). This can be useful when you need to delegate tasks to people in your organization who are working on different computers.
- Using the **/a** command-line option

  When consoles are opened with this option, they are opened in author mode, regardless of their default mode. This does not permanently change the default mode setting for files. When you omit this option, MMC opens console files according to their default mode settings.
- After you open MMC or a console file in author mode, you can open any existing console by clicking **Open** on the **Console** menu.
- You can use the command line to create shortcuts for opening MMC and saved consoles. A command-line command works with the **Run** command on the **Start** menu, in any command-prompt window, in shortcuts, or in any batch file or program that calls the command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Set

Dir

Path

Top of page

Manage Your Profile |Legal |Contact Us

Mode

Displays system status, changes system settings, or reconfigures ports or devices. Used without parameters, **mode** displays all the controllable attributes of the console and the available COM devices. Because you can use **mode** to perform many different tasks, the syntax you need to use to carry out each task is different. Click the task that you want to perform.

To configure a serial communications port

Syntax

mode com*m*[**:**] [**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**}]

Top of page

Parameters

com*m*[:] : Specifies the number of the asynchronous-communications (COM) port.

baud=*b* : Specifies the transmission rate in bits per second. The following table lists valid abbreviations for *b* and its related rate.

Value	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 following table lists valid *p* values. The default value is **e**. Not all computers support the values **m** and **s**.

Value	Description
n	none
e	even
o	odd
m	mark
s	space

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|hs} : Specifies whether the Data Terminal Ready (DTR) circuit is on or off, or set to handshake.

rts={on|off|hs|tg} : Specifies whether the Request To Send (RTS) circuit is set to on, off, handshake, or toggle.

idsr={on|off} : Specifies whether the DSR circuit sensitivity is on or off.

/?: Displays help at the command prompt.

To display the status of all devices or of a single device

⇡Top of page

Syntax

mode [*device*] [**/status**]

⇡Top of page

Parameters

***device* :** Specifies the name of the device for which you want to display the status.

/status : Requests the status of any redirected parallel printers. You can abbreviate the **/status** command-line option as **/sta**.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Used without parameters, **mode** displays the status of all devices installed on your system.

To redirect output from a parallel port to a serial communications port

⇡Top of page

Syntax

mode lpt*n*[**:**]=**com***m*[**:**]

⇡Top of page

Parameters

lpt*n* : Required. Specifies the parallel port. Valid values for *n* are in the range 1 through 3.

com*m* [:] : Required. Specifies the serial port. Valid values for *m* are in the range 1 through 4.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- You must be a member of the Administrators group to redirect printing.

⇡Top of page

Examples

To set up your system so that it sends parallel-printer output to a serial printer, you must use the **mode** command twice. The first time, use **mode** to configure the serial port. The second time, 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), type:

```
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, type the following command before you print the file:

```
mode lpt1
```

This command prevents the redirection the file from LPT1 to COM1.

To select, refresh, or display the numbers of the code pages for the console

Top of page

Syntax

mode *device* **codepage select=***yyy*

mode *device* **codepage** [**/status**]

Top of page

Parameters

device **:** Required. Specifies the device for which you want to select a code page. CON is the only valid name for a device.

codepage select= : Required. Specifies which code page to use with the specified device. You can abbreviate **codepage** and **select** as **cp** and **sel**, respectively.

yyy **:** Required. Specifies the number of the code page to select. The following table lists each code page supported and its Country/Region or language.

Value	Country/Region
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 : Required. 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 **/status** to **/sta**. Whether or not you specify **/status**, **mode codepage** displays the numbers of the code pages that are selected for the specified device.

/?: Displays help at the command prompt.

To change the size of the command prompt screen buffer

⇡Top of page

Syntax

mode con[**:**] [**cols=***c*] [**lines=***n*]

⇡Top of page

Parameters

con[:] : Required. Indicates that the 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.

/?: Displays help at the command prompt.

To set the keyboard typematic rate

⇡Top of page

Syntax

mode con[**:**] [**rate=***r* **delay=***d*]

Top of page

Parameters

con[:] : Required. Refers to the keyboard.

rate=*r* : Specifies the rate at which a character is repeated on the screen when you hold down a key.

delay=*d* : Specifies the amount of time that must elapse, after you press and hold down a key, before the character output starts to repeat.

/?: Displays help at the command prompt.

Top of page

Remarks

- The typematic rate is the rate at which a character repeats 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.
- Using **rate=***r*

  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.
- Using **delay=***d*

  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.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Mountvol

Creates, deletes, or lists a volume mount point. **Mountvol** is a way to link volumes without requiring a drive letter.

Syntax

mountvol [*Drive*:]*Path VolumeName*

mountvol [*Drive*:]*Path* **/d**

mountvol [*Drive*:]*Path* **/L**

mountvol *Drive*: **/s**

⇡Top of page

Parameters

[*Drive*:]*Path* : Specifies the existing NTFS directory folder where the mount point will reside.

VolumeName* :** Specifies the volume name that is the target of the mount point. The volume name is of the form **\\?\Volume{GUID***}**\, where {*GUID*} is a globally unique identifier (GUID) (for example, \\?\Volume\{2eca078d-5cbc-43d3-aff8-7e8511f60d0e}\).

/d : Removes the volume mount point from the specified folder.

/L : Lists the mounted volume name for the specified folder.

/s : Itanium-based computers only. Mounts the EFI System Partition on the specified drive.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- If you are running out of drive letters to use, mount your local volumes with no drive letters.
- If you need to expand your volume space without reformatting or replacing a hard drive, you can add a mount path to another volume.
- The benefit is that if you use one volume with several mount paths, you can access all local volumes using a single drive letter (such as C:). You need not remember which volume corresponds to which drive letter, although you can mount local volumes and still have them assigned to drive letters.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Dir

Net use

Command-line reference A-Z

Top of page

Move

Moves one or more files from one directory to the specified directory.

Syntax

move [{**/y**|**/-y**}] [*Source*] [*target*]

⇡Top of page

Parameters

/y : Suppresses prompting to confirm you want to overwrite an existing destination file.

/-y : Causes prompting to confirm you want to overwrite an existing destination file.

***Source* :** Specifies the path and name of the file or files to move. If you want to move or rename a directory, *Source* should be the current directory path and name.

***target* :** Specifies the path and name to move files to. If you want to move or rename a directory, *target* should be the desired directory path and name.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•Using the **/y** command-line option

The **/y** command-line option might be preset in the COPYCMD environment variable. You can override this with **/-y** on the command line. The default is to prompt on overwrites unless the **copy** command is being executed from within a batch script.

•Moving encrypted files

Moving encrypted files to a volume that does not support Encrypting File System (EFS) results in an error. Decrypt the files first or move the files to a volume that does support EFS.

⇡Top of page

Examples

To move all files with the .xls extension from the \Data directory to the \Second_Q\Reports directory, type the following:

move \data*.xls \second_q\reports\

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Copy

Move

Command-line reference A-Z

MS-DOS subsystem commands

Windows XP includes 16-bit commands (nonnative) for the MS-DOS subsystem and other subsystems. These include 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 XP or the MS-DOS subsystem. The commands are accepted to preserve compatibility with existing files, but they have no effect at the command line because the functionality is automatic.

Note

- The following 16-bit MS-DOS subsystem commands are not available on Windows XP 64-Bit Edition.

For more information, click a command:

- Append
- Debug
- Debug subcommands
- Edit
- Edlin
- Edlin subcommands
- Exe2bin
- Fastopen
- Forcedos
- Graphics
- Loadfix
- Loadhigh (lh)
- Mem
- Nlsfunc
- Setver
- Share

To configure the MS-DOS subsystem, use configuration commands, such as **device** or **lastdrive**. Place these commands in the Config.nt file in the *systemroot*\System32 directory or in the Config file specified by an application's program information file (PIF). These commands affect the MS-DOS subsystem only. The MS-DOS subsystem ignores many of these commands, such as **buffers** and **break**, because the MS-DOS subsystem works without them. They are accepted for compatibility only.

For more information about MS-DOS subsystem configuration commands, click a command:

Top of page

Related Topics

- Buffers
- Country
- Device
- Devicehigh
- Dos
- Dosonly
- Driveparm
- Echoconfig
- Fcbs
- Files
- Install
- Lastdrive
- Ntcmdprompt

Shell

Stacks

Switches

↑Top of page

Manage Your Profile |Legal |Contact Us

Append

Enables programs to open data files in specified folders as if these files were in the current folder. Used without parameters, **append** displays the appended directory list.

Syntax

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

⇡Top of page

Parameters

; : Cancels the list of appended folders.

[*Drive*:]*Path* : Specifies the drive and folder that you want to append to the current folder. If you do not specify a drive, the current drive is the default. You can specify multiple entries of [*Drive*:] *Path* by separating the entries with semicolons.

/x:{on|off} : Specifies whether you want the MS-DOS subsystem to search appended folders when it executes programs. **/x:on** searches appended folders. **/x:off** does not search appended folders.

/path:{on|off} : Specifies whether you want a program to search appended folders for a data file when a path is already included with the name of the file for which the program is looking. The default setting is **/path:on**.

/e : Assigns the list of appended folders to an environment variable named APPEND. This command-line option works only the first time you use **append** after you start your system.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- Storing the appended folders list

  You can use the **/e** command-line option with **append** to assign the appended folders list to an environment variable named APPEND. To do this, first use the **append** command with only the **/e** command-line option. Then, use **append** again and include the folders you want to append. You cannot specify **/e** and [*Drive***:**]*Path* on the same command line.
- Specifying multiple appended folders

  To append more than one folder, separate multiple entries with semicolons. If you use **append** with the [*Drive***:**]*Path* parameters again, the specified folder or folders replace any folders specified in a previous **append**.
- Using **dir**

  When you type **dir** to see a list of a directory's files and subdirectories, **dir** does not include the file names from appended folders.
- Resolving file name conflicts

  If a file in an appended folder has the same name as a file in the current folder, programs open the file in the current folder.
- Using **append** with programs that create new files

  When a program opens a file in an appended folder, the file can be found as if it is in the current folder. If the program saves the file by creating a new file with the same name, the new file is created in the current folder, not the appended folder. You can use **append** 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 usually save modified data files by making new copies. To avoid confusion, avoid using **append** with these programs.
- Using **/x:on** with **path**

  When you use **/x:on**, you can run a program located in an appended folder by typing the program name at the command prompt. Usually, you use the **path** command to specify folders that contain programs, however, you do not need to use the **path** command to specify an appended folder that contains programs. MS-DOS subsystem finds a program in an appended folder by following the usual order in which it searches for a program: first in the current folder, then in the appended folders, and then in the search path.
- Abbreviating **/x:on** to **/x**

  You can abbreviate **/x:on** to **/x**. To do this, specify **/x:on** the first time you use **append** after you start your system. After that, you can switch between **/x:on** and **/x:off**.

- Using **/e** with **set**

  You can use **/e** with the **set** command to display a list of the appended folders. For information about environment variables and **set**, see Related Topics.

Top of page

Examples

To allow programs to open data files in B:\Letters and A:\Reports as if the files are in the current folder, type:

append b:\letters;a:\reports

To append the same folders and keep a copy appended folders list in the Windows XP environment (this must be the first time you use **append** after you start your system), type:

append /e

append b:\letters;a:\reports

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Set

Dir

Path

Top of page

Manage Your Profile |Legal |Contact Us

Debug

Starts Debug.exe, a program you can use to test and debug MS-DOS executable files. Used without parameters, **debug** starts Debug.exe and the **debug** prompt appears, which is represented by a hyphen (-).

Syntax

debug [[*Drive***:**][*Path*] *FileName* [*parameters*]]

↑Top of page

Parameters

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the executable file that you want to test.

***parameters* :** Specifies any command-line information required by the executable file that you want to test.

?: Displays a list of **debug** subcommands.

↑Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- Using MS-DOS subsystem commands

Debug is a MS-DOS subsystem command that runs under WOW/NTVDM.

- Separating command parameters

  You can separate parameters with a comma or white space, but these separators are required only between two hexadecimal values. For example, the following commands are equivalent:

  dcs:100 110

  d cs:100 110

  d,cs:100,110
- Using **debug** subcommands

  There are several **debug** subcommands that you can use. For more information about **debug** subcommands, see Related Topics.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Debug subcommands

↑Top of page

Debug subcommands

For more information, click a command:

- Debug: a (assemble)
- Debug: c (compare)
- Debug: d (dump)
- Debug: e (enter)
- Debug: f (fill)
- Debug: g (go)
- Debug: h (hexadecimal)
- Debug: i (input)
- Debug: l (load)
- Debug: m (move)
- Debug: n (name)
- Debug: o (output)
- Debug: p (proceed)
- Debug: q (quit)
- Debug: r (register)
- Debug: s (search)
- Debug: t (trace)
- Debug: u (unassemble)
- Debug: w (write)
- Debug: xa (allocate expanded memory)
- Debug: xd (deallocate expanded memory)
- Debug: xm (map expanded memory pages)
- Debug: xs (display expanded memory status)

Top of page

Manage Your Profile |Legal |Contact Us

Edit

Starts MS-DOS Editor, which creates and changes ASCII text files. Used without parameters, **edit** starts MS-DOS Editor.

Syntax

edit [[*Drive*:][*Path*] *FileName*] [**/b**] [**/g**] [**/h**] [**/nohi**]

⇡Top of page

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. Use this command-line option if a monochrome monitor does not display MS-DOS Editor correctly.

/g : Uses the fastest screen updating possible for a CGA monitor.

/h : Displays the maximum number of lines possible for the current monitor.

/nohi : Enables you to use eight-color monitors with MS-DOS Editor. Typically, Windows XP uses 16 colors.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

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

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

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command shell overview

Top of page

Manage Your Profile |Legal |Contact Us

Edlin

Starts Edlin.exe, a line-oriented text editor that you can use to create and change ASCII files.

Syntax

edlin [*Drive*:][*Path*] *FileName* [**/b**]

⇡Top of page

Parameters

[*Drive*:][*Path*] *FileName* : Specifies the location and name of an ASCII file on a disk. *Filename* is required. If the file exists, Edlin.exe opens it. If the file does not exist, Edlin.exe creates a file in memory and uses the specified location and file name to create the file on a disk when you use the **e** subcommand.

/b : Specifies that Edlin.exe is to ignore the end-of-file character (CTRL+Z).

?: Displays a list of **edlin** subcommands.

⇡Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- This tool is not available on Windows XP 64-Bit Edition.
- Using Edlin.exe

  Edlin.exe numbers each line of the text file that is located in memory. You can use Edlin.exe to insert, modify, copy, move, and delete lines of the file. If you want to use a full-screen editor, use the **edit** command.
- Maximum line length

  Edlin.exe accepts a maximum of 253 characters per line.

- Using **edlin** subcommands

  There are several **edlin** subcommands that you can use. For more information about **edlin** subcommands, see Related Topics.
- Understanding the asterisk character

  When an asterisk appears as the only character on the line, it is the **edlin** * prompt, after which you type **edlin** subcommands. When an asterisk appears after a line number on a line, it indicates that it is the line where the insertion point is located.
- Displaying a page of text

  A page of text is one full screen of information. With a 25-line screen mode, Edlin.exe displays 24 lines of text per page. The number of lines per page depends on the screen mode you use.
- Starting and stopping insert mode

  To insert lines into the file in memory, use the **i** (insert) subcommand. After you have finished inserting lines, press ENTER and then CTRL+C to stop insert mode. For more information about inserting lines by using the **i** (insert) subcommand, see Related Topics.
- Editing keys

  Several editing keys are provided that you can use to edit the file in memory.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Edit

Edlin subcommands

For more information, click a command:

- Edlin: [line]
- Edlin: a (append)
- Edlin: c (copy)
- Edlin: d (delete)
- Edlin: e (end)
- Edlin: i (insert)
- Edlin: l (list)
- Edlin: m (move)
- Edlin: p (page)
- Edlin: q (quit)
- Edlin: r (replace)
- Edlin: s (search)
- Edlin: t (transfer)
- Edlin: w (write)

Top of page

Manage Your Profile |Legal |Contact Us

Exe2bin

Converts executable (.exe) files to binary format.

Syntax

exe2bin [*drive1***:**][*path1*]*InputFile* [[*drive2***:**][*path2*]*OutputFile*]

Top of page

Parameters

[*drive1*:][*path1*]*InputFile* : Specifies the location and name of the input file that you want to convert. *InputFile* is required.

[*drive2*:][*path2*]*OutputFile* : Specifies the location and name of the output file that you want to create.

/?: Displays help at the command prompt.

Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- This tool is not available on Windows XP 64-Bit Edition.
- **Exe2bin** is included as a courtesy to software developers. It is not useful for general users.
- The following restrictions apply when you use **exe2bin**:

  - 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 64KB.
  - There must be no **stack** segment.

- **Exe2bin** performs the following specific actions, depending on the values you use for *InputFile* and *OutputFile* parameters:

  - The default file extension for *InputFile* is .exe. **Exe2bin** converts the *InputFile* .exe to a file in .bin format (that is, a memory image of the program) and uses the location and file name you specify for [*drive2***:**][*path2*]*OutputFile* to store the 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 a file name for *OutputFile*, **exe2bin** uses the *InputFile* file name.
  - The default extension for the file name specified for *OutputFile* is .bin.
- The following conversion types are possible:

  - If you do not specify the CS:IP in the .exe file, **exe2bin** performs a pure binary conversion. If the program contains instructions requiring segment relocation, **exe2bin** prompts you for the fixup value. The fixup value is the absolute segment at which the program is to be loaded. You can use the resulting program only when you load it at the absolute memory address specified by your program. The command interpreter cannot load the program.
  - If you specify the CS:IP 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 *OutputFile*. No segment fixups are allowed, because .com files must be able to relocate segments. That is, .com files must assume the entry conditions explained in Microsoft Macro Assembler documentation. The command interpreter can then load and run the program in the same way that it loads and runs the .com programs included on your Windows XP disk.

⇧Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Fastopen

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

Note

- This tool is not available on Windows XP 64-Bit Edition.

Command-line reference A-Z

↑Top of page

Manage Your Profile |Legal |Contact Us

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 XP.

Syntax

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

⇡Top of page

Parameters

/d *directory* : Specifies the current directory for the specified program to use.

***FileName* :** Required. Specifies the program to start. If not in the current directory or path, you must specify the drive letter or directories to the program.

***parameters* :** Specifies parameters to pass to the program.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- This tool is not available on Windows XP 64-Bit Edition.
- Some MS-DOS programs are not recognized by Windows XP as MS-DOS programs. Use **forcedos** to start these programs.
- All programs that you start with **forcedos** are opened as MS-DOS applications.

⇡Top of page

Examples

To start the Myprog program in the \Oldapps directory so that you can use the \Reports directory, type:

forcedos /d reports \oldapps\myprog

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Graphics

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

Top of page

Manage Your Profile |Legal |Contact Us

Loadfix

Ensures that a program is loaded above the first 64KB of conventional memory, and then runs the program.

Syntax

loadfix [*Drive*:][*Path*] *FileName*

↑Top of page

Parameters

[*Drive*:][*Path*] : Specifies the drive and directory of the program.

***FileName* :** Specifies the name of the program.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- To use **loadfix**, include it at the beginning of the command that starts the program.
- When all or a portion of the program has been loaded in the first 64KB of conventional memory and it cannot run successfully, some programs display the following message:

  ```
  Packed file corrupt
  ```

  This error typically occurs when you load device drivers into the upper memory area (UMA), thereby freeing more of the first 64KB of conventional memory. If this message appears, use the **loadfix** command to ensure that a program is loaded above the first 64KB of virtual conventional memory.

Top of page

Examples

To load a program named Myapp.exe (in the Apps directory of drive C) and use the **/c** command-line option (for example, to load the program in character mode), type:

loadfix c:\apps\myapp.exe /c

You can also include this command in a batch program.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Loadhigh

Loads a program into the upper memory area (UMA). Loading a program into the UMA leaves more room in conventional memory for other programs. Use the *systemroot*\System32\Config.nt file, or the equivalent startup file specified in a program's program information file (PIF), to specify the programs to load high.

Syntax

loadhigh [*Drive***:**][*Path*] *FileName* [*parameters*]

⇡Top of page

Parameters

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the program you want to load. *Filename* is required.

***parameters* :** Specifies any command line information required by the program.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- 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 **dos** in Related Topics.
- 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.

- If you use the **loadhigh** command to load a program, Windows XP attempts to load it into the UMA. If there is insufficient space in the UMA, the program is loaded into conventional memory; the memory area used is not indicated.
- The most convenient way to use the **loadhigh** command is to include it in your Autoexec.nt or equivalent startup file.

Top of page

Examples

To load a driver into the high memory area (HMA), include the following line in your Autoexec.nt or equivalent startup file:

lh %systemroot%\system32\dosx

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Dos

Devicehigh

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Mem

Displays information about allocated memory areas, free memory areas, and programs that are currently loaded into memory in the MS-DOS subsystem. Used without parameters, **mem** displays the status of the MS-DOS subsystem's used and free memory.

Syntax

mem [{**/program** | **/debug** | **/classify**}]

⇡Top of page

Parameters

/program : Displays the status of programs that are currently loaded into memory.

/debug : Displays the status of currently loaded programs and internal drivers, and other programming information.

/classify : Displays the status of programs loaded into conventional memory and the upper memory area (UMA).

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- This tool is not available on Windows XP 64-Bit Edition.
- Using **/program**

  You cannot use **/program** with **/debug** or **/classify**. You can abbreviate **/program** to **/p**.

- Using **/debug**

  You cannot use **/debug** with **/program** or **/classify**. You can abbreviate **/debug** to **/d**.
- Using **/classify**

  /classify 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 **/classify** with **/program** or **/debug**. You can abbreviate **/classify** to **/c**.
- Displaying memory status

  The MS-DOS subsystem displays the status of extended memory only if you have allocated memory above 1 MB.

Top of page

Examples

Suppose your MS-DOS subsystem is configured with expanded memory. To display the status of your system's total memory (that is, conventional, expanded, extended) and a list of programs currently loaded into memory, type:

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 640KB.
- "Available to MS-DOS" is the amount of conventional memory allocated, including the memory needed 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" (not shown in the preceding example) is the amount of expanded memory available for programs.
- "Total contiguous extended memory" is the amount of memory beyond 1 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.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Nlsfunc

Loads information specific to the country/region. Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.

Syntax

nlsfunc [[*Drive*:][*Path*][*FileName*]]

⇡Top of page

Parameters

[[*Drive*:][*Path*][*FileName*]] : Specifies the file containing information specific to the country/region.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- This tool is not available on Windows XP 64-Bit Edition.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Setver

Sets the MS-DOS version number that the MS-DOS subsystem reports to a program. Used without parameters, **setver** displays the current version table.

Syntax

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*]

⇡Top of page

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 also use **/d**.

/quiet : Hides the message that is usually displayed when you delete an entry from the version table.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- This tool is not available on Windows XP 64-Bit Edition.
- Using the version table

  Many programs designed to run with a previous version of MS-DOS run correctly with Windows XP. 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 should run correctly, however, **setver** does not solve the problem if the program is not compatible with Windows XP.
- Loading the version table into memory

  Before you can use **setver**, the version table must be loaded into memory by inserting 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 file name that is already in the version table, the new entry replaces the existing entry.
- Understanding **setver** exit codes

  To process exit codes returned by **setver**, use the *errorlevel* parameter on the **if** command line in a batch program. For an example of a batch program that processes exit codes using **if**, see Related Topics. The following table lists each exit code and a description.

Exit code	Description
0	**Setver** successfully completed its task.
1	The user specified an invalid command-line option.
2	The user specified an invalid file name.
3	There is insufficient system memory to carry out the command.

4	The user specified an invalid version number format.
5	**Setver** cannot find the specified entry in the version table.
6	**Setver** cannot find the Setver.exe file.
7	The user specified an invalid drive.
8	The user specified too many command-line options.
9	**Setver** detected missing command-line options.
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.

Top of page

Examples

To run Myprog.exe, a file that runs with MS-DOS version 3.30, create an entry in the version table that enables Myprog.exe to interpret the MS-DOS subsystem as version 3.30 by typing:

setver myprog.exe 3.30

To delete the Myprog.exe entry from the version table, without otherwise affecting the Myprog.exe file, type:

setver myprog.exe /delete

To list the contents of the version table on drive C, type:

setver c:

Top of page

Formatting legend

<table>
<tr><th>Format</th><th>Meaning</th></tr>
<tr><td><i>Italic</i></td><td>Information that the user must supply</td></tr>
<tr><td><b>Bold</b></td><td>Elements that the user must type exactly as shown</td></tr>
<tr><td>Ellipsis (...)</td><td>Parameter that can be repeated several times in a command line</td></tr>
<tr><td>Between brackets ([])</td><td>Optional items</td></tr>
<tr><td>Between braces ({}); choices separated by pipe (|). Example: {even|odd}</td><td>Set of choices from which the user must choose only one</td></tr>
<tr><td>Courier font</td><td>Code or program output</td></tr>
</table>

If

Device

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Share

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

Note

- This tool is not available on Windows XP 64-Bit Edition.

Top of page

Related Topics

- Command-line reference A-Z

Top of page

Msiexec

Provides the means to install, modify, and perform operations on Windows Installer from the command line.

To install or configure a product

Syntax

msiexec /i {*package*|*ProductCode*}

⇡Top of page

Parameters

/i : Installs or configures a product.

***package* :** Specifies the name of the Windows Installer package file.

***ProductCode* :** Specifies the globally unique identifier (GUID) of the Windows Installer package.

⇡Top of page

Remarks

- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

⇡Top of page

Examples

To install a product from A:\Example.msi, type:

msiexec /i A:\Example.msi

To use the administrative installation option

⇡Top of page

Syntax

msiexec /a *package*

⇡Top of page

Parameters

/a : Applies the administrative installation option.

***package* :** The name of the Windows Installer package file.

⇡Top of page

Remarks

- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

To repair a product

⇡Top of page

Syntax

msiexec /f [**p**][**o**][**e**][**d**][**c**][**a**][**u**][**m**][**s**][**v**]{*package*|*ProductCode*}

⇡Top of page

Parameters

/f : Enables one or more of the command-line options listed in the following table.

Command	Description
p	Reinstalls only if file is missing.
o	Reinstalls if file is missing or if an older version is installed.
e	Reinstalls if file is missing or an equal or older version is installed.
d	Reinstalls if file is missing or a different version is installed.
c	Reinstalls if file is missing or the stored checksum does not match the calculated value.
a	Forces all files to be reinstalled.
u	Rewrite all required user-specific registry entries.
m	Rewrites all required computer-specific registry entries.
s	Overwrites all existing shortcuts.
v	Runs from source and re-caches the local package.

***package* :** Name of the Windows Installer package file.

***ProductCode* :** Globally unique identifier (GUID) of the Windows Installer package.

Top of page

Remarks

- This option ignores any property values that you type at the command line.
- The default parameter for this command-line option is **/fpecms**.
- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

Top of page

Examples

To repair the installation package, type:

msiexec /fpecms Example.msi

To uninstall a product

⇡Top of page

Syntax

msiexec /x {*package*|*ProductCode*}

⇡Top of page

Parameters

/x : uninstalls a product.

***package* :** Name of the Windows Installer package file.

***ProductCode* :** Globally unique identifier (GUID) of the Windows Installer package.

⇡Top of page

Remarks

- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

⇡Top of page

Examples

To remove or uninstall a package, type:

msiexec /x Example.msi

To advertise a product

⇡Top of page

Syntax

msiexec /j [{**u**|**m**}] *package*

msiexec {**u**|**m**} *package* **/t** *TransformList*

msiexec {**u**|**m**} *package* **/g** *LanguageID*

⇡Top of page

Parameters

/j : Advertises a product.

u : Advertises to the current user.

m : Advertises to all users of the computer.

***package* :** Specifies the Windows Installer package file.

/g *LanguageID* : Identifies the language.

/t *TransformList* : Applies transform to advertised package.

⇡Top of page

Remarks

- This option ignores any property values that you type at the command line.
- If you want to install the application with elevated privileges, use **/jm**.
- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

⇡Top of page

Examples

To advertise a package to all users of this computer, type:

msiexec /jm Example.msi

To advertise a package to all users of this computer, type:

msiexec /jm Example.msi

To set logging level

Top of page

Syntax

msiexec /L [**i**][**w**][**e**][**a**][**r**][**u**][**c**][**m**][**p**][**v**][**+**][**!**]*LogFile***.txt**

Top of page

Parameters

/L : Specifies the path to the log file.

i : Logs status messages.

w : Logs nonfatal warnings.

e : Logs all error messages.

a : Logs startup of actions.

r : Logs action-specific records.

u : Logs user requests.

c : Logs initial user interface parameters.

m : Logs out-of-memory.

p : Logs terminal properties.

v : Logs verbose output. To use **v**, specify **/L*v**.

+ : Appends to existing file.

! : Flushes each line to the log.

*** :** Logs all information except for the **v** option. This is a wildcard.

***LogFile*.txt :** Name and path of the text log file.

⇡Top of page

Remarks

- Windows Installer command-line options are not case-sensitive.
- To include the **v** option in a log file using the wildcard flag, type **/L*v** at the command prompt.
- The Windows Installer log file options can also be used with the uninstall and repair processes.
- For more information about the Windows Installer command-line options, see Related Topics.

⇡Top of page

Examples

To install a package and create a log file that contains the information related to the status, out-of-memory, and error messages, type:

msiexec /i Example.msi /Lime logfile.txt

To apply a patch

⇡Top of page

Syntax

msiexec /p *PatchPackage*

⇡Top of page

Parameters

/p : Applies a patch.

PatchPackage **:** Specific patch.

⇡Top of page

Remarks

- To apply a patch to an administrative installation package, use the following syntax:

  msiexec /p *PatchPackage* **/a** *Example.msi*
- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

To install a transform using the command line

⇡Top of page

Syntax

msiexec /i *package***TRANSFORMS**=*TransformList*

⇡Top of page

Parameters

/i : Installs or configures a product.

package **:** Specifies the Windows Installer package file.

TRANSFORMS= : Property that is used to specify what transform (.mst) files should be applied to the package.

TransformList **:** List of paths separated by semicolons.

⇡Top of page

Remarks

- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

To advertise a product using a transform with the command line

Top of page

Syntax

msiexec /j[u][m] *package* **/t** *TransformList*

Top of page

Parameters

/j : Advertises a product. This option ignores any property values entered on the command line.

u : Advertises to the current user.

m : Advertises to all users of this computer.

***package* :** Name of the Windows Installer package file.

/t : Applies transform to advertised package.

***TransformList* :** List of paths separated by semicolons.

Top of page

Remarks

- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

To set the user interface level

Top of page

Syntax

msiexec /q{n|b|r|f|n+|b+|b-}

Top of page

Parameters

/qn : Displays no user interface.

/qb : Displays a basic user interface.

/qr : Displays a reduced user interface with a modal dialog box displayed at the end of the installation.

/qf : Displays the full user interface with a modal dialog box displayed at the end.

/qn+ : Displays no user interface, except for a modal dialog box displayed at the end.

/qb+ : Displays a basic user interface with a modal dialog box displayed at the end.

/qb- : Displays a basic user interface with no modal dialog boxes.

Top of page

Remarks

- **/qb+-** is not a supported user interface level. The modal box is not displayed if the user cancels the installation.
- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

Top of page

Examples

To display the basic user interface options during the package installation of Example.msi, type:

msiexec /qb Example.msi

To display copyright information for the Windows Installer

Top of page

Syntax

msiexec {**/?**|**/h**}

Top of page

Parameters

{/?|/h} : Displays the Windows Installer version and copyright information.

Top of page

Remarks

- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

To call the system API DllRegisterServer to self-register modules passed on the command line

Top of page

Syntax

msiexec /y *module*

Top of page

Parameters

/y : Calls the system API DllRegisterServer to self-register modules passed on the command line.

module **:** Specifies the file name of module.

⇡Top of page

Remarks

- This option is used only for registry information that cannot be added using the registry tables of the .msi file.
- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

⇡Top of page

Examples

The following example shows how you can use the **msiexec /y** command:

msiexec /y my_file.dll

To call the system API DllUnRegisterServer to unregister modules passed on the command line

⇡Top of page

Syntax

msiexec /z *module*

⇡Top of page

Parameters

/z : Calls the system API DllUnRegisterServer to unregister modules passed on the command line.

module **:** File name of module.

⇡Top of page

Remarks

- This option is used only for registry information that cannot be added using the registry tables of the .msi file.
- Windows Installer command-line options are not case-sensitive.
- For more information about the Windows Installer command-line options, see Related Topics.

↑Top of page

Examples

The following example shows how you can use the **msiexec /z** command:

msiexec /z my_file.dll

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Install a package using the command line

Install a transform using the command line

Install and advertise a product with a transform using the command line

Create an administrative installation package from the command line

Repair a package using the command line

Remove a package using the command line

Advertise a package using the command line

Create an installation log file using the command line

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Msinfo32

Displays a comprehensive view of your hardware, system components, and software environment.

Syntax

msinfo32 [**/?**] [**/pch**] [**/nfo** *FileName*] [**/report** *FileName*] [**/computer** *ComputerName*] [**/showcategories**] [**/category** *categoryID*] [**/categories** *categoryID*]

Top of page

Parameters

***FileName* :** Specifies the file to be opened. This can be an .nfo, .xml, .txt, or .cab file.

/? : Displays help for the msinfo32 command.

/pch : Displays the history view.

/nfo *FileName* : Saves the exported file as an .nfo file.

/report *FileName* : Saves the exported file as a .txt file.

/computer *ComputerName* : Starts System Information for the specified remote computer.

/showcategories : Starts System Information with all available category IDs displayed.

/category *categoryID* : Starts System Information with the specified category selected. Use **/showcategories** to display a list of available category IDs.

/categories *categoryID* : Starts System Information with only the specified category or categories displayed. It also limits the output to the selected category or categories. Use **/showcategories** to display a list of available category IDs.

/? : Displays help at the command prompt.

Top of page

Remarks

- Some System Information categories contain large amounts of data. You can use the **start /wait** command to optimize reporting performance for these categories. For more information, see Related Topics.

Top of page

Examples

To list the available category IDs, type:

```
msinfo32 /showcategories
```

To start System Information with all available information displayed, except Loaded Modules, type:

```
msinfo32 /categories +all -loadedmodules
```

To display only System Summary information and create an .nfo file called syssum.nfo that contains information in the System Summary category, type:

```
msinfo32 /nfo syssum.nfo /categories +systemsummary
```

To display resource conflict information and create an .nfo file called conflicts.nfo that contains information about resource conflicts, type:

```
msinfo32 /nfo conflicts.nfo /categories +componentsproblemdevices+resourcesconflicts +resourcesforcedhardware
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply

Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

System Information

Start

Top of page

Manage Your Profile |Legal |Contact Us

Nbtstat

Displays NetBIOS over TCP/IP (NetBT) protocol statistics, NetBIOS name tables for both the local computer and remote computers, and the NetBIOS name cache. **Nbtstat** allows a refresh of the NetBIOS name cache and the names registered with Windows Internet Name Service (WINS). Used without parameters, **nbtstat** displays help.

Syntax

nbtstat [**-a** *RemoteName*] [**-A** *IPAddress*] [**-c**] [**-n**] [**-r**] [**-R**] [**-RR**] [**-s**] [**-S**] [*Interval*]

⇡Top of page

Parameters

-a *RemoteName* : Displays the NetBIOS name table of a remote computer, where *RemoteName* is the NetBIOS computer name of the remote computer. The NetBIOS name table is the list of NetBIOS names that corresponds to NetBIOS applications running on that computer.

-A *IPAddress* : Displays the NetBIOS name table of a remote computer, specified by the IP address (in dotted decimal notation) of the remote computer.

-c : Displays the contents of the NetBIOS name cache, the table of NetBIOS names and their resolved IP addresses.

-n : Displays the NetBIOS name table of the local computer. The status of **Registered** indicates that the name is registered either by broadcast or with a WINS server.

-r : Displays NetBIOS name resolution statistics. On a Windows XP computer that is configured to use WINS, this parameter returns the number of names that have been resolved and registered using broadcast and WINS.

-R : Purges the contents of the NetBIOS name cache and then reloads the #PRE-tagged entries from the Lmhosts file.

-RR : Releases and then refreshes NetBIOS names for the local computer that is registered with

WINS servers.

-s : Displays NetBIOS client and server sessions, attempting to convert the destination IP address to a name.

-S : Displays NetBIOS client and server sessions, listing the remote computers by destination IP address only.

***Interval* :** Redisplays selected statistics, pausing the number of seconds specified in *Interval* between each display. Press CTRL+C to stop redisplaying statistics. If this parameter is omitted, **nbtstat** prints the current configuration information only once.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•**Nbtstat** command-line parameters are case-sensitive.
•The following table describes the column headings that are generated by **nbtstat**.

Heading	Description
Input	The number of bytes received.
Output	The number of bytes sent.
In/Out	Whether the connection is from the computer (outbound) or from another computer 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.
<03>	The last byte of a NetBIOS name converted to hexadecimal. Each NetBIOS name is 16 characters long. This last byte often has special significance because the same name might be present several times on a computer, differing only in the last byte. For example, <20> is a space in ASCII text.
Type	The type of name. A name can either be a unique name or a group name.

Status	Whether the NetBIOS service on the remote computer is running (Registered) or a duplicate computer name has registered the same service (Conflict).
State	The state of NetBIOS connections.

•The following table describes the possible NetBIOS connection states.

State	Description
Connected	A 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	A session is in the connecting phase and 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 (it failed to connect on the first attempt).
Outbound	A session is in the connecting phase and 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.

•This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To display the NetBIOS name table of the remote computer with the NetBIOS computer name of CORP07, type:

nbtstat -a CORP07

To display the NetBIOS name table of the remote computer assigned the IP address of 10.0.0.99, type:

nbtstat -A 10.0.0.99

To display the NetBIOS name table of the local computer, type:

nbtstat -n

To display the contents of the local computer NetBIOS name cache, type:

nbtstat -c

To purge the NetBIOS name cache and reload the #PRE-tagged entries in the local Lmhosts file, type:

nbtstat -R

To release the NetBIOS names registered with the WINS server and re-register them, type:

nbtstat -RR

To display NetBIOS session statistics by IP address every five seconds, type:

nbtstat -S 5

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net services overview

Many services use networking commands that 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 /?** at a command prompt.
- You can get syntax help at the command line for a **net** command by typing **net help** *command* at a command prompt. For example, for help with the **net accounts** command, type the following:

  net help accounts
- All **net** commands accept **/y** (yes) and **/n** (no) command-line options. For example, the **net stop server** command prompts you to confirm that you want to stop all dependent Server services, while **net stop server /y**, by automatically answering yes, bypasses the confirmation prompt and shuts down the Server service.
- If a service name contains spaces, use quotation marks around the text (that is, **"***Service Name***"**). For example, the following command starts the net logon service:

  net start "net logon"
- For more information about Net services commands, see Net services commands

Top of page

Manage Your Profile |Legal |Contact Us

Net services commands

For more information, click a command:

- Net accounts
- Net computer
- Net config
- 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 statistics
- Net stop
- Net time
- Net use
- Net user
- Net view

Top of page

Manage Your Profile |Legal |Contact Us

Net accounts

Updates the user accounts database and modifies password and logon requirements for all accounts.

Syntax

net accounts [**/forcelogoff:**{*minutes* | **no**}] [**/minpwlen:***length*] [**/maxpwage:**{*days* | **unlimited**}] [**/minpwage:***days*] [**/uniquepw:***number*] [**/domain**]

⇡Top of page

Parameters

/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 default value, **no**, prevents users from being forced to log off.

/minpwlen:*length* : Sets the minimum number of characters for a user account password. The range is from 0 through 127 characters and the default is six characters.

/maxpwage:{*days* | unlimited} : Sets the maximum number of days that a user account's password is valid. The **unlimited** value sets no maximum time. The **/maxpwage** command-line option must be greater than **/minpwage**. The range is from 1 through 49,710 days (that is, **unlimited** equals 49,710 days), and the default value is 90 days.

/minpwage:*days* : Sets the minimum number of days before a user can change a new password. The default value is zero days, which sets no minimum time. The range is from 0 through 49,710 days.

/uniquepw:*number* : Requires a user to not repeat the same password for the specified *number* of password changes. The range is from 0 through 24 password changes, and the default is five password changes.

/domain : Performs the operation on the primary domain controller of the current domain. Otherwise, the operation is performed on the local computer.

net help *command* : Displays help for the specified **net** command.

⇡Top of page

Remarks

- The Net Logon service must be running on the computer where you want to change account parameters. Used without parameters, **net accounts** displays the current settings for password, logon limitations, and domain information.
- You must do the following before you can use **net accounts**:

  - Create user accounts. Use User Manager or **net user** to set up user accounts.
  - Run the Net Logon service on all servers that verify logon in the domain. Net Logon starts automatically at startup.
- When you use **/forcelogoff:***minutes*, a warning is sent minutes before users are forced to log off from the network. Users are also notified if any files are open. If *minutes* is less than two, users are warned to log off from the network immediately.

⇡Top of page

Examples

To display the current settings, 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 users can reuse a password only after the fifth time they change passwords, type:

net accounts /uniquepw:5

To prevent users from changing passwords more often than every seven days, force users to change passwords every 30 days, and force users to log off after the logon time expires with a five-minute warning, type:

net accounts /minpwage:7 /maxpwage:30 /forcelogoff:5

To ensure that the preceding settings take effect for the domain that the computer is logged on to, type:

net accounts /minpwage:7 /maxpwage:30 /domain

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net computer

Adds or deletes computers from a domain database.

Syntax

net computer \\ComputerName {**/add** | **/del**}

⇡Top of page

Parameters

\\***ComputerName* :** Specifies the computer to add or delete from the domain.

{/add | /del} : Adds or removes the specified computer from the domain.

net help *command* : Displays help for the specified **net** command.

⇡Top of page

Remarks

- **Net computer** forwards all computer additions and deletions to a domain controller.

⇡Top of page

Examples

To add the computer Grizzlybear to a domain, type:

net computer \\\\grizzlybear /add

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net config

Displays the configurable services that are running, or displays and changes settings for a Server service or a Workstation service. Used without parameters, **net config** displays a list of configurable services.

Syntax

net config [{**server**|**workstation**}]

⇡Top of page

Parameters

server : Displays and allows you to make changes to the settings for the Server service while the service is running.

workstation : Displays and allows you to make changes to the settings for the Workstation service while the service is running.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Use **net config server** to change configurable Server service settings. The changes take effect immediately and are permanent.

- You cannot change all of the Server service settings by using **net config server**. **Net config server** displays the following information that you cannot configure:

  - Server computer name
  - Server comment
  - Server version (that is, the software version number)
  - Server is active on (that is, a description of the network)
  - Server hidden (that is, the **/hidden** setting)
  - Maximum logged on users (that is, the maximum number of users who can use the shared resources of the server)
  - Maximum open files per session (that is, the maximum number of server files that users can open during a session)
  - Idle session time (min)
- Use **net config workstation** to change configurable workstation service settings.
- **Net config workstation** displays the following information:

  - Computer name
  - Full computer name
  - User name
  - Workstation active on (that is, a description of the network)
  - Software version (that is, the software version number)
  - Workstation domain
  - Workstation domain DNS name
  - Logon domain
  - COM open timeout (sec)
  - COM send count (byte)
  - COM send timeout (msec)

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Net continue

Continues a service that has been suspended by **net pause**.

Syntax

net continue *service*

⇡Top of page

Parameters

service **:** Required. Specifies the service that you want to continue. The following table lists some of the values for *service*.

Value	Description
netlogon	Continues the Net Logon service.
"nt lm security support provider"	Continues the NT LM Security Support Provider service.
schedule	Continues the Task Scheduler service.
server	Continues the Server service.
workstation	Continues the Workstation service.

net help *command* : Displays help for the specified **net** command.

⇡Top of page

Remarks

- To make a minor correction to a resource, pause the service. Use the **net continue** command to reactivate the service or printer, without canceling user connections.
- The effect of pausing and continuing a service can vary with each service.

- If a service name contains spaces, use quotation marks around the text (for example, "*Service Name*").

⇡Top of page

Examples

The following command continues the Workstation service:

net continue workstation

If the service name consists of two or more words, you must enclose the service name in quotation marks. For example, to continue the NT LM Security Support Provider Service, type:

net continue "nt lm security support provider"

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Net services overview

Net start

Net stop

Net pause

⇡Top of page

Manage Your Profile |Legal |Contact Us

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. Used without parameters, **net file** displays a list of the open files on a server.

Syntax

net file [*ID* [**/close**]]

⇡Top of page

Parameters

ID **:** Specifies the identification number of the file.

/close : Closes an open file and releases locked records. Type this command from the command prompt on the server where the file is shared.

net help *command* : Displays help for the specified **net** command.

⇡Top of page

Remarks

•You can also type **net files** to run this command.

- Use **net file** to view and control files shared on the network. Sometimes, a user leaves a shared file open and locked by mistake. When this happens, other computers on the network cannot gain access to the locked portions of the file. Use **net file /close** to remove the lock and close the file. **Net file** output is similar to the following:

```
File      Path             Username    #locks

---------------------------------------------

0         C:\A_FILE.TXT    MARYSL      0

1         C:\DATABASE      DEBBIET     2
```

Top of page

Examples

To view information about shared files, type:

net file

To close a file that has the identification number 1, type:

net file 1 /close

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net group

Adds, displays, or modifies global groups in domains.

Syntax

net group [*groupname* [**/comment:"***text***"**]] [**/domain**]

net group [*groupname* {**/add** [**/comment:"***text***"**] | **/delete**} [**/domain**]]

net group [*groupname username*[...] {**/add** | **/delete**} [**/domain**]]

Top of page

Parameters

***groupname* :** Specifies the name of the group to add, expand, or delete. Specify a group name to view a list of users in a group only.

/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 domain controller in the current domain. Otherwise, the operation is performed on the local computer.

/add : Adds a group, or adds a user name to a group. You must establish an account for users that you add to a group with this command.

/delete : Removes a group, or removes a user name from a group.

***UserName*[...] :** Lists one or more user names to add to or remove from a group. Separate multiple user name entries with a space.

net help *command* : Displays help for the specified **net** command.

Top of page

Remarks

- Used without parameters, **net group** displays the name of a server and the names of groups on the server.
- You can also type **net groups**.
- Use **net group** 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. In the output, **net group** precedes groups that include both users and groups with an asterisk (*).

  The **net group** output of groups on the server is similar to the following:

```
Group Accounts for \\PRODUCTION

----------------------------------------------

*Domain Admins           *Domain Users
```

Top of page

Examples

To display a list of all of 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 the domain database, type:

net group exec /add /domain

To add the existing user accounts stevev, ralphr, and jennyt to the Exec group on the local computer, type:

net group exec stevev ralphr jennyt /add

To add the existing user accounts stevev, ralphr, and jennyt to the Exec group in the domain database, type:

net group exec stevev ralphr jennyt /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"

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net localgroup

Net user

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net help

Provides a list of network commands and topics for which you can get help, or provides information about a specific command. Used without parameters, **net help** displays a list of commands and topics for which you can get help.

Syntax

net help [*command*]

⇧Top of page

Parameters

command

/help : Displays the correct syntax and the help for the command.

⇧Top of page

Examples

To display help for the **net use** command, type:

net help use

Or, type:

net use /help

To view just the syntax for the **net help** command, type:

net help /?

⇧Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net helpmsg

Explains why an error occurred and provides problem-solving information.

Syntax

net helpmsg *message#*

↑Top of page

Parameters

***message#* :** Required. Specifies the four-digit number of the message you for which you want to get more information.

net help *command* : Displays help for the specified **net** command.

↑Top of page

Remarks

•When a network operation fails, a message similar to the following appears:

```
2182: The requested service has already been started.
```

↑Top of page

Examples

To get more information about error message NET 2182, type:

net helpmsg 2182

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net localgroup

Adds, displays, or modifies local groups. Used without parameters, **net localgroup** displays the name of the server and the names of local groups on the computer.

Syntax

net localgroup [*GroupName* [**/comment:"***text***"**]] [**/domain**]

net localgroup [*GroupName* {**/add** [**/comment:"***text***"**] | **/delete**} [**/domain**]]

net localgroup [*GroupName name* [...] {**/add** | **/delete**} [**/domain**]]

Top of page

Parameters

***GroupName* :** Specifies the name of the local group to add, expand, or delete. Used without additional parameters, **net localgroup** *GroupName* displays 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 contain up to 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.

***name* [...] :** Lists one or more user names or group names to add or remove from a local group.

/add : Adds a global group name or user name to a local group. You must first establish an account for users or global groups before you can add it to a local group with this command.

/delete : Removes a group name or user name from a local group.

net help *command* : Displays help for the specified **net** command.

Top of page

Remarks

- Using **/domain**

  /domain applies only to Windows XP Professional computers that are members of a domain. By default, server computers perform operations on the primary domain controller.
- Using *name*

  Separate multiple entries with a space. Names can be local users, users on other domains, or global groups, but not other local groups. If a user is from another domain, preface the user name with the domain name (for example, Sales\Ralphr).
- Grouping users

  Use **net localgroup** 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 the same rights.

Top of page

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 domain user accounts database, type:

```
net localgroup exec /add /domain
```

To add the existing user accounts stevev, ralphr (from the Sales domain), and jennyt to the Exec local group on the local computer, type:

```
net localgroup exec stevev sales\ralphr jennyt /add
```

To add the existing user accounts stevev, ralphr, and jennyt to the Exec group of a domain, type:

net localgroup exec stevev ralphr jennyt /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."

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Net user

Net group

Command-line reference A-Z

Top of page

Net name

Adds or deletes a messaging name (that is, an alias), or displays the list of names for which the computer can accept messages. Used without parameters, **net name** displays a list of names currently in use.

Syntax

net name [*name* {**/add**|**/delete**}]

⇧Top of page

Parameters

***name* :** Specifies the name to receive messages. The name can have as many as 15 characters.

/add : Adds a name to a computer.

/delete : Removes a name from a computer.

⇧Top of page

Remarks

- Use **net name** to specify a name for receiving messages. You must start the Messenger service before you can use **net name**. Each messaging name must be unique on the network. Names created with **net name** are strictly for messaging, not for group names. Windows XP uses the three following name types:

  - Any name for messaging (added with **net name**)
  - Computer's machine name (added when the Workstation service starts)
  - User name (added when you log on, provided your user name is not in use as a message name elsewhere on the network)
- *add* is optional. **Net name** *name* and **net name** *name add* have the same effect. You can use either one to add a name to the computer.
- You cannot delete the computer's machine name. You can delete the user name.

⇡Top of page

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

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net pause

Pauses services that are currently running.

Syntax

net pause *service*

Top of page

Parameters

***service* :** Required. Specifies the service that you want to suspend. The following table lists some of the values for *service*.

Value	Description
netlogon	Suspends the Net Logon service.
"nt lm security support provider"	Suspends the NT LM Security Support Provider service.
schedule	Suspends the Task Scheduler service.
server	Suspends the Server service.
workstation	Suspends the Workstation service.

net help *command* : Displays help for the specified **net** command.

Top of page

Remarks

- On a server, use **net pause** before you stop a service to let users complete jobs or disconnect from resources. Pausing a service puts it on hold, but it does not 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, pause the service first. Use **net send** to broadcast a message that you are going to stop the service. After providing people with sufficient time to finish using the resource, use **net stop** to stop the service.

  To reactivate a paused service, use **net continue**.
- You cannot pause all services. Pausing affects 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 shared resources of the server. If there are no other logon servers on the network, it prevents users from logging on to the network. Existing connections are not affected. Administrators can make connections to the server even if it is paused.
  - Pausing the Workstation service keeps the user name, password, and connections defined, but directs requests for print devices to printers attached to the computer rather than to printers connected to the network.
- If a service name contains spaces, use quotation marks around the text (for example, **"***Service Name***"**).

Top of page

Examples

To pause the Server service, type:

net pause server

If the service name consists of two or more words, you must enclose the service name in quotation marks. For example, to pause the "NT LM Security Support Provider" service, type:

net pause "nt lm security support provider"

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Net services overview

Net continue

Net start

Net stop

Net use

Top of page

Net print

Displays information about a specified print queue, displays information about all print queues hosted by a specified print server, displays information about a specified print job, or controls a specified print job. Used without parameters, **net print** displays command-line help for the **net print** command.

To display information about a specified print queue or about all print queues hosted by a specified print server

Syntax

net print \\ComputerName[\QueueName]

⇡Top of page

Parameters

\\***ComputerName* :** Required. Specifies, by name, the computer that hosts the print queue about which you want to display information.

\***QueueName* :** Specifies, by name, the print queue for which you want to display information. If you specify a computer but you do not specify a queue, information about all print queues on the specified computer is displayed.

net help net print : Displays help for the **net print** command.

⇡Top of page

Remarks

- **Net print** displays several types of information about print queues. The following is an example of a report for all print queues on the server named Production:

```
Printer queues at \\PRODUCTION

Name              Job #      Size      Status

---------------------------------------------------------

LASER Queue       1 jobs               *Queue Active*

     USER1          84         0      Spooling
```

[Top of page]

Examples

To list the contents of the Dotmatrix print queue on the \\Production computer, type:

net print \\production\dotmatrix

To display information about or control a specified print job

[Top of page]

Syntax

net print [\\ComputerName] *JobNumber* [{**/hold** | **/release** | **/delete**}]

[Top of page]

Parameters

\\***ComputerName* :** Specifies, by name, the computer that hosts the print job you want to control. If you do not specify a computer, the local computer is assumed.

***JobNumber* :** Required. Specifies the number of the print job you want to control. This number is assigned by the computer that hosts the print queue to which the print job is sent. After a computer assigns a number to a print job, that number is not assigned to any other print jobs in any queue hosted by that computer.

{/hold | /release | /delete} : Specifies the action to take with the print job. The **/hold** parameter delays the job, allowing other print jobs to bypass it until it is released. The **/release** parameter releases a print job that has been delayed. The **/delete** parameter removes a print job from a print queue. If you specify a job number but do not specify any action, information about the print job is displayed.

net help net print : Displays help for the **net print** command.

⇡Top of page

Remarks

- The following is an example of a report for a print job:

```
Job #            35

Status           Waiting

Size             3096

Remark

Submitting user  USER2

Notify           USER2

Job data type

Job parameters

Additional info
```

⇡Top of page

Examples

To display information about job number 35 on the \\Production computer, type:

net print \\production 35

To delay 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

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net send

Sends messages to other users, computers, or messaging names on the network.

Syntax

net send {*name* | ***** | **/domain**[**:***name*] | **/users**} *message*

Top of page

Parameters

name **:** Specifies the user name, computer name, or messaging name to which you want to send the message. If the information that you supply contains spaces, use quotation marks around the text (for example, **"***Computer Name***"**). Long user names might cause problems when you use them as NetBIOS names. NetBIOS names are limited to 16 characters, and the sixteenth character is reserved.

*** :** Sends the message to all the names in your domain or workgroup.

/domain:*name* **:** Sends the message to all the names in the computer's domain. You can specify *name* to send the message to all the names in the specified domain or workgroup.

/users : Sends the message to all users connected to the server.

message **:** Required. Specifies the text of the message.

net help *command* **:** Displays help for the specified **net** command.

Top of page

Remarks

- You can send a message only to a name that is active on the network. If you send the message to a user name, that user must be logged on and running the Messenger service to receive the message.
- You can broadcast a message to all of the names in your computer's domain (use ***** or **/domain**), or a different domain (**/domain:***DomainName*). Broadcast messages can contain up to 128 characters. Use discretion when you send messages to multiple users.
- **/users** lets you send a message to all users who have sessions with the server. Use discretion when you send messages to multiple users.
- The Messenger service must be running for messages to be received. For more information about how to start a service, see Related Topics.

Top of page

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"

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Net name

Start, stop, pause, resume, or restart a service

Top of page

Manage Your Profile |Legal |Contact Us

Net session

Manages server computer connections. Used without parameters, **net session** displays information about all sessions with the local computer.

Syntax

net session [\\ComputerName] [**/delete**]

⇡Top of page

Parameters

\\ComputerName **:** Identifies the computer for which you want 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 you omit *ComputerName*, all sessions on the local computer are canceled.

net help *command* : Displays Help for the specified **net** command.

Caution

- Using **net session** can result in a loss of data. You might want to warn users before you disconnect a session.

⇡Top of page

Remarks

- You can also use **net sessions** or **net sess** to run **net session**.

- Use **net session** to view the computer names and user names of users on a server, to see if users have files open, and to see how long each user's session has been idle.

  The information appears in a format similar to the following:

```
Computer      User name        Client type  Opens   Idle time

------------------------------------------------------------------------
\\BASSETT     CHRISDR          Windows 2000   1     00:00:13
\\SHARONCA    Administrator    DOS LM 2.1     0     01:05:13
```

- To display one user's session, use *ComputerName* with the command. The information about a single user 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 computers are on the same network, and the user has a user name and password that are accepted by the server. A user at a client has to have a session with a server before he or she can use the resources of the server, 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 by using **net config server /autodisconnect**. For more information about **net config server**, see Related Topics. An automatic disconnection is transparent to the user because the session is automatically reconnected when the user accesses the resource again.
- To end a session with the server, use **net session** \\ComputerName **/delete**.

Top of page

Examples

To display a list of session information for the local server, type:

net session

To display session information for a client with the computer name Shepherd, type:

net session \\shepherd

To end all sessions between the server and the clients connected to it, type:

net session /delete

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net config

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net share

Manages shared resources. Used without parameters, **net share** displays information about all of the resources that are shared on the local computer.

Syntax

net share [*ShareName*] **net share** [*ShareName*=*Drive*:*Path* [{**/users:***number*|**/unlimited**}] [/**remark:"***text***"**] [/**cache:** {**manual**|**automatic**|**no**}]] **net share** [*ShareName* [{**/users:***number*|**unlimited**}] [**/remark:"***text***"**] [/**cache:** {**manual**|**automatic**|**no**}]] **net share** [{*ShareName*|*Drive*:*Path*} **/delete**]

Top of page

Parameters

ShareName **:** Specifies the network name of the shared resource. Type **net share** with a *ShareName* to display information about that share only.

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.

/cache:automatic : Enables offline client caching with automatic reintegration.

/cache:manual : Enables offline client caching with manual reintegration.

/cache:no : Advises the client that offline caching is inappropriate.

/delete : Stops sharing the shared resource.

net help *command* : Displays Help for the specified **net** command.

⇡Top of page

Remarks

- To share a directory with a path that contains a white space, enclose the drive and the path of the directory in quotation marks (for example, **"C:\Path Name"**).
- When you display all of the shared resources on a computer, the share name of the resource, the device names or path associated with the resource, and a descriptive comment about the resource is displayed. The output 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. For more information about services, see Services
- Shared resource names that end in a **$** character are not displayed when you are browsing the local computer from a remote computer.

⇡Top of page

Examples

To display information about shared resources on the computer, type:

net share

To share a computer's C:\Data directory with the share name DataShare and include a remark, type:

net share DataShare=c:\Data /remark:"For department 123."

To stop sharing the DataShare folder you created in the previous example, type:

net share DataShare /delete

To share a computer's C:\Art Lst directory with the share name List, type:

```
net share list="c:\art lst"
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Shared Folders

Top of page

Manage Your Profile |Legal |Contact Us

Net start

Starts a service. Used without parameters, **net start** displays a list of services that are currently operating.

Syntax

net start [*service*]

Top of page

Parameters

***service* :** Starts the specified service. The following table lists values for *service*.

Value	Description	Remarks
alerter	Starts the Alerter service.	•Start the Alerter service to send alert messages to specified users that are connected to the server. Alert messages warn users about security, access, and user session problems. •Use Server Manager (that is, *systemroot* \System32\Srvmgr.exe) to specify the administrators who you want to receive administrative alerts. Server Manager is available on Windows Server 2000 computers only. •Alert messages are sent as 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 alert messages.

browser	Starts the Computer Browser service.	•The Computer Browser service maintains an up-to-date list of computers on your network and supplies the list to programs that request it.
"client service for netware"	Starts the Client Service for NetWare service.	•This command is available only if you have installed Client Service for NetWare.
clipbook	Starts the ClipBook service.	•ClipBook permits you to cut and paste text and graphics over the network. •The ClipBook service supports ClipBook Viewer, which allows pages to be seen by remote ClipBooks.
dhcp client	Starts the DHCP Client service.	•This command is available only if you have installed the TCP/IP protocol. •The DHCP Client service manages network configuration by registering and updating IP addresses and DNS names. The DHCP Client service supports obtaining an IP address from the DHCP service. •You cannot stop or pause the DHCP Client service.
eventlog	Starts the Event Log service.	•The Event Log service logs event messages issued by programs and Windows XP. Event Log reports contain information that can be useful in diagnosing problems. Reports are viewed in Event Viewer. The Event Log service must be started prior to using Event Viewer to view the logged events. •You cannot stop or pause the Event Log service.
file replication	Starts the File Replication service.	

messenger	Starts the Messenger service.	•The Messenger service enables a computer to receive messages. •A message is sent to a computer using the names assigned to the computer as identification.
netlogon	Starts the Net Logon service.	•The Net Logon service verifies logon requests and controls domain-wide replication of the user accounts database. •Start the Net Logon service on all the servers in a domain that use a copy of the domain's user accounts database.
"nt lm security support provider"	Starts the NT LM Security Support Provider service.	•This command is available only if you have installed the NT LM Security Support Provider.
"plug and play"	Starts the Plug and Play service.	
"remote access connection manager"	Starts the Remote Access Connection Manager service.	•This command is available only if you have installed the Remote Access Service.
"routing and remote access"	Starts the Routing and Remote Access service.	
rpclocator	Starts the Remote Procedure Call (RPC) Locator service.	•Allows distributed applications to use the Microsoft RPC name service. •The Remote Procedure Call (RPC) Locator service is the RPC name service for Microsoft Windows XP. 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 RPC Locator service to find available compatible server applications.

rpcss	Starts the Remote Procedure Call (RPC) service.	•The Remote Procedure Call (RPC) service is the RPC subsystem for Windows XP. The RPC subsystem includes the endpoint mapper and other miscellaneous RPC services. **Net start rpcss** starts the Remote Procedure Call (RPC) service to allow distributed applications to use dynamic endpoints. The Remote Procedure Call (RPC) service manages the endpoint map database. The server side of the distributed application registers its endpoints with the Remote Procedure Call (RPC) service. The client run-time library, on behalf of the client side of the distributed application, queries the Remote Procedure Call (RPC) service to obtain the endpoint information. To determine whether a distributed application uses the endpoint mapper service, see the documentation for that distributed application.
schedule	Starts the Task Scheduler service.	•With Task Scheduler, you can start programs at a specified time with the **at** command. You might need to have other services running before you can run scheduled commands. •Task Scheduler is initially configured to run in the System account on the local computer. When Task Scheduler runs using this account, there are no restrictions on the jobs that you can run with Task Scheduler. However, these jobs 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 Task Scheduler to run in a user's account. If you do this, jobs executed by Task Schedule are governed by the user account's network access. However, because Task Scheduler is not using the local System account in this case, you can only run jobs that do not require the presence of a window.
server	Starts the Server service.	•You can use the Server service to share server resources with users on the network.
spooler	Starts the Print Spooler service.	•The Print Spooler service loads files to memory for printing.
"tcp/ip netbios helper"	Starts the NetBIOS helper over TCP service, which enables NetBIOS over TCP/IP (NetBT) services.	•NetBT services provide NetBIOS data grams, NetBIOS sessions, and NetBIOS name management (such as name registration and resolution) for NetBIOS applications that are using the TCP/IP protocol. •This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections
ups	Starts the Uninterruptible Power Supply service.	•The Uninterruptible Power Supply service manages an uninterruptible power supply (UPS) connected to the computer.

		•You can configure the Uninterruptible Power Supply service by using Power Options in Control Panel. If you configure the Uninterruptible Power Supply service to execute a command file upon shutdown the command file must finish running in 30 seconds. A run time that is greater than 30 seconds threatens the capability of Windows XP to complete a graceful system shutdown.
workstation	Starts the Workstation service.	•The Workstation service enables a computer to connect to and use network resources.

net help *command* : Displays help for the specified **net** command.

Top of page

Remarks

- The set of services and applications you see might vary depending on the options you selected during installation or configuration.
- For more information about services, see the "System Essentials Guide" at the Microsoft Web site.
- Some services are dependent on other services.
- You can use the Services snap-in to configure services to stop and start automatically. Use this snap-in to manually stop, start, pause, and continue network services.
- You can use **Net start** to start services that are not provided with Windows XP.
- If a service name contains spaces, use quotation marks around the text (for example, "*Service Name*").

Top of page

Examples

To list services that are currently operating, type:

net start

To start the Client Service for Netware service, type:

```
net start "client service for netware"
```

Command-line reference A-Z

Net services overview

Net pause

Net stop

Net continue

Top of page

Manage Your Profile |Legal |Contact Us

Net statistics

Displays the statistics log for the local Workstation or Server service, or the running services for which statistics are available. Used without parameters, **net statistics** lists the running services for which statistics are available.

Syntax

net statistics [{**workstation** | **server**}]

⇡Top of page

Parameters

workstation : Displays statistics for the local Workstation service.

server : Displays statistics for the local Server service.

net help *command* : Displays help for the specified **net** command.

⇡Top of page

Remarks

- You can also type **net stats**.
- For the Workstation service, Windows XP reports the computer's name, the date and time when the statistics were last updated, and the following information:

  - Number of bytes and server message blocks (SMB) received and transmitted
  - Number of read and write operations that succeeded or failed
  - Number of network errors
  - Number of sessions that failed, disconnected, or were reconnected
  - Number of connections to shared resources that succeeded or failed

- For the Server service, Windows XP reports the computer's name, the date and time when the statistics were last updated, and the following information:
  - Number of sessions that were started, disconnected automatically, and disconnected because of an error
  - Number of kilobytes sent and received, and the average server-response time
  - Number of errors and violations of password and permission limits
  - Number of times the shared files, printers, and communication devices were used
  - Number of times the size of the memory buffer was exceeded

Top of page

Examples

To display statistics for the Server service and prevent the output from scrolling, type:

net statistics server | more

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

More

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net stop

Stops a running service.

Syntax

net stop *service*

Top of page

Parameters

service **:** Stops the specified service. The following table lists values for *service*.

Value	Description	Remarks
alerter	Stops the Alerter service.	•Stop the Alerter service that sends alert messages to specified users that are connected to the server computer. Alert messages warn users about security, access, and user session problems. •Use Server Manager (that is, *systemroot* \System32\Srvmgr.exe) to specify the administrators who you want to receive administrative alerts. Server Manager is available on Windows 2000 Server computers only. •Alert messages are sent as 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 alert messages.

browser	Stops the Computer Browser service.	•The Computer Browser service maintains an up-to-date list of computers on your network and supplies the list to programs that request it.
"client service for netware"	Stops the Client Service for NetWare service.	•This command is available only if you have installed Client Service for NetWare.
clipbook	Stops the ClipBook service.	•ClipBook permits you to cut and paste text and graphics over the network. •The ClipBook service supports ClipBook Viewer, which allows pages to be seen by remote ClipBooks.
dhcp client	You cannot stop or pause the DHCP Client service.	•The DHCP Client service manages network configuration by registering and updating IP addresses and DNS names. The DHCP Client service supports obtaining an IP address from the DHCP service.
file replication	Starts the File Replication service.	
messenger	Stops the Messenger service.	•The Messenger service enables a computer to receive messages. •A message is sent to a computer using the names assigned to the computer as identification.
netlogon	Stops the Net Logon service.	•The Net Logon service verifies logon requests and controls domain-wide replication of the user accounts database. •Stop the Net Logon service on all the servers in a domain that use a copy of the domain's user accounts database.
"nt lm security support provider"	Stops the NT LM Security Support Provider service.	•This command is available only if you have installed the NT LM Security Support Provider.

"remote access connection manager"	Stops the Remote Access Connection Manager service.	•This command is available only if you have installed the Remote Access Service.
"routing and remote access"	Stops the Routing and Remote Access service.	
rpclocator	Stops the Remote Procedure Call (RPC) Locator service.	•Allows distributed applications to use the Microsoft RPC name service. •The Remote Procedure Call (RPC) Locator service is the RPC name service for Microsoft Windows XP. 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 RPC Locator service to find available compatible server applications.
schedule	Stops the Task Scheduler service.	•With Task Scheduler, you can start programs at a specified time with the **at** command. You might need to have other services running before you can run scheduled commands. •Task Scheduler is initially configured to run in the System account on the local computer. When Task Scheduler runs using this account, there are no restrictions on the jobs that you can run with Task Scheduler. However, these jobs 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 Task Scheduler to run in a user's account. If you do this, jobs executed by Task Schedule are governed by the user account's network access. However, because Task Scheduler is not using the local System account in this case, you can only run jobs that do not require the presence of a window.
server	Stops the Server service.	•You can use the Server service to share server resources with users on the network.
spooler	Stops the Print Spooler service.	•The Print Spooler service loads files to memory for printing.
"tcp/ip netbios helper"	Stops the NetBIOS helper over TCP service, which enables NetBIOS over TCP/IP (NetBT) services.	•NetBT services provide NetBIOS data grams, NetBIOS sessions, and NetBIOS name management (such as name registration and resolution) for NetBIOS applications that are using the TCP/IP protocol. •This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections
ups	Stops the Uninterruptible Power Supply service.	•The Uninterruptible Power Supply service manages an uninterruptible power supply (UPS) connected to the computer.

		•You can configure the Uninterruptible Power Supply service by using Power Options in Control Panel. If you configure the Uninterruptible Power Supply service to execute a command file upon shutdown the command file must finish running in 30 seconds. A run time that is greater than 30 seconds threatens the capability of graceful system shutdown completing successfully.
workstation	Stops the Workstation service.	•The Workstation service enables a computer to connect to and use network resources.

net help *command* : Displays help for the specified **net** command.

Top of page

Remarks

- The set of services and applications you see might vary depending on the options you selected during installation or configuration.
- For more information about services, see Microsoft Windows Resource Kits
- Some services are dependent on other services.
- Use **net stop** to stop a service and remove the software from memory.
- Stopping the Server service prevents users from gaining access to the computer's shared resources. If you stop the Server service when users are using server resources, a warning message appears. A **y** (that is, **yes**) response cancels all connections to the computer. Before you stop the Server service, do the following:
- Use **net pause** to pause the service, which prevents new connections.
- Use **net send** to send a message that advises users to disconnect from the server resources.
- If a service name contains spaces, use quotation marks around the text (for example, "*Service Name*").
- **Net stop** can also stop services not provided with Windows XP.

Top of page

Examples

The following command stops the Server service:

net stop server

Enclose service names with two or more words in quotation marks. For example, the following command stops the Client Service for Netware service:

net stop "client service for netware"

⇧Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Net services overview

Net start

Net pause

Net continue

Net send

Top of page

Manage Your Profile |Legal |Contact Us

Net time

Synchronizes the computer's clock with that of another computer or domain. Used without parameters, **net time** displays the time for another computer or domain.

Syntax

net time [{\\ComputerName | **/domain**[**:***DomainName*] | **/rtsdomain**[**:***DomainName*]}] [**/set**]

net time [\\ComputerName] [**/querysntp**] [**/setsntp**[**:***NTPServerList*]]

Top of page

Parameters

\\***ComputerName* :** Specifies the name of a server you want to check or with which you want to synchronize.

/domain[:*DomainName*] : Specifies the domain with which to synchronize clocks.

/rtsdomain[:*DomainName*] : Specifies the domain of the Reliable Time Server with which to synchronize clocks.

/set : Synchronizes the computer's clock with the time on the specified computer or domain.

/querysntp : Displays the name of the Network Time Protocol (NTP) server currently configured for the local computer or the one specified in *ComputerName*.

/setsntp[:*NTPServerList*] : Specifies a list of NTP time servers to be used by the local computer. The list can contain IP addresses or DNS names, separated by spaces. If you use multiple time servers, you must enclose the list in quotation marks.

net help *command* : Displays help for the specified **net** command.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

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. Used without parameters, **net use** retrieves a list of network connections.

Syntax

net use [{*DeviceName* | *****}] [\\*ComputerName*\ShareName[\volume]] [{*Password* | *****}]] [**/user:**[*DomainName*\\]*UserName*] [**/user:**[*DottedDomainName*\\]*UserName*] [**/user:**[*UserName@DottedDomainName*] [**/savecred**] [**/smartcard**] [{**/delete** | **/persistent:**{**yes** | **no**}}]

net use [*DeviceName* [**/home**[{*Password* | *****}] [**/delete:**{**yes** | **no**}]]

net use [**/persistent:**{**yes** | **no**}]

Top of page

Parameters

DeviceName **:** Assigns a name to connect to the resource or specifies the device to be disconnected. There are two kinds of device names: disk drives (that is, D: through Z:) and printers (that is, LPT1: through LPT3:). Type an asterisk (*) instead of a specific device name to assign the next available device name.

\\*ComputerName*\*ShareName* **:** Specifies the name of the server and the shared resource. If *ComputerName* contains spaces, use quotation marks around the entire computer name from the double backslash (\\\\) to the end of the computer name (for example, **"**\\Computer Name \*Share Name***"**). The computer name can be from 1 to 15 characters long.

\*volume* **:** Specifies a NetWare volume on the server. You must have Client Service for NetWare installed and running to connect to NetWare servers.

Password **:** Specifies the password needed to access the shared resource. Type an asterisk (*) to produce a prompt for the password. The password is not displayed when you type it at the password prompt.

/user : Specifies a different user name with which the connection is made.

***DomainName* :** Specifies another domain. If you omit *DomainName*, **net use** uses the current logged on domain.

***UserName* :** Specifies the user name with which to log on.

***DottedDomainName* :** Specifies the fully-qualified domain name for the domain where the user account exists.

/savecred : Stores the provided credentials for reuse.

/smartcard : Specifies the network connection is to use the credentials on a smart card. If multiple smart cards are available, you are asked to specify the credential.

/delete : Cancels the specified network connection. If you specify the connection with an asterisk (*), all network connections are canceled.

/persistent:{yes | no} : 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 are restored at the next logon. Use **/delete** to remove persistent connections.

/home : Connects a user to the home directory.

net help *command* : Displays help for the specified **net** command.

↑Top of page

Remarks

•Connecting and disconnecting from a network resource

Use **net use** 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 you use it as your current drive or an active process is using it.

- Viewing connection information

  To view information about a connection, you can do either of the following:

  - Type **net use** *DeviceName* to get information about a specific connection.
  - Type **net use** to get a list of all the computer's connections.
- Using deviceless connections

  Deviceless connections are not persistent.
- Connecting to NetWare servers

  After you install and run Client Service for NetWare, you can connect to a NetWare server on a Novell network. Use the same syntax that you use to connect to a Windows Networking server, except you must include the volume you to which you want to connect.
- Using quotation marks

  If the *ServerName* that you supply contains spaces, use quotation marks around the text (that is, "*Server Name*"). If you omit quotation marks, an error message appears.

⇡Top of page

Examples

To assign the disk-drive device name E: to the Letters shared directory on the \\Financial server, type:

net use e: \\financial\letters

To assign (map) the disk-drive device name M: to the directory Mike within the Letters volume on the \\Financial NetWare server, type:

net use m: \\financial\letters\mike

To connect the user identifier Dan as if the connection were made from the Accounts domain, type:

net use d:\\server\share /user:Accounts\Dan

To disconnect from the \\Financial\Public directory, type:

net use f: \\financial\public /delete

To connect to the resource memos shared on the \\Financial 2 server, type:

net use k: "\\financial 2" \memos

To restore the current connections at each logon, regardless of future changes, type:

net use /persistent:yes

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Net user

Adds or modifies user accounts or displays user account information.

Syntax

net user [*UserName* [*Password* | *****] [*options*]] [**/domain**]

net user [*UserName* {*Password* | *****} **/add** [*options*] [**/domain**]]

net user [*UserName* [**/delete**] [**/domain**]]

⇡Top of page

Parameters

***UserName* :** Specifies 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. Type an asterisk (*) to produce a prompt for the password. The password is not displayed when you type it at the password prompt.

/domain : Performs the operation on the domain controller in the computer's primary domain.

***options* :** Specifies a command-line option. The following table lists valid command-line options that you can use.

Command-line option syntax	Description
/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** (that is, 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/ Region codes to implement the specified language files for a user's Help and error messages. A value of 0 signifies the default Country/Region code.
/expires:{{*mm*/*dd*/*yyyy* | *dd*/*mm*/*yyyy* | *mmm***,**dd **,** *yyyy*} | **never**}	Causes the user account to expire if you specify *date*. Expiration dates can be in [*mm*/*dd*/*yyyy*], [*dd*/*mm*/*yyyy*], or [*mmm***,** dd **,***yyyy*] format, depending on the Country/Region code. Note that the account expires at the beginning of the specified date. For the month value, you can use numbers, spell it out, or use a three-letter abbreviation (that is, Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec). You can use two or four numbers for the year value. Use commas or slashes to separate parts of the date. Do not use spaces. If you omit *yyyy*, the next occurrence of the date (that is, according to your computer's date and time) is assumed. For example, the following 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 user name. Enclose the name in quotation marks.
/homedir:*Path*	Sets the path for the user's home directory. The path must exist.

/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 path 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:{*day*[-*day*][**,***day*[-*day*]] **,***time*[-*time*][**,***time*[-*time*]] [**;**] | **all**}	Specifies the times that users are allowed to use the computer. *Time* is limited to 1-hour increments. For the *day* values, you can spell out or use abbreviations (that is, M,T,W,Th,F,Sa, Su). You can use 12-hour or 24-hour notation for hours. If you use 12-hour notation, use AM and PM, or A.M. and 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***"**	Specifies that an administrator can 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 an asterisk (*), users can log on from any computer.

net help *command* : Displays help for the specified **net** command.

⇡Top of page

Remarks

- Used without parameters, **net user** displays a list of the user accounts on the computer. You can also type **net users**.
- A password must satisfy the minimum length set with **net accounts /minpwlen**. It can have as many as 127 characters. However, if you are using Windows 2000 or Windows XP on a network that also has computers using Windows 95 or Windows 98, consider using passwords not longer than 14 characters. Windows 95 and Windows 98 support passwords of up to 14 characters. If your password is longer, you might not be able to log on to your network from those computers.

⇡Top of page

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 Jay Jamison, with logon rights from 8 A.M. to 5 P.M., Monday through Friday (no spaces in time designations), a mandatory password (jayj), and the user's full name, type:

net user jayj /add /passwordreq:yes /times:monday-friday,8am-5pm/fullname:"Jay Jamison"

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

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Net accounts

Net localgroup

Net group

Command-line reference A-Z

↑Top of page

Manage Your Profile |Legal |Contact Us

Net view

Displays a list of domains, computers, or resources that are being shared by the specified computer. Used without parameters, **net view** displays a list of computers in your current domain.

Syntax

net view [\\ComputerName] [**/domain**[**:***DomainName*]]

net view /network:nw [\\ComputerName]

⇡Top of page

Parameters

\\*ComputerName* : Specifies the computer that contains the shared resources that you want to view.

/domain[:*DomainName*] : Specifies the domain for which you want to view the available computers. If you omit *DomainName*, **/domain** displays all of the domains in the network.

/network:nw : Displays all available servers on a NetWare network. If you specify a computer name, **/network:nw** displays the resources available on that computer in the NetWare network. You can also specify other networks that are added to the system.

net help *command* : Displays help for the specified **net** command.

⇡Top of page

Remarks

- Use the **net view** command to display a list of computers. The output is similar to the following:

```
Server Name           Remark

-------------------------------------------------

\\Production          Production file server

\\Print1              Printer room, first floor

\\Print2              Printer room, second floor
```

Top of page

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

Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Net services overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Using Netsh

Netsh is a command-line scripting utility that allows you to, either locally or remotely, display or modify the network configuration of a computer that is currently running. **Netsh** also provides a scripting feature that allows you to run a group of commands in batch mode against a specified computer. **Netsh** can also save a configuration script in a text file for archival purposes or to help you configure other servers.

Netsh contexts

Netsh interacts with other operating system components using dynamic-link library (DLL) files. Each Netsh helper DLL provides an extensive set of features called a context, which is a group of commands specific to a networking component. These contexts extend the functionality of **netsh** by providing configuration and monitoring support for one or more services, utilities, or protocols. For example, Dhcpmon.dll provides **netsh** the context and set of commands necessary to configure and manage DHCP servers.

To run a **netsh** command, you must start **netsh** from the Cmd.exe prompt and change to the context that contains the command you want to use. The contexts that are available to you depend on which networking components you have installed. For example, if you type **dhcp** at the Netsh command prompt, you change to the DHCP context, but if you do not have DHCP installed the following message appears:

The following command was not found: dhcp.

For more information about Netsh contexts, see the following topics:

- Netsh commands for AAAA
- Netsh commands for DHCP
- Netsh diagnostic (diag) commands
- Netsh commands for Interface IP
- Netsh commands for RAS
- Netsh commands for Routing
- Netsh commands for WINS

⇡Top of page

Using multiple contexts

A context can exist within a context. For example, within the Routing context, you can change to the IP and IPX subcontexts.

To display a list of commands and subcontexts that you can use within a context, at the **netsh** prompt, type the context name, and then type either **/?** or **help**. For example, to display a list of subcontexts and commands that you can use in the Routing context, at the **netsh** prompt (that is, netsh>), type either of the following:

routing /?

routing help

To perform tasks in another context without changing from your current context, type the context path of the command you want to use at the **netsh** prompt. For example, to add the Local Area Connection interface in the IGMP context without changing to the IGMP context, at the **netsh** prompt, type:

routing ip igmp add interface "Local Area Connection" startupqueryinterval=21

⇡Top of page

Running Netsh commands from the Cmd.exe command prompt

When you run Netsh from the Cmd.exe command prompt, **netsh** uses the following syntax. To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP.

To view the command syntax, click the following command:

netsh

Netsh is a command-line scripting utility that allows you to, either locally or remotely, display or modify the network configuration of a currently running computer. Used without parameters, **netsh** opens the Netsh.exe command prompt (that is, netsh>).

Syntax

netsh [**-a** *AliasFile*] [**-c** *Context*] [**-r** *RemoteComputer*] [{*NetshCommand*|**-f** *ScriptFile*}]

Parameters

-a : Returns you to the **netsh** prompt after running *AliasFile*.

***AliasFile* :** Specifies the name of the text file that contains one or more **netsh** commands.

-c : Changes to the specified **netsh** context.

***Context* :** Specifies the **netsh** context. The following table lists the available **netsh** contexts.

Context	Description
AAAA	Shows and sets the configuration of the authentication, authorization, accounting, and auditing (AAAA) database used by the Internet Authentication Service (IAS) and the Routing and Remote Access service.
DHCP	Administers DHCP servers and provides an equivalent alternative to console-based management.
Diag	Administers and troubleshoots operating system and network service parameters.
Interface	Configures the TCP/IP protocol (including addresses, default gateways, DNS servers, and WINS servers) and displays configuration and statistical information.
RAS	Administers remote access servers.
Routing	Administers Routing servers.
WINS	Administers WINS servers.

-r : Configures a remote computer.

***RemoteComputer* :** Specifies the remote computer to configure.

***NetshCommand* :** Specifies the **netsh** command that you want to run.

-f : Exits Netsh.exe after running the script.

***ScriptFile* :** Specifies the script that you want to run.

/?: Displays help at the command prompt.

Remarks

- Using **-r**

  If you specify **-r** followed by another command, **netsh** executes the command on the remote computer and then returns to the Cmd.exe command prompt. If you specify **-r** without another command, **netsh** opens in remote mode. The process is similar to using **set machine** at the Netsh command prompt. When you use **-r**, you set the target computer for the current instance of **netsh** only. After you exit and reenter **netsh**, the target computer is reset as the local computer. You can run **netsh** commands on a remote computer by specifiying a computer name stored in WINS, a UNC name, an internet name to be resolved by the DNS server, or a numerical IP address.

⇡Top of page

Running Netsh commands from the Netsh.exe command prompt

Netsh uses the following standard commands in all contexts that you can run from a Netsh.exe command prompt (that is, netsh>). To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP.

To view the command syntax, click a command:

..

Moves to the context that is one level up.

Syntax

..

Parameters

/?: Displays help at the command prompt.

abort

Discards any changes made in offline mode. **Abort** has no effect in online mode.

Syntax

abort

Parameters

/?: Displays help at the command prompt.

add helper

Installs the helper DLL in **netsh**.

Syntax

add helper *DLLName*

Parameters

***DLLName* :** Required. Specifies the name of the helper DLL you want to install.

/?: Displays help at the command prompt.

alias

Adds an alias that consists of a user-defined character string, which **netsh** treats as equivalent to another character string. Used without parameters, **alias** displays all available aliases.

Syntax

alias [*AliasName*] [*string1* [*string2* ...]]

Parameters

alias [*AliasName*] : Displays the specified alias.

alias [*AliasName*][*string1*[*string2* ...]] : Sets *AliasName* to the specified string(s).

/?: Displays help at the command prompt.

Examples

The following **netsh** sample script sets two **netsh** aliases, Shaddr and Shp, and then leaves the Netsh command prompt in the Interface IP context:

```
alias shaddr show interface ip addr
alias shp show helpers
interface ip
```

If you type **shaddr** at the Netsh command prompt, Netsh.exe interprets this as the command **show interface ip addr**. If you type **shp** at the Netsh command prompt, Netsh.exe interprets this as the command **show helpers**.

bye

Exits Netsh.exe.

Syntax

bye

Parameters

/?: Displays help at the command prompt.

commit

Commits any changes made in the offline mode to the router. **Commit** has no effect in online mode.

Syntax

commit

Parameters

/?: Displays help at the command prompt.

delete helper

Removes the helper DLL from **netsh**.

Syntax

delete helper *DLLName*

Parameters

DLLName **:** Required. Specifies the name of the helper DLL you want to uninstall.

/?: Displays help at the command prompt.

dump

Creates a script that contains the current configuration. If you save this script to a file, you can use the file to restore configuration settings that have been changed. Used without parameters, **dump** displays all of the **netsh** context configurations.

Syntax

dump [*FileName*]

Parameters

[*FileName*] : Specifies the name of the file to which you want to redirect output.

/?: Displays help at the command prompt.

exec

Loads a script file and runs commands from it.

Syntax

exec *ScriptFile*

Parameters

***ScriptFile* :** Required. Specifies the name of the script that you want to load and run.

Remarks

- The *ScriptFile* can run on one or more computers.

exit

Exits Netsh.exe.

Syntax

exit

Parameters

/?: Displays help at the command prompt.

help

Displays help.

Syntax

{**/?**|**?**|**help**|**h**}

Parameters

none

offline

Sets the current mode to offline.

Syntax

offline

Parameters

/?: Displays help at the command prompt.

Remarks

- Changes that you make in this mode are saved, but you need to run the **commit** or **online** command to set the changes in the router.
- When you switch from offline mode to online mode, changes that you made in offline mode are reflected in the configuration that is currently running.
- Changes that you make in online mode are immediately reflected in the configuration that is currently running.

online

Sets the current mode to online.

Syntax

online

Parameters

/?: Displays help at the command prompt.

Remarks

- Changes that you make in online mode are immediately reflected in the configuration that is currently running.
- When you switch from offline mode to online mode, changes that you made in offline mode are reflected in the configuration that is currently running.

popd

Restores a context from the stack.

Syntax

popd

Parameters

/?: Displays help at the command prompt.

Remarks

- Used in conjunction with **pushd**, **popd** enables you to change the context, run the command in the new context, and then resume the prior context.

Examples

The following sample script changes a context from the root context to the **interface ip** context, adds a static IP route, and then returns to the root context:

```
netsh>
```

pushd

```
netsh>
```

interface ip

```
netsh interface ip>
```

set address local static 10.0.0.9 255.0.0.0 10.0.0.1 1

```
netsh interface ip>
```

popd

```
netsh>
```

pushd

Saves the current context on a first-in-last-out (FILO) stack.

Syntax

pushd

Parameters

/?: Displays help at the command prompt.

Remarks

- Used in conjunction with **popd**, **pushd** enables you to change the context, run the command in the new context, and then resume the prior context.

quit

Exits Netsh.exe.

Syntax

quit

Parameters

/?: Displays help at the command prompt.

set file

Copies the Command Prompt window output to a file.

Syntax

set file {**open** *FileName*|**append** *FileName*|**close**}

Parameters

open *FileName* : Sends the Command Prompt window output to the specified file.

append *FileName* : Appends the Command Prompt window output to the specified existing file.

close : Stops sending output and closes a file.

/?: Displays help at the command prompt.

Remarks

- Using *FileName*

  If the specified *FileName* does not currently exist, **netsh** creates a new file with that name. If the specified *FileName* does currently exist, **netsh** overwrites the existing data.

Examples

To create a new log file called Session.log and copy all succeeding **netsh** input and output to Session.log, type:

set file open c:\session.log

set machine

Sets the current computer on which to perform configuration tasks. Used without parameters, **set machine** sets the local computer.

Syntax

set machine [[*ComputerName*=]*string*]

Parameters

***ComputerName* :** Specifies the name of the computer on which to perform configuration tasks.

***string* :** Specifies the location of the remote computer.

/?: Displays help at the command prompt.

Remarks

- Performing configuration tasks on multiple computers

  You can run commands on multiple computers from a single script. You can use **set machine** in a script to specify a destination computer (for example, ComputerA), and then run the commands that follow **set machine** on this computer (that is, ComputerA). You can then use **set machine** to specify another destination computer (for example, ComputerB), and then run commands on this computer (that is, ComputerB).

set mode

Sets the current mode to online or offline.

Syntax

set mode {**online**|**offline**}

Parameters

online : Sets the current mode to online.

offline : Sets the current mode to offline.

/?: Displays help at the command prompt.

show

Displays alias, helper, and mode information.

Syntax

show {**alias**|**helper**|**mode**}

Parameters

alias : Lists all defined aliases.

helper : Lists all top-level helpers.

mode : Displays the current mode.

/?: Displays help at the command prompt.

unalias

Deletes the specified alias.

Syntax

unalias *AliasName*

Parameters

***AliasName* :** Required. Specifies the name of the alias.

/?: Displays help at the command prompt.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Netsh commands for AAAA

You can use commands in the Netsh AAAA context to show and set the configuration of the authentication, authorization, accounting, and auditing (AAAA) database used by the Internet Authentication Service (IAS) and the Routing and Remote Access service. The AAAA database is also known as the IAS database (Ias.mdb). The primary use of commands in the Netsh AAAA context is to export the configuration of one IAS server as a Netsh script and then to import that configuration using the **netsh exec** command on another IAS server.

You cannot run these Netsh commands locally on Windows XP Professional or Windows XP Home Edition. To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. You can run these commands from the Windows 2000 command prompt or from the command prompt for the Netsh AAAA context. For these commands to work at the Windows 2000 command prompt, you must type **netsh aaaa** before typing commands and parameters as they appear in the syntax below. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP.

For more information on Netsh, see Using Netsh

To view the command syntax, click a command:

dump

Displays the configuration of the IAS database file (Ias.mdb) as a Netsh command script.

Syntax

dump

⇡Top of page

Remarks

- The **dump** command displays the Netsh command script that you can use to duplicate the configuration of the server running IAS or the Routing and Remote Access service on which the command is executed. The Netsh command script contains the configuration of the IAS database file (Ias.mdb) in a compressed text format as a large data block. This large data block is used by the **set config** command within the script to import the configuration of a saved data block into an existing IAS database on the same or another computer using the **netsh exec** command. To save the Netsh command script to a file, type the following at a command prompt: **netsh aaaa show config >** *Path\File.txt*
- The **dump** command is not supported on computers running Windows 2000 Server.

show config

Displays the configuration of the IAS database file (Ias.mdb) as a Netsh command script.

⇡Top of page

Syntax

show config

⇡Top of page

Remarks

- The **show config** command is equivalent to the **dump** command.
- The **show config** command displays the Netsh command script that you can use to duplicate the configuration of the server running IAS or the Routing and Remote Access service on which the command is executed. The Netsh command script contains the configuration of the IAS database file (Ias.mdb) in a compressed text format as a large data block. This large data block is used by the **set config** command within the script to import the configuration of a saved data block into an existing IAS database on the same or another computer using the **netsh exec** command. To save the Netsh command script to a file, type the following at a command prompt: **netsh aaaa show config >** *Path\File.txt*
- The **show config** command is not supported on computers running Windows 2000 Server.

set config

Configures an IAS database with the information in the specified data block.

⇡Top of page

Syntax

set config [blob=] *DataBlock*

Top of page

Parameters

[blob=] ***DataBlock*** **:** Required. Specifies the configuration of the IAS database in a compressed text format as output by the **dump** or **show config** commands.

/?: Displays help at the command prompt.

Top of page

Remarks

- There is no reason to manually issue the **set config** command. It should be used only within a Netsh command script that is created by using the **dump** or **show config** commands.
- The version of the IAS database from which the data block was exported should match the version of the IAS database to which the block is being imported. To view the version of the IAS database on which the configuration was exported, view the Netsh command script file. To view the version of the IAS database on which the Netsh script is being executed, use the **show version** command.

show version

Displays the version of the IAS database.

Top of page

Syntax

show version

Top of page

Examples

To display the Netsh command script that reflects the IAS database in its current state, type:

show config

To display the version of the IAS database running on a server, type:

show version

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Netsh commands for DHCP

The Netsh commands for DHCP offer a command-line tool that helps with the administration of DHCP servers and provides an equivalent alternative to console-based management. This can be useful in the following situations:

- When managing DHCP servers in wide area networks (WANs), commands can be used in interactive mode at the Netsh command prompt to better manage across slow-speed network links.
- When managing a large number of DHCP servers, commands can be used in batch mode at the Netsh command prompt to help script and automate recurring administrative tasks that need to be performed for all DHCP servers.

The following commands are available at the Netsh command prompt (as distinguished from the Windows XP command prompt). You cannot run these Netsh commands locally on Windows XP Professional or Windows XP Home Edition. To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP. For more information about how to use the Netsh commands that are provided for DHCP, see Use DHCP command-line tools

For an example of how to use Netsh commands for DHCP in batch files and scripts, see Netsh DHCP example

For more information about Netsh commands, see The Netsh command-line utility

For more information about **netsh**, see Using Netsh

Netsh DHCP

The following commands are available at the **dhcp>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

add server

Adds a DHCP server to the list of authorized servers in Active Directory.

Syntax

add server *ServerDNS ServerIP*

Parameters

***ServerDNS* :** Required. Specifies the DHCP server to add. Identifies the server by DNS domain name.

***ServerIP* :** Required. Specifies the DHCP server to add. Identifies the server by IP address.

Examples

In the following example, this command adds a server with the DNS name dhcpsrv1.example.microsoft.com and IP address 10.2.2.2 to the list of authorized servers in Active Directory.

add server dhcpsrv1.example.microsoft.com 10.2.2.2

delete server

Deletes a DHCP server from the list of authorized servers in Active Directory.

Syntax

delete server *ServerDNS ServerIP*

Parameters

***ServerDNS* :** Required. Specifies the DHCP server to delete. Identifies the server by DNS domain name.

***ServerIP* :** Required. Specifies the DHCP server to delete. Identifies the server by IP address.

Examples

In the following example, this command deletes the server with the DNS name dhcpsrv1.example.

microsoft.com and IP address 10.2.2.2 from the list of authorized servers in Active Directory.

delete server dhcpsrv1.example.microsoft.com 10.2.2.2

dump

Dumps the configuration of a DHCP server to the command prompt window or to a text file. Used without parameters, this command dumps the configuration of the local server to the command prompt window.

Syntax

[{\\ServerName | *IPAddress*}] **dump** > [*FileName*]

Parameters

[{\\*ServerName* | *IPAddress*}] : Specifies the name of the remote (non-local) DHCP server from which you want to dump configuration output. Identifies the server by name or IP address. If unspecified, the configuration of the local server is dumped.

***FileName* :** Specifies the name of the destination file to which you want to dump DHCP server configuration. If unspecified, DHCP server configuration is dumped to the command prompt window.

Remarks

- This command dumps the DHCP configuration as text output within the command prompt window, or you can copy or redirect this output to a file for later viewing in a text editor. When file output is obtained, you can use the **exec** command to execute the output file in order to configure another DHCP server.
- Before performing the **dump** command at the source server, reconcile all scopes and fix any inconsistencies that are reported.
- The following command dumps the current configuration for the local DHCP server (the source) to a text file named *dhcpcfg.dmp*. This file can then be copied to another new DHCP server (the destination) and used to recreate the source configuration on the destination server:

  dump > dhcpcfg.dmp

- Before you use the **exec** command to execute a dump file at the destination server, do the following:

  1. Close the DHCP console at the destination server. Do not reopen the console until after you use the **exec** command.
  2. Delete all default option definitions and any user and vendor classes at the destination server.
- These examples demonstrate how to use the **delete optiondef** and **del class** Netsh commands to delete all default option definitions and any user and vendor classes. You can run the following commands at the **dhcp server>** prompt, which is rooted within the **netsh** environment.

  - **delete optiondef 01**
  - **delete optiondef 76**
  - **del class "Default Routing and Remote Access Class"**
  - **del class "Default BOOTP Class"**
  - **del class "Microsoft Windows 2000 Options"**
  - **del class "Microsoft Windows 98 Options"**
  - **del class "Microsoft Options"**
- On the destination server, the **exec** command is used to load and execute the saved configuration:

  netsh exec dhcpcfg.dmp
- After you use the **exec** command to load the dump file, reconcile all scopes again. Use **net stop dhcpserver** to stop the DHCP Server service and **net start dhcpserver** to restart it. Once the service is restarted, DHCP database changes take effect.

Examples

In the first example, this command dumps the configuration of DHCP server \\DHCP-A to a text file named dhcpcfg.dmp.

In the second example, this command dumps the configuration of DHCP server 10.0.0.1 to a text file named dhcpcfg.dmp.

In the third example, this command dumps the configuration of the local DHCP server to a text file named dhcpcfg.dmp.

\\DHCP-A dump > dhcpcfg.dmp

10.0.0.1 dump > dhcpcfg.dmp

dump > dhcpcfg.dmp

server

Shifts the current Netsh DHCP command-line context to a different DHCP server. Used without parameters, **server** shifts the current command-line context to the local computer.

Syntax

server [{\\ServerName | *ServerIP*}]

Parameters

{\\*ServerName* | *ServerIP*} : Specifies the DHCP server to which you want to shift the current command-line context. Identifies the server by NetBIOS name or IP address.

Examples

In the first example, this command changes the Netsh DHCP command-line context to the DHCP server named \\DHCP-SRV1.

In the second example, this command changes the Netsh DHCP command-line context to the DHCP server with IP address 10.0.0.1.

server \\DHCP-SRV1

server 10.0.0.1

show server

Displays a list of authorized servers in Active Directory.

Syntax

show server

Parameters

none

↑Top of page

Netsh DHCP server

The following commands are available at the **dhcp server>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

add class

Adds a class to the specified DHCP server.

Syntax

add class *ClassName* [*ClassComment*] [*Data*] [[**IsVendor=**]{**0** | **1**}] [[**IsBinary=**]{**B** | **b**}]

Parameters

ClassName **:** Required. Specifies the name of the class to create.

ClassComment **:** Specifies the comment to associate with the class.

Data **:** Specifies class data in ASCII or binary format depending on the **IsBinary** parameter. If unspecified, it is assumed that the format of the data is ASCII.

[IsVendor=]{0 | 1} : Indicates whether the class to add is a user class or a vendor class: **0**-User class (default), **1**-Vendor class.

[IsBinary=]{B | b} : Indicates if the class data is in binary format. If unspecified, it is assumed that the format of the data is ASCII.

Examples

In the first example, this command creates a vendor class named MyClass, with the comment TestClass, and the class data TestDataForMyClass in ASCII format.

In the second example, this command creates a user class named MyUserClass with binary data of 4345335532 (ASCII-CE3U2).

The third example is an alternate command syntax for the second example.

add class MyClass TestClass TestDataForMyClass 1

add class MyUserClass TestClass1 4345335532 B

add class MyUserClass TestClass1 4345335532 0 b

add mscope

Adds a multicast scope to the specified DHCP server.

Syntax

add mscope *MscopeName* [*MscopeComment*] [*TTL*]

Parameters

***MscopeName* :** Required. Specifies the name of the multicast scope to add.

***MscopeComment* :** Specifies a description of the current multicast scope.

***TTL* :** Specifies the Time-To-Live value for the multicast scope. The valid range for this field is 1 to 255, with a default of 32.

Examples

In the following example, this command creates a multicast scope with the name My MultiCast Scope, a comment of MyTestMulticastScope, and a multicast TTL value of 40.

add mscope "My MultiCast Scope" MyTestMulticastScope 40

add optiondef

Adds a new option type definition to the specified DHCP server.

Syntax

add optiondef *OptCode OptName* {**BYTE** | **WORD** | **DWORD** | **STRING** | **IPADDRESS**} [[**IsArray=**]{**0** | **1**}] [**vendor=***VendorClass*] [**comment=***OptComment*] [*DefValue*]

Parameters

***OptCode* :** Required. Specifies the option type to define. Identifies the option type with a unique option code. For the value of the code, choose a number between 0 and 255.

***OptName* :** Required. Specifies the name of the option type.

{BYTE | WORD | DWORD | STRING | IPADDRESS} : Required. Indicates the data type of the option.

[IsArray=]{0 | 1} : Indicates whether the data type is arrayed or non-arrayed: **0**-Unary or non-arrayed (default), **1**-Arrayed.

vendor=*VendorClass* : Specifies the vendor class for the option. If unspecified, the default vendor class, the DHCP standard options class, is used.

comment=*OptComment* : Specifies a comment to add to the new option type definition. If unspecified, the default value is NULL.

***DefValue* :** Specifies a default value to use for this option type, if one is not specified otherwise.

Remarks

- **IsArray** must be **0** for STRING type options.

Examples

In the first example, this command defines a new option type named ExtensionsPath with code 18, a STRING data type, and joins it to the DHCP standard options class, assigning it a default value of c:\temp.

In the second and third examples, this command defines a new option type named TestOption with code 231 and an IPADDRESS data type. In the second example, the option is defined to provide only a single IP address. IsArray and *DefValue* are not necessary because defaults are assumed. In the third example, *DefValue*, and IsArray are required to create an array containing the IP address values 10.1.1.1, 10.1.1.2, and 10.1.1.3.

add optiondef 18 ExtensionsPath STRING 0 c:\temp

add optiondef 231 TestOption IPADDRESS

add optiondef 231 TestOption IPADDRESS 1 vendor=MyVendorClass 10.1.1.1 10.1.1.2 10.1.1.3

add scope

Adds a scope to the specified DHCP server.

Syntax

add scope *ScopeAddress SubnetMask ScopeName* [*ScopeComment*]

Parameters

***ScopeAddress* :** Required. Specifies the scope to add. Identifies the scope by IP network number.

***SubnetMask* :** Required. Specifies the subnet mask for the scope to add.

***ScopeName* :** Required. Specifies an identifying name for the scope.

***ScopeComment* :** Specifies a comment for the scope.

Examples

In the following example, this command adds a scope for the DHCP server with scope address 10.2.2.0, subnet mask 255.255.255.0, a scope name of MyScope, and a scope comment of MyComment.

add scope 10.2.2.0 255.255.255.0 MyScope MyComment

delete class

Deletes a class from the specified DHCP server.

Syntax

delete class *Name*

Parameters

***Name* :** Required. Specifies the user or vendor class to delete.

Examples

In the following example, this command deletes the class named MyClass.

delete class MyClass

delete mscope

Deletes a multicast scope from the specified DHCP server.

Syntax

delete mscope *MscopeName*

Parameters

***MscopeName* :** Required. Specifies the multicast scope to delete. *MscopeName* is case-sensitive.

Examples

In the following example, this command deletes the multicast scope MyMulitcastScope from the currently specified DHCP server.

delete mscope MyMulitcastScope

delete optiondef

Deletes a defined option type from the specified DHCP server.

Syntax

delete optiondef *OptCode* [[**vendor=**]*VendorName*]

Parameters

***OptCode* :** Required. Specifies the unique identifier for the option type to delete.

vendor=*VendorName* : Specifies the vendor class name with which the option type is associated. If unspecified, the option type definition is deleted from the non-vendor specific DHCP standard options class.

Remarks

- This option is applicable only for DHCP servers running Windows 2000 Server.

Examples

In the following example, this command deletes an option type with code 18 from the DHCP standard options class.

delete optiondef 18

delete optionvalue

Deletes the current value of the specified option type for the specified DHCP server.

Syntax

delete optionvalue *OptCode* [[**vendor=**]*VendorClass*] [[**user=**]*UserClass*]

Parameters

OptCode **:** Required. Specifies the unique identifier of the global server option with a value to delete.

vendor=*VendorClass* **:** Specifies the vendor class name with which the option is associated. If the tag is not included, the global *VendorName* that was set by **set vendorclass** is assumed. If the tag is used, but no value is specified, then no vendor class is used. This parameter is available only for computers running Windows 2000 Server.

user=*UserClass* **:** Specifies the user class name with which the option is associated. If the tag is not included, the global *UserName* that was set by **set userclass** is assumed. If the tag is used, but no value is specified, then no user class is used. This parameter is available only for computers running Windows 2000 Server.

Examples

In the following example, this command deletes the current value of the option type with code 18.

delete optionvalue 18

delete scope

Deletes a scope from the specified DHCP server.

Syntax

delete scope *ScopeAddress* {**DHCPNOFORCE | DHCPFULLFORCE**}

Parameters

ScopeAddress **:** Required. Specifies the address of the scope to delete.

{DHCPNOFORCE | DHCPFULLFORCE} : Required. Indicates the force with which the scope will be deleted: **DHCPNOFORCE**-Deletes the scope if there are no active clients in the scope, **DHCPFULLFORCE-**Deletes the scope even if there are active clients in the scope.

Examples

In the following example, this command forces the deletion of the scope 10.2.2.0 from the DHCP server and ignores all warnings.

delete scope 10.2.2.0 dhcpfullforce

delete superscope

Deletes a superscope from the specified DHCP server.

Syntax

delete superscope *SuperscopeName*

Parameters

SuperscopeName **:** Required. Specifies the name of the superscope to delete.

Remarks

- You can set the superscope from the scope context with the **set superscope** command.
- The superscope name is case-sensitive.

Examples

In the following example, this command deletes the superscope named My Superscope from the specified DHCP server.

delete superscope "My Superscope"

export

Exports the DHCP service configuration to a file.

Syntax

export [*Path*]*FileName* {**all** | *ScopeList*}

Parameters

[*Path*]*FileName* : Required. Specifies, by name, the file where the DHCP configuration will be stored. If the path, the file name, or both contain spaces, quotation marks must be used.

{all | *ScopeList*} : Required. Specifies which scopes you want to export. The parameter **all** exports all scopes. The parameter *ScopeList* exports the scopes that correspond to the IP addresses you list. Each IP address in the list must be separated by spaces.

Remarks

- This command works only on the local server.
- While the **export** command runs, the DHCP service is stopped and does not respond to DHCP clients seeking new leases or lease renewals.
- If the DHCP service has a large number of scopes or a large number of client address leases, this command can take a long time to run.

Examples

In the first example, this command exports the complete DHCP service configuration to the file c:\temp\dhcpdb.

In the second example, this command exports the DHCP configuration for scopes 10.0.0.0 and 192.168.0.0 to the file c:\temp\dhcpdb.

In the third example, this command exports the complete DHCP service configuration to the file C:\My Folder\Dhcp Configuration. Note that both the path and file name contain spaces, so quotation marks are used.

export c:\temp\dhcpdb all

export c:\temp\dhcpdb 10.0.0.0 192.168.0.0

export "C:\My Folder\Dhcp Configuration" all

import

Imports a DHCP service configuration from a file to the local service.

Syntax

import [*Path*]*FileName* {**all** | *ScopeList*]

Parameters

[*Path*]*FileName* : Required. Specifies, by name, the file from which the DHCP configuration will be imported. If the path, the file name, or both contain spaces, quotation marks must be used.

{all | *ScopeList*} : Required. Specifies which scopes you want to import. The parameter **all** imports all scopes represented in the file you specify. The parameter *ScopeList* imports the scopes that correspond to the IP addresses you list. Each IP address in the list must be separate by spaces.

Remarks

- This command works only on the local server.
- While the **import** command runs, the DHCP service is stopped and does not respond to DHCP clients seeking new leases or lease renewals.
- If the DHCP service has a large number of scopes or a large number of client address leases, this command can take a long time to run.

Examples

In the first example, this command imports the complete DHCP service configuration from the file c:\temp\dhcpdb.

In the second example, this command imports the DHCP configuration for scopes 10.0.0.0 and 192.168.0.0 from the file c:\temp\dhcpdb

In the third example, this command imports the complete DHCP service configuration from the file C:\My Folder\Dhcp Configuration. Note that both the path and file name contain spaces, so quotation marks are used.

import c:\temp\dhcpdb all

import c:\temp\dhcpdb 10.0.0.0 192.168.0.0

import "C:\My Folder\Dhcp Configuration" all

initiate auth

Initiates authorization of the specified DHCP server in Active Directory.

Syntax

initiate auth

Parameters

none

mscope

Switches the command context to a DHCP multicast scope.

Syntax

mscope *MscopeName*

Parameters

***MscopeName* :** Required. Specifies the name of the multicast scope to which to switch the command context.

Remarks

- This command enables you to use any of the available DHCP multicast scope-level commands on the specified DHCP multicast scope. Subsequent operations will be performed on this multicast scope at the specified server.
- The name of the multicast scope is case-sensitive.

Examples

In the following example, this command switches the command context to a multicast scope named My Multicast Scope.

mscope "My Multicast Scope"

scope

Switches the command context to a DHCP scope.

Syntax

scope *ScopeIPAddress*

Parameters

***ScopeIPAddress* :** Required. Specifies the scope to which to switch the command context. Identifies the scope by network or subnet IP address.

Remarks

- This command enables you to use any of the available DHCP scope-level commands on the specified DHCP scope. Subsequent operations will be performed on this scope at the specified server.

Examples

In the following example, this command switches the command context to the scope with IP address 10.0.0.0.

scope 10.0.0.0

set auditlog

Sets the audit log path parameters for the specified DHCP server.

Syntax

set auditlog *NewAuditPath*

Parameters

NewAuditPath **:** Required. Specifies a local folder for the DHCP server to use when storing audit log files. A valid folder path is required, and the folder must be local to the DHCP server where the path is defined.

Remarks

- In order for this change to take effect, you need to stop and restart the DHCP service after you run this command.

Examples

In the following example, this command sets the DHCP server to store audit log files in the directory c:\logpath.

set auditlog c:\logpath

set databasebackupinterval

Sets the backup interval for the specified DHCP server.

Syntax

set databasebackupinterval *NewInterval*

Parameters

NewInterval **:** Required. Specifies the backup interval in minutes.

Examples

In the following example, this command sets the database backup interval for the specified DHCP server to 1,440 minutes (24 hours).

set databasebackupinterval 1440

set databasebackuppath

Sets the database backup path for the specified DHCP server.

Syntax

set databasebackuppath *NewBackupPath*

Parameters

NewBackupPath **:** Required. Specifies a local folder for the DHCP server to use when storing audit log files. A valid folder path is required, and the folder must be local to the DHCP server where the path is defined.

Examples

In the following example, this command sets the database backup path to c:\winnnt\dhcp\backup.

set databasebackuppath c:\winnt\dhcp\backup

set databasecleanupinterval

Sets the database cleanup interval of the specified DHCP server.

Syntax

set databasecleanupinterval *NewInterval*

Parameters

NewInterval **:** Required. Specifies the database cleanup interval, in minutes.

Examples

In the following example, this command sets the database cleanup interval to 10,080 minutes (every seven days).

set databasecleanupinterval 10080

set databaseloggingflag

Sets or resets the database logging flag for the specified DHCP server. Used without parameters, this command resets the database logging flag.

Syntax

set databaseloggingflag [{**0** | **1**}]

Parameters

{0 | 1} : Indicates the database logging flag setting: **0**-Resets the flag (default), **1**-Sets the flag.

Examples

In the following example, this command resets the database logging flag for the currently specified DHCP server.

set databaseloggingflag 0

set databasename

Sets the name of the DHCP server database file for the specified DHCP server.

Syntax

set databasename *NewFileName*

Parameters

***NewFileName* :** Required. Specifies a new name for the DHCP server database file.

Remarks

•In order for this change to take effect, you need to stop and restart the DHCP Server service.

Examples

In the following example, this command sets the DHCP server database file name to newdatabase. mdb.

set databasename newdatabase.mdb

set databasepath

Sets the path of the DHCP server database file for the specified DHCP server.

Syntax

set databasepath *NewPath*

Parameters

NewPath **:** Required. Specifies the path of the DHCP database file.

Remarks

- In order for this change to take effect, you need to stop and restart the DHCP Server service.

Examples

In the following example, this command sets the path of the DHCP server database file to c:\winnt \dhcp.

set databasepath c:\winnt\dhcp

set databaserestoreflag

Sets or resets the database restore flag for the specified DHCP server. Used without parameters, this command resets the database restore flag.

Syntax

set databaserestoreflag [{**0** | **1**}]

Parameters

{0 | 1} : Indicates the setting of the database restore flag: **0**-Resets the flag (default), **1**-Sets the flag.

Examples

In the following example, this command sets the database restore flag for the currently specified DHCP server.

set databaserestoreflag 1

set detectconflictretry

Sets the number of conflict detection attempts for the specified DHCP server.

Syntax

set detectconflictretry *RetryNumber*

Parameters

***RetryNumber* :** Required. Specifies the number of conflict detection attempts that the DHCP server will make before it leases a scope IP address to a client.

Remarks

- The maximum value for *RetryNumber* is five. All values greater than five will default down to five.

Examples

In the following example, this command sets the DHCP server to make four attempts to detect if a there is a conflict for a scope IP address before the DHCP server uses this scope IP address in a new address lease.

set detectconflictretry 4

set dnsconfig

Sets the DNS dynamic update configuration for the specified DHCP server.

Syntax

set dnsconfig [**Enable=**]{**0**|**1**} [[**Update=**]{**0**|**1**}] [[**Lookup=**]{**0**|**1**}] [[**NonDyn=**]{**0**|**1**}]

Parameters

[Enable=]{0|1} : Required. Indicates whether to enable or disable the dynamic updates of DNS client information: **0**-Disables dynamic updates of DNS client information, **1**-Enables dynamic updates of DNS client information.

[Update=]{0|1} : Indicates the type of update lookups: **0**-Updates according to client request, **1**-Updates forward and reverse name lookups.

[Lookup=]{0|1} : Indicates whether to enable or disable forward lookups for clients when leases expire: **0**-Disables forward lookups, **1**-Enables forward lookups.

[NonDyn=]{0|1} : Indicates whether to update DNS information for clients that do not support dynamic updates: **0**-Disables updates for clients that do not support dynamic updates, **1**-Enables these updates.

Remarks

- This command is only supported for DHCP servers running Windows 2000 Server.
- The configured Preferred DNS server, in the TCP/IP properties of the DHCP server, will be updated with DHCP client records when dynamic updates are enabled. If the Preferred DNS server is not available on the network, the Alternate DNS server will be updated.

Examples

In the following example, this command enables the dynamic update of DNS information by the DHCP server and configures updates to always be made for both forward and reverse lookups. This example also disables the option to perform forward lookups when leases expire and enables updates for non-dynamic clients.

set dnsconfig 1 1 0 1

set optionvalue

Sets a DHCP server option value that will be applied for all scopes defined at the specified DHCP server.

Syntax

set optionvalue *OptCode* {**BYTE** | **WORD** | **DWORD** | **STRING** | **IPADDRESS**} [[**user=**] *UserName*] [[**vendor=**]*VendorName*] [*OptionValue*]

Parameters

OptCode **:** Required. Specifies the unique identifier for the option type with the value to set.

{BYTE | WORD | DWORD | STRING | IPADDRESS} : Required. Specifies the data type for the option type with the value to set.

[user=]*UserName* : Sets the applicable user class name to which this command will apply. If unspecified, the default user class is assumed.

[vendor=]*VendorName* : Sets the applicable vendor class name to which this command will apply. If unspecified, the DHCP standard options class is assumed.

[*OptionValue*] : Specifies the new server default value for the option type identified by *OptCode*. The value must be of the associated data type.

Remarks

- Vendor and user classes are supported only for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class, use either **set userclass** or **set vendorclass** commands.

Examples

In the first example, this command sets the value of option code 003 to list two router IP addresses, 10.1.1.1 and 10.1.1.2.

In the second example, this command further qualifies the first example and sets the specified value to apply only for those scope clients that identify themselves as members of the vendor class Vendor1. In order for this example to work, this class must already be defined at the server, with the specified option type defined for its use.

set optionvalue 003 IPADDRESS 10.1.1.1 10.1.1.2

set optionvalue 003 IPADDRESS vendor=Vendor1 10.1.1.1 10.1.1.2

set server

Changes the Netsh command-line context to the specified server. Used without parameters, the local server is assumed.

Syntax

set server [{*ServerIP* | \\ServerDNS}]

Parameters

{*ServerIP* | \\*ServerDNS*} : Specifies the IP address or the DNS name of the specified DHCP server.

Remarks

- When this command is used, and the Netsh command-line context is changed to another server, all subsequent command operations are performed on the specified server. To perform command operations on another server, this command must be executed again, with the IP address or DNS name of the new server provided as the parameter.

Examples

In the following example, this command switches the DHCP server command-line context to the server at IP address 10.1.1.1

set server 10.1.1.1

set userclass

Sets the name for the current user class. Used without parameters, the current user class is reset to the default user class

Syntax

set userclass [*UserClass*]

Parameters

***UserClass* :** Specifies the name of the user class for the currently specified DHCP server.

Remarks

- This command is available only for use with DHCP servers running Windows 2000 Server.

Examples

In the first example, this command sets the user class name to MyUserClass.

In the second example, this command resets the current user class to the default user class.

set userclass MyUserClass

set userclass

set vendorclass

Sets the name for the current vendor class. Used without parameters, the current vendor class is reset to the default class, DHCP standard options.

Syntax

set vendorclass [*VendorClass*]

Parameters

***VendorClass* :** Specifies the name of the vendor class for the current server.

Remarks

- This command is available only for use with DHCP servers running Windows 2000 Server.

Examples

In the first example, this command sets the vendor class name to MyVendorClass.

In the second example, this command resets the current vendor class to the default class, DHCP standard options.

set vendorclass MyVendorClass

set vendorclass

show all

Displays all status and configuration information for the specified DHCP server.

Syntax

show all

Parameters

none

Remarks

- This command lists all current server status and configuration details, including the stored management information base (MIB) information, for the currently specified DHCP server.

show auditlog

Displays all audit log information for the specified DHCP server.

Syntax

show auditlog

Parameters

none

show bindings

Displays bindings information for the specified DHCP server.

Syntax

show bindings

Parameters

none

Remarks

- This command displays bindings information for the current DHCP server, indicating which active network connections that are configured with static IP addresses are enabled or disabled for use in servicing DHCP clients on your network.
- The DHCP service automatically disables network connections that obtain their IP address configuration dynamically from service bindings. This happens by default.

show class

Enumerates and displays all class information for the specified DHCP server. This includes both user and vendor class information.

Syntax

show class

Parameters

none

show detectconflictretry

Displays the configured number of conflict detection attempts for the specified DHCP server.

Syntax

show detectconflictretry

Parameters

none

Remarks

- This command displays the current number of ping retries that are used to attempt the detection of address conflicts for the scope IP addresses that are distributed by the server.

show dnsconfig

Displays the DNS dynamic update configuration for the specified DHCP server.

Syntax

show dnsconfig

Parameters

none

Remarks

- If DNS configuration has not been enabled or set, you can use the DNS console or the **set dnsconfig** command to configure these settings.

show mibinfo

Displays management information base (MIB) information for the specified DHCP server.

Syntax

show mibinfo

Parameters

none

show mscope

Displays all information about multicast scopes for the specified DHCP server.

Syntax

show mscope

Parameters

none

show optiondef

Displays all defined and available options types for use at the specified DHCP server. Used without parameters, all options are displayed.

Syntax

show optiondef [**vendor=***VendorName*]

Parameters

vendor=*VendorName* : Indicates which defined and available option types to display. If unspecified, the default for *VendorName* is the default vendor class that is currently set for the DHCP server.

Remarks

- The default vendor class can be reset with the **set vendorclass** command.
- *VendorName* is applicable only for DHCP servers running Windows 2000 Server.

Examples

In the first example, this command displays all defined option types for the current vendor class.

In the second example, this command displays the defined option types that are available for use with the specified class Vendor1.

show optiondef

show optiondef vendor=Vendor1

show optionvalue

Displays all of the available option values that are currently set for the specified DHCP server. Used without parameters, all option values are displayed, including those that are used for both user and vendor classes.

Syntax

show optionvalue [**user=***UserName* | **vendor=***VendorName*]

Parameters

user=*UserName* : Specifies the default values that are set for the option types that are available for use with the currently set user class.

vendor=*VendorName* : Specifies the default values that are set for the option types that are available for use with the currently set vendor class.

Remarks

- **User=***UserName* and **vendor=***VendorName* are only available for use with DHCP servers running Windows 2000 Server.
- If you provide values for *UserName* or *VendorName*, the respective tags (**user=** or **vendor=**) are required. If the tags are not included, the default that is assumed is the current user class, previously set with the **set userclass** command.

Examples

In the following example, this command displays all of the available option types that have values set at the currently specified DHCP server.

show optionvalue

show scope

Displays information about the scopes for the specified DHCP server.

Syntax

show scope

Parameters

none

show server

Displays information about the specified DHCP server, including the server's fully qualified domain name and IP address.

Syntax

show server

Parameters

none

show dbproperties

Displays information about server database configuration for the specified DHCP server.

Syntax

show dbproperties

Parameters

none

Remarks

- This command displays the following database configuration information:

  - DatabaseName
  - DatabasePath
  - DatabaseBackupPath
  - DatabaseBackupInterval
  - DatabaseLoggingFlag
  - DatabaseRestoreFlag
  - DatabaseCleanupInterval

show serverstatus

Displays status information for the specified DHCP server.

Syntax

show serverstatus

Parameters

none

Remarks

- This is an example of the type of status information that this command displays:

  Server Attrib - Server Servicing Clients :TRUE

  Server Attrib - Dynamic BootP Support Enabled :TRUE

  Server Attrib - DHCP Server Part Of DS :TRUE

  Server Attrib - DHCP Server Bindings Aware :TRUE

  Server Attrib - Administrative Rights :TRUE

show userclass

Displays the current user class setting at the specified DHCP server.

Syntax

show userclass

Parameters

none

Remarks

- This is an example of the type of setting information that this command displays:

  Current Class Name set for the Server dhcpsrv1.example.microsoft.com is None.

show vendorclass

Displays the current vendor class setting at the specified DHCP server.

Syntax

show vendorclass

Parameters

none

Remarks

- This is an example of the type of setting information that this command displays:

  Current Vendor Name set for the Server dhcpsrv1.example.microsoft.com is None.

show version

Displays current version information for the specified DHCP server.

Syntax

show version

Parameters

none

Remarks

- This is an example of the type of version information that this command displays:

  The version of the DHCP Server 192.168.0.100 is 5.6.

⇡Top of page

Netsh DHCP server scope

The following commands are available at the **dhcp server scope>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

add excluderange

Adds a range of addresses to exclude from distribution in the current scope.

Syntax

add excluderange *StartIP EndIP*

Parameters

StartIP **:** Required. Specifies the IP address that starts the exclusion range.

EndIP **:** Required. Specifies the IP address that ends the exclusion range.

Examples

In the following example, this command excludes the IP addresses in the range 10.2.2.10 to 10.2.2.20 from distribution in the scope.

add excluderange 10.2.2.10 10.2.2.20

add iprange

Adds a range of IP addresses to the current scope.

Syntax

add iprange *StartIP EndIP* [{**DHCP** | **BOOTP** | **BOTH**}] [*MaxBootP*]

Parameters

***StartIP* :** Required. Specifies the IP address that starts the range.

***EndIP* :** Required. Specifies the IP address that ends the range.

{DHCP | BOOTP | BOTH} : Specifies the type of client to be serviced by this scope and IP range. **DHCP** is the default.

***MaxBootP* :** Specifies the maximum number of BOOTP clients.

Remarks

- *MaxBootP* is useful only if *ClientType* is set to support BOOTP-type clients for the scope. If an optional command-line option is used, all command-line options (mandatory as well as optional) that come before the optional command that is used are required and must appear in proper sequence.

Examples

In the following example, this command adds DHCP clients that are in the IP address range 10.2.2.10 to 10.2.2.20 to the distribution range in the scope.

add iprange 10.2.2.10 10.2.2.20

add reservedip

Reserves an IP address for use by a specified media access control (MAC) address in the current scope.

Syntax

add reservedip *ReservedIP MACAddress* [*ClientName*] [*ClientComment*] [{**DHCP** | **BOOTP** | **BOTH**}]

Parameters

ReservedIP **:** Required. Specifies the IP address to reserve.

MACAddress **:** Required. Specifies the physical hardware or MAC address to associate with the reserved IP address.

ClientName **:** Specifies a client name to associate with this reserved client entry. If unspecified, a client name is not associated with this reserved client entry.

ClientComment **:** Specifies a client comment to associate with this reserved client entry. If unspecified, a client comment is not associated with this reserved client entry.

{DHCP | BOOTP | BOTH} : Specifies the type of clients to associate with this reserved client entry. **DHCP** is the default.

Remarks

- If an optional command-line option is used, all command-line options (mandatory as well as optional) that come before the optional command-line option that is used are required and must appear in proper sequence.

Examples

In the following example, this command reserves the IP address 10.2.2.32 for use by the DHCP client that identifies its MAC address as 08-00-2b-30-36-9b when it obtains a lease in the current scope.

add reservedip 10.2.2.32 08002b30369B

delete excluderange

Deletes a range of previously excluded IP addresses from the current scope.

Syntax

delete excluderange *StartIP EndIP*

Parameters

***StartIP* :** Required. Specifies the IP address that starts the exclusion range.

***EndIP* :** Required. Specifies the IP address that ends the exclusion range.

Examples

In the following example, this command deletes the exclusion range that starts with an IP address of 10.2.1.5 and ends with an IP address of 10.2.1.10 from the current scope.

delete excluderange 10.2.1.5 10.2.1.10

delete iprange

Deletes a range of IP addresses from the current scope.

Syntax

delete iprange *StartIP EndIP*

Parameters

***StartIP* :** Required. Specifies the IP address that starts the range to delete.

***EndIP* :** Required. Specifies the IP address that ends the range to delete.

Examples

In the following example, this command deletes IP addresses in the range 10.2.1.5 to 10.2.1.10 from the current scope.

delete iprange 10.2.1.5 10.2.1.10

delete optionvalue

Removes or clears the currently set scope option value from the current scope.

Syntax

delete optionvalue *OptCode* [**user=***UserName*] [**vendor=***VendorName*]

Parameters

***OptCode* :** Required. Specifies the unique identifier of the option whose value is to delete.

user=*UserName* : Specifies the user class from which to delete the currently set option value. If the tag is provided, but the value unspecified, the current default user class is assumed.

vendor=*VendorName* : Specifies the vendor class from which to delete the currently set option value. If the tag is provided, but the value unspecified, the current default vendor class is assumed.

Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- If you specify a user class, a vendor class, or both, this command deletes the set option value from only the scope clients that are identified as members of the specified class or classes.
- To modify the current defaults for an unspecified class that is used with this command, use either the **set userclass** or **set vendorclass** commands.

Examples

In the first example, this command deletes the currently set option value for option code 18 from the current scope.

In the second example, this command deletes the currently set option value for option code 18 from the vendor class MyVendorClass.

delete optionvalue 18

delete optionvalue 18 vendor="MyVendorClass"

delete reservedip

Deletes a reservation for an IP address in the current scope.

Syntax

delete reservedip *ReservedIP MACAddress*

Parameters

***ReservedIP* :** Required. Specifies the IP address reservation to delete from the current scope.

***MACAddress* :** Required. Specifies the media access control (MAC) or physical hardware address string for which the IP address was reserved.

Examples

In the following example, this command removes the IP address reservation 10.2.1.32 for the MAC address 08002B30369B from the current scope.

delete reservedip 10.2.1.32 08002B30369B

delete reservedoptionvalue

Deletes an option value that is currently assigned to a reserved client in the current scope.

Syntax

delete reservedoptionvalue *ReservedIP OptCode* [**User=***UserName*] [**vendor=***VendorName*]

Parameters

***ReservedIP* :** Required. Specifies the IP address of the reserved client.

***OptCode* :** Required. Specifies the unique code for the option type that is currently assigned to the reserved client.

User=*UserName* : Specifies the user class from which to delete the option value. If the tag is provided, but no value is specified, the current default user class is assumed.

vendor=*VendorName* : Specifies the vendor class from which to delete the option value. If the tag is provided, but no value is specified, the current vendor class is assumed.

Remarks

•Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.

- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

Examples

In the following example, this command deletes the currently set option value for the option identified by code 18 for reserved IP address of 10.2.2.32 in the current scope.

delete reservedoptionvalue 10.2.2.32 18

initiate reconcile

Checks and reconciles the current scope. Used without parameters, this command verifies the scopes and check for inconsistencies but does not fix any inconsistencies that it finds in the database.

Syntax

initiate reconcile [**fix**]

Parameters

[fix] : Indicates that the command will fix, if possible, any inconsistencies that it finds in the database.

Examples

In the first example, this command verifies the scopes and checks for inconsistencies.

In the second example, this command verifies the scopes, checks for inconsistencies, and fixes any inconsistencies that are found.

initiate reconcile

initiate reconcile fix

set comment

Sets the comment for the current scope. Used without parameters, **set comment** deletes the current comment.

Syntax

set comment [*NewComment*]

Parameters

[*NewComment*] : Specifies a new or modified comment string for the scope.

Examples

In the first example, this command sets a comment string that has no spaces.

In the second example, this command sets a comment string that includes spaces.

set comment NewCommentNoSpaces

set comment "New Comment With Spaces"

set name

Sets the name of the current scope.

Syntax

set name *NewName*

Parameters

***NewName* :** Required. Specifies the new name of the scope.

Examples

In the first example, this command sets a name that has no spaces.

In the second example, this command sets a name that includes spaces.

set name NewNameNoSpaces

set name "New Name With Spaces"

set optionvalue

Sets an option value for the current scope.

Syntax

set optionvalue *OptCode* {**BYTE** | **WORD** | **DWORD** | **STRING** | **IPADDRESS**} [**user=***UserName*] [**vendor=***VendorName*] *OptionValue*

Parameters

***OptCode* :** Required. Specifies the code for the option type whose value is to be set.

{BYTE | WORD | DWORD | STRING | IPADDRESS} : Required. Specifies the data type for the option type whose value is to be set.

user=*UserName* : Specifies the user class. If the tag is provided, but the value is unspecified, the current default user class is assumed.

vendor=*VendorName* : Specifies the vendor class. If the tag is provided, but the value is unspecified, the current default vendor class is assumed.

***OptionValue* :** Required. Specifies the assigned value for the option type that is specified in *OptCode*. If the option type supports an array that contains more than a single numeric or IP address value, provide the additional values, in the order that you prefer them, at the end of the command, with each value separated by a space.

Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

Examples

In the first example, this command sets the value of option code 003 to list two router IP addresses (10.1.1.1, 10.1.1.2).

In the second example, this command sets the value that is specified in the first example to apply only to those scope clients that identify themselves as members of the vendor class Vendor1, a class that was previously defined at the server with this specified option type defined for its use.

set optionvalue 003 IPADDRESS 10.1.1.1 10.1.1.2

set optionvalue 003 IPADDRESS vendor=Vendor1 10.1.1.1 10.1.1.2

set reservedoptionvalue

Sets the value of an option for a reservation IP address in the current scope.

Syntax

set reservedoptionvalue *ReservedIP OptCode* {**BYTE** | **WORD** | **DWORD** | **STRING** | **IPADDRESS**} [**user=***UserName*] [**vendor=***VendorName*] *OptValue*

Parameters

ReservedIP **:** Required. Specifies the reserved IP address for which the option value is to be set.

OptCode **:** Required. Specifies the code for the option type whose value is to be set.

{BYTE | WORD | DWORD | STRING | IPADDRESS} : Required. Specifies the data type for the option type whose value is to be set.

user=*UserName* : Specifies either the current default user class or the class specified as *UserName*. If the tag is provided, but no value is specified, the current default user class is assumed.

vendor=*VendorName* : Specifies either the current default vendor class or the class specified as *VendorName*. If the tag is provided, but no value is specified, the current default vendor class is assumed.

OptValue **:** Required. Specifies the assigned value for the option type specified in *OptCode*. If the option type supports an array that contains more than a single numeric or IP address value, provide the additional values, in the order that you prefer them, at the end of the command with each value separated by a space.

Remarks

•Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.

- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

Examples

In the following example, this command sets the value of the router option (code 003) for the reserved client IP address of 10.1.1.50 in the current scope to set IP addresses of 10.1.1.1 and 10.1.1.2 for its configured routers (default gateways).

set reservedoptionvalue 10.1.1.50 003 IPADDRESS 10.1.1.1 10.1.1.2

set scope

Sets the scope for use in subsequent operations.

Syntax

set scope *ScopeAddress*

Parameters

ScopeAddress **:** Required. Specifies the IP address of the scope to use in subsequent command operations.

Examples

In the following example, this command sets the current scope to 10.2.2.0 for subsequent operations.

set scope 10.2.2.0

set state

Sets or resets the state of the current scope to either the active or inactive state. Used without parameters, this command activates the scope.

Syntax

set state [{**0** | **1** | **2** | **3**}]

Parameters

{0 | 1 | 2 | 3} : Indicates the state of the scope: **0**-Deactivates the scope, **1**-Activates the scope (default), **2**-Deactivates the scope and marks the scope as "Switched," **3**-Activates the scope and marks the scope as "Switched."

Remarks

- **2** and **3** are typically used for switched networks or networks where multiple logical networks are hosted on a single physical network.

Examples

In the first example, this command activates a scope.

In the second example, this command deactivates a scope.

set state 1

set state 0

set superscope

Sets the superscope to use in subsequent operations.

Syntax

set superscope *SuperscopeName* {**0** | **1**}

Parameters

SuperscopeName **:** Required. Specifies the name of the superscope to include the current scope. The *SuperscopeName* is case-sensitive.

{0 | 1} : Required. Indicates whether to set the state of the superscope to active or inactive: **0**-Deactivates the superscope, **1**-Activates the superscope.

Examples

In the following example, this command adds the current scope to the superscope MySuperScope and activates the superscope.

set superscope MySuperScope 1

show clients

Displays all of the available version 4 clients for the current scope. Used without parameters, this command displays the following information for each client: IP address, subnet mask, unique ID, lease expiration, and type.

Syntax

show clients [{**0** | **1**}]

Parameters

{0 | 1} : Indicates the detail level of the output: **0**-Shows the following information for each client: IP address, subnet mask, unique ID, lease expiration, and data type (default), **1**-Shows all of the information that **0** provides and also displays the fully qualified domain name of each client.

Remarks

- To view the output of this command effectively, increase the width of the command prompt window to at least 95 characters.

show clientsv5

Displays all of the available version 5 clients for the current scope. Used without parameters, this command displays the following information for each client: IP address, subnet mask, unique ID, lease expiration, and type.

Syntax

show clientsv5 [{**0** | **1**}]

Parameters

{0 | 1} : Indicates the detail level of the output: **0**-Shows the following information for each client: IP address, subnet mask, unique ID, lease expires, type (default), **1**-Shows all of the information that **0** provides and also displays the fully qualified domain name of each client.

Remarks

- To view the output of this command effectively, increase the width of the command prompt window to at least 95 characters.

show excluderange

Displays all of the currently set exclusion ranges of IP addresses for the current scope.

Syntax

show excluderange

Parameters

none

Remarks

- Use other scope-level commands to add and delete exclusion ranges.

show iprange

Displays all of the address ranges that are available for the current scope.

Syntax

show iprange

Parameters

none

show optionvalue

Displays all of the option values that are set for the current scope. Used without parameters, this command assumes the current default user and vendor classes.

Syntax

show optionvalue [**user=***UserName*] [**vendor=***VendorName*]

Parameters

user=*UserName* : Specifies that the options that are set for the specified user class will display. If the tag is provided, but no value is specified, the current default user class is assumed.

vendor=*VendorName* : Specifies that the options that are set for the specified vendor class will display. If the tag is provided, but no value is specified, the current default vendor class is assumed.

Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

Examples

In the following example, this command displays all options and values set for the current scope for the user defined class My User Class.

show optionvalue user="My User Class"

show reservedip

Displays all of the IP addresses that are currently reserved for the current scope.

Syntax

show reservedip

Parameters

none

show reservedoptionvalue

Displays all currently set option values for a reserved client IP address in the current scope.

Syntax

show reservedoptionvalue *ReservedIP* [**user=***UserName*] [**vendor=***VendorName*]

Parameters

ReservedIP **:** Required. Specifies the IP address reservation for which currently assigned options are to display.

user=*UserName* **:** Specifies that the options that are set for the specified user class will display. If the tag is provided, but no value is specified, the current default user class is assumed.

vendor=*VendorName* **:** Specifies that the options that are set for the specified vendor class will display. If the tag is provided, but no value is specified, the current default vendor class is assumed.

Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

Examples

In the following example, this command displays the option values set for the reserved IP address 10.2.2.100 in the current scope.

show reservedoptionvalue 10.2.2.100

show scope

Displays information for the current scope.

Syntax

show scope

Parameters

none

show state

Displays the state of the current scope, indicating whether it is active or inactive.

Syntax

show state

Parameters

none

⇡Top of page

Netsh DHCP server mscope

The following commands can run from the **dhcp server mscope>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

add excluderange

Adds a range of excluded addresses to the current multicast scope.

Syntax

add excluderange *StartIP EndIP*

Parameters

***StartIP* :** Required. Specifies the IP address that starts the exclusion range.

***EndIP* :** Required. Specifies the IP address that ends the exclusion range.

Remarks

- The exclusion range must be a subset of the overall scope address range.

Examples

In the following example, this command adds an exclusion range that starts with 224.2.2.10 and ends with 224.2.2.20 to the distribution range of the current multicast scope.

add excluderange 224.2.2.10 224.2.2.20

add iprange

Adds a range of IP addresses to the current multicast scope.

Syntax

add iprange *StartIP EndIP*

Parameters

***StartIP* :** Required. Specifies the IP address that starts the range.

***EndIP* :** Required. Specifies the IP address that ends the range.

Remarks

- The range must be within the valid range of multicast IP addresses (from 224.0.0.0 to 239.255.255.255).

Examples

In the following example, this command adds the IP address range 224.2.2.10 to 224.2.2.20 to the distribution range in the multicast scope.

add iprange 224.2.2.10 224.2.2.20

delete excluderange

Deletes an exclusion range of previously excluded IP addresses in the current multicast scope.

Syntax

delete excluderange *StartIP EndIP*

Parameters

***StartIP* :** Required. Specifies the IP address that starts the exclusion range.

***EndIP* :** Required. Specifies the IP address that ends the exclusion range.

Examples

In the following example, this command removes a multicast scope exclusion with a range of IP addresses that starts at 224.2.2.10 and ends with 224.2.2.20 for the multicast scope.

delete excluderange 224.2.2.10 224.2.2.20

delete iprange

Deletes a range of IP addresses from the current multicast scope.

Syntax

delete iprange *StartIP EndIP*

Parameters

***StartIP* :** Required. Specifies the IP address that starts the range to delete.

***EndIP* :** Required. Specifies the IP address that ends the range to delete.

Examples

In the following example, this command deletes the IP address range 224.2.2.10 to 224.2.2.20 from the overall range of the full multicast scope.

delete iprange 224.2.2.10 224.2.2.20

initiate reconcile

Checks and reconciles the current multicast scope.

Syntax

initiate reconcile

Parameters

none

Remarks

- This command checks the integrity of the current multicast scope by comparing the current contents of the server database with a mirrored copy of the same information in the Windows registry. If inconsistencies are detected in the database, they are repaired based on the information that is duplicated in the registry. Repair is always attempted for any inconsistency that is found.

set comment

Sets the comment for the current multicast scope.

Syntax

set comment *NewComment*

Parameters

NewComment **:** Required. Specifies the new or revised comment for the multicast scope

Examples

In the first example, this command modifies the multicast scope comment with a new comment that contains no spaces.

In the second example, this command modifies the multicast scope comment with a new comment that contains spaces.

set comment NewCommentNoSpaces

set comment "New Comment With Spaces"

set lease

Sets the lease duration for the multicast scope IP addresses.

Syntax

set lease *Time*

Parameters

***Time* :** Required. Specifies the lease duration for clients of the multicast scope. Specifying **-1** sets the duration of the IP address lease to an unlimited or infinite time.

Examples

In the following example, this command sets the lease duration for the clients of the current multicast scope to 691200 seconds (eight days).

set lease 691200

set mscope

Sets the multicast scope to use in subsequent operations.

Syntax

set mscope *NewName*

Parameters

***NewName* :** Required. Specifies the name of the multicast scope to which the command context is changed. *NewName* is case-sensitive.

Remarks

- This command changes the context of the **netsh dhcp server mscope>** prompt from one multicast scope to another.

- This command does not rename the current multicast scope. To change the name of the current multicast scope, use the command **set name**.

Examples

In the first example, this command changes the command context to a multicast scope named MyMulticastScope. Note that the multicast scope name contains no spaces.

In the second example, this command changes the command context to a multicast scope named My Multicast Scope. Note that the multicast scope name contains spaces.

set mscope MyMulticastScope

set mscope "My Multicast Scope"

set name

Changes the name of the current multicast scope.

Syntax

set name *NewName*

Parameters

***NewName* :** Required. Specifies a new name for the current multicast scope.

Examples

In the first example, this command sets a new name that contains no spaces for the current multicast scope.

In the second example, this command sets a new name that contains spaces for the current multicast scope.

set name NewNameNoSpaces

set name "New Name With Spaces"

set state

Sets or resets the state of the current multicast scope to either an active or inactive state.

Syntax

set state {**0** | **1**}

Parameters

{0 | 1} : Required. Sets the state of the current multicast scope: **0**-Deactivates the current multicast scope, **1**-Activates the current multicast scope .

Examples

In the first example, this command activates the current multicast scope.

In the second example, this command deactivates the current multicast scope.

set state 1

set state 0

set ttl

Sets the Time-To-Live (TTL) value for the current multicast scope.

Syntax

set ttl *TTL*

Parameters

TTL **:** Required. Specifies the Time-to-Live (TTL) value. The valid range for this value is a number from 1 to 255.

Examples

In the following example, this command sets the TTL value for the current multicast scope to 32.

set ttl 32

show clients

Displays all available clients for the current multicast scope.

Syntax

show clients

Parameters

none

Remarks

- To view the output of this command effectively, increase the width of the command prompt window to at least 95 characters.

show excluderange

Displays all currently excluded ranges of IP addresses for the current multicast scope.

Syntax

show excluderange

Parameters

none

Remarks

- If no exclusion ranges have been previously defined for the scope, this command outputs an empty list.

show iprange

Displays all available IP address ranges for the current multicast scope.

Syntax

show iprange

Parameters

none

show lease

Displays the current lease duration settings for the current multicast scope.

Syntax

show lease

Parameters

none

show mibinfo

Displays management information base (MIB) information for the current multicast scope.

Syntax

show mibinfo

Parameters

none

show mscope

Displays information for the current multicast scope.

Syntax

show mscope

Parameters

none

show state

Displays the state of the current multicast scope.

Syntax

show state

Parameters

none

show ttl

Displays the Time-To-Live (TTL) value for the current multicast scope.

Syntax

show ttl

Parameters

none

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Netsh diagnostic (diag) commands

You can use the Netsh Network Diagnostics commands, or diag context, to administer and troubleshoot operating system and network service parameters from the command line. The command prompt for the Netsh diagnostic context is **netsh diag>**.

The following commands are available at the command prompt for the Netsh diagnostic context (as distinguished from the Windows XP command prompt). The Netsh Diag context is new to Windows XP. You cannot run Netsh Diag commands remotely on a Windows 2000 Server.

For more information about **netsh**, see Using Netsh

To view the command syntax, click a command:

connect ieproxy

Establishes, verifies, and then drops a TCP/IP connection with the proxy server specified in Internet Explorer.

Syntax

connect ieproxy

Top of page

Parameters

none

Top of page

Remarks

- If no proxy server is specified in Internet Explorer, the command generates output similar to the following:

```
Internet Explorer Proxy (Not Configured)
    Not Configured
```

- If you have specified in Internet Explorer a proxy server named corpproxy with a port of 80, the command generates output similar to the following:

```
Internet Explorer Proxy (corpproxy)
    IEProxyPort = 80
    IEProxy = corpproxy
    Server appears to be running on port(s) [80]
```

connect iphost

Establishes, verifies, and then drops a TCP/IP connection to a specified host through a specified port.

Top of page

Syntax

connect iphost *Host Port*

Top of page

Parameters

Host **:** Specifies the IP address, NetBIOS name, or fully qualified domain name of the host to which you want to connect.

Port **:** Specifies the port through which you want to connect.

Top of page

Examples

To connect to the host named microsoft.com through port 28, type:

connect iphost microsoft.com 28

connect mail

Establishes, verifies, and then drops a TCP/IP connection with the node specified as the mail server in the default Outlook Express profile of the current user.

Top of page

Syntax

connect mail

Top of page

Parameters

none

connect news

Establishes, verifies, and then drops a TCP/IP connection with the node specified as the news server in the default Outlook Express profile of the current user.

Top of page

Syntax

connect news

Top of page

Parameters

none

dump

Creates a script that contains the current configuration.

Top of page

Syntax

dump

Top of page

Parameters

none

Top of page

Remarks

- If you save this script to a file, you can use it to restore altered configuration settings.

gui

Starts the network diagnostics tool in Help and Support Center.

↑Top of page

Syntax

gui

↑Top of page

Parameters

none

ping

Verifies connectivity with a specified remote host.

↑Top of page

Syntax

ping *Host*

↑Top of page

Parameters

Host **:** Specifies the IP address, NetBIOS name, or fully qualified domain name of the host with which you want to verify connectivity.

ping adapter

Verifies connectivity with computers and other devices (for example, routers) identified in the TCP/IP properties of the specified network adapter or adapters. Used without parameters, the **ping adapter** command verifies connectivity through all installed adapters.

Top of page

Syntax

ping adapter [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to verify connectivity. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

Top of page

Remarks

- This command verifies connectivity with all computers and other devices configured in the TCP/IP properties for each network adapter you specify. For example, if you have a network adapter whose TCP/IP properties are configured with IP addresses for a preferred DNS server, a default gateway, and a DHCP server, **ping adapter** verifies connectivity with each of these devices.
- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To verify connectivity with all computers and other devices configured in the TCP/IP properties for each network adapter, type:

ping adapter

To verify connectivity with all computers and other devices configured in the TCP/IP properties for each network adapter with an index number of 2, type:

ping adapter 2

To verify connectivity with all computers and other devices configured in the TCP/IP properties for each network adapter with "net" at the beginning of its name, type:

ping adapter net*

ping dhcp

Verifies connectivity with the Dynamic Host Configuration Protocol (DHCP) servers configured in the TCP/IP properties for the specified adapter or adapters. Used without parameters, the **ping dhcp** command verifies connectivity with DHCP servers through all installed adapters.

Top of page

Syntax

ping dhcp [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to verify connectivity. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To verify connectivity with the DHCP server configured in the TCP/IP properties for each network adapter, type:

ping dhcp

To verify connectivity with the DHCP server configured in the TCP/IP properties for the network adapter with an index number of 2, type:

ping dhcp 2

To verify connectivity with the DHCP server configured in the TCP/IP properties for each network adapter with "net" at the beginning of its name, type:

ping dhcp net*

ping dns

Verifies connectivity with the Domain Name System (DNS) servers configured in the TCP/IP properties for the specified adapter or adapters. Used without parameters, the **ping dns** command verifies connectivity with DNS servers through all installed adapters.

Top of page

Syntax

ping dns [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}]

Top of page

Parameters

***{IndexNumber* | *FullAdapterName* | *PartialAdapterName}* :** Specifies the adapter or adapters for which you want to verify connectivity. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To verify connectivity with the DNS server configured in the TCP/IP properties for each network adapter, type:

ping dns

To verify connectivity with the DNS server configured in the TCP/IP properties for the network adapter with an index number of 2, type:

ping dns 2

To verify connectivity with the DNS server configured in the TCP/IP properties for each network adapter with "net" at the beginning of its name, type:

ping dns net*

ping gateway

Verifies connectivity with the default gateway servers configured in the TCP/IP properties for the specified adapter or adapters. Used without parameters, the **ping gateway** command verifies connectivity through all installed adapters.

Top of page

Syntax

ping gateway [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to verify connectivity. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To verify connectivity with the gateway server configured in the TCP/IP properties for each network adapter, type:

ping gateway

To verify connectivity with the gateway server configured in the TCP/IP properties for the network adapter with an index number of 2, type:

ping gateway 2

To verify connectivity with the gateway server configured in the TCP/IP properties for each network adapter with "net" at the beginning of its name:

ping gateway net*

ping ieproxy

Verifies connectivity with the Web server proxy for Internet Explorer.

Top of page

Syntax

ping ieproxy

Top of page

Parameters

none

ping ip

Verifies connectivity with all IP addresses for the specified adapter or adapters. Used without parameters, the **ping ip** command verifies connectivity through all installed adapters.

Top of page

Syntax

ping ip [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to verify connectivity. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To verify connectivity with each network adapter, type:

ping ip

To verify connectivity with the network adapter with an index number of 2, type:

ping ip 2

To verify connectivity with each network adapter with "net" at the beginning of its name, type:

ping ip net*

ping iphost

Verifies connectivity with a remote or local host.

Top of page

Syntax

ping iphost *Node*

Top of page

Parameters

Node **:** Specifies a host by its IP address, its NetBIOS name, or its fully qualified domain name.

Top of page

Examples

To verify connectivity with the local host (at the loopback address of 127.0.0.1), type:

ping iphost 127.0.0.1

To verify connectivity with the host microsoft.com, type:

ping iphost microsoft.com

ping loopback

Verifies connectivity with the local host loopback address (127.0.0.1).

Top of page

Syntax

ping loopback

Top of page

Parameters

none

ping mail

Verifies connectivity with the mail server configured in the default Outlook Express profile of the current user.

Top of page

Syntax

ping mail

Top of page

Parameters

none

ping news

Verifies connectivity with the mail server configured in the default Outlook Express profile of the current user.

Top of page

Syntax

ping news

Top of page

Parameters

none

ping wins

Verifies connectivity with the preferred and alternate Windows Internet Name Service (WINS) servers for the specified network adapter or adapters. Used without parameters, **ping wins** verifies connectivity through all installed adapters.

Top of page

Syntax

ping wins [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}]

Top of page

Parameters

***{IndexNumber* | *FullAdapterName* | *PartialAdapterName}* :** Specifies the adapter or adapters for which you want to verify connectivity. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To verify connectivity with the WINS servers whose IP addresses are configured in the TCP/IP properties of each installed adapter, type:

ping wins

To verify connectivity with the WINS servers whose IP addresses are configured in the TCP/IP properties of each installed adapter with an index number number of 2, type:

ping wins 2

show adapter

Shows information about the specified logical adapter or adapters. Used without parameters, the **show adapter** command shows information about all adapters defined for this server.

Top of page

Syntax

show adapter [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}] [{**/v** | **/p**}]

Top of page

Parameters

***{IndexNumber* | *FullAdapterName* | *PartialAdapterName}* :** Specifies the adapter or adapters for which you want to show information. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- This command lists each adapter by index number in square brackets followed by the name of the adapter as it is registered. Typical output from this command follows:

  Network Adapters

  1. [00000001] 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible)
  2. [00000002] RAS Async Adapter
  3. [00000003] WAN Miniport (L2TP)
  4. [00000004] WAN Miniport (PPTP)
  5. [00000005] WAN Miniport (PPPOE)
  6. [00000006] Direct Parallel
  7. [00000007] WAN Miniport (IP)
  8. [00000008] WAN Miniport (Network Monitor)

Top of page

Examples

To display the logical adapters defined on the computer, type:

show adapter

show all

Lists all network objects defined for the local server. Used without parameters, the **show all** command lists overview configuration information for the following hardware and software components: Default Outlook Express Mail, Default Outlook Express News, Internet Explorer Web Proxy, Loopback, Computer System, Operating System, Version, Modems, Network Adapters, and Network Clients.

Top of page

Syntax

show all [{**/v** | **/p**}]

Top of page

Parameters

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- The output from this command can be lengthy and includes configuration information for mail, news, and proxy servers. It lists information for the TCP/IP loopback address and details about the computer, the operating system, and all network adapters and clients.

show client

Lists all network clients defined for the specified adapter or adapters. Used without parameters, **show client** lists clients for all installed adapters.

Top of page

Syntax

show client [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}] [{**/v** | **/p**}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to list clients. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The

PartialAdapterName parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To list all network clients defined for all network adapters in the host, type:

show client

To list all fields with non-null values for all network clients defined for adapter number 2, type:

show client 2 /p

show computer

Lists Windows Management Interface (WMI) details for the server. Used without parameters, the **show computer** command lists the NetBIOS name of the local computer.

Top of page

Syntax

show computer [{**/v** | **/p**}]

Top of page

Parameters

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Examples

To list all non-null WMI interface parameters for the server:

show computer /p

show dhcp

Lists all DHCP servers defined for the specified adapter. Used without parameters, the **show dhcp** command lists the servers defined for all installed adapters.

Top of page

Syntax

show dhcp {*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} [{**/v** | **/p**}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to list servers. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To list details of all DHCP servers defined for all network adapters, type:

show dhcp /v

To list details of any DHCP servers defined for network adapter 2, type:

show dhcp 2

show dns

Lists all DNS servers defined for a specified adapter. Used without parameters, the **show dns** command lists servers defined for all installed adapters.

Top of page

Syntax

show dns [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}] [{**/v** | **/p**}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to list servers. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To list all non-null details of all DNS servers defined for all network adapters, type:

show dns /p

To list any DNS servers defined for network adapter 2, type:

show dns 2

show gateway

Lists all Internet gateways defined for the specified adapter or adapters. Used without parameters, the **show gateway** command lists gateways defined for all installed adapters.

Top of page

Syntax

show gateway [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}] [{**/v** | **/p**}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to list gateways. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To list all available details about all Internet gateways defined for all adapters, type:

show gateway /v

To list all non-null values available for any gateways defined for network adapter 3, type:

show gateway 3 /p

show ieproxy

Lists Internet Explorer proxy servers defined for the specified adapter or adapters. Used without parameters, **show ieproxy** lists the servers for all installed adapters.

Top of page

Syntax

show ieproxy [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}] [{**/v** | **/p**}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to list servers. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, you must type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, you must type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, you must type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To list all Internet Explorer proxy servers defined for all network adapters in the server, type:

show ieproxy

show ip

Shows information about the specified adapter or adapters for which TCP/IP is enabled. Used without parameters, the **show ip** command shows information for all installed adapters.

Top of page

Syntax

show ip [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}] [{**/v** | **/p**}]

Top of page

Parameters

{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*} : Specifies the adapter or adapters for which you want to show information. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.

Top of page

Examples

To list the index number, name, and TCP/IP address for all adapters for which TCP/IP is enabled, type:

show ip

show mail

Shows the Outlook Express mail server specified for the local computer. Used without parameters, the **show mail** command lists overview configuration information for Default Outlook Express Mail.

Top of page

Syntax

show mail [{**/v** | **/p**}]

Top of page

Parameters

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Examples

To display the mail server configured for the local computer, type:

show mail

show modem

Lists details about the specified modem device or devices configured on the local computer. Used without parameters, **show modem** lists details about all modem devices.

Top of page

Syntax

show modem [{*IndexNumber* | *FullModemName* | *PartialModemName*}] [{**/v** | **/p**}]

Top of page

Parameters

{*IndexNumber* | *FullModemName* | *PartialModemName*} : Specifies the modem device or devices for which you want to list details. The *IndexNumber* parameter specifies the device by its index number. The *FullModemName* parameter specifies the device by its name as registered in the operating system. The *PartialModemName* specifies the device or devices whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Examples

To show all details, even those with null values, about all modem devices configured on the local computer, type:

show modem /v

To show the name and index number of all modem devices configured on the local computer with "dialup" at the beginning of their names, type:

show modem dialup*

show news

Displays configuration information for the news server defined for the local computer. Used without parameters, the **show news** command lists overview configuration information for Default Outlook Express News.

Top of page

Syntax

show news [{**/v** | **/p**}]

↑Top of page

Parameters

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

↑Top of page

Examples

To display the name and the TCP/IP address of the news server, type:

show news

show os

Displays information about the operating system running on the local computer.

↑Top of page

Syntax

show os [**/v**]

↑Top of page

Parameters

/v : Specifies that fields with a null value should appear.

↑Top of page

Remarks

- Information shown by this command is collected by the WMI provider win32_operatingsystem.

show test

Shows all network objects defined for the local computer and verifies connectivity with each object using the **ping** command. Used without parameters, the **show test** command lists overview configuration information for the following hardware and software components: Default Outlook Express Mail, Default Outlook Express News, Internet Explorer Web Proxy, Loopback, Computer System, Operating System, Version, Modems, Network Adapters, and Network Clients. In addition, connectivity is verified with the Loopback adapter and all IP addresses configured as a client option in the TCP/IP properties of each network adapter.

Top of page

Syntax

show test [{**/v** | **/p**}]

Top of page

Parameters

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- The output of this command can be lengthy and might require a few minutes to complete. The output includes configuration information for mail, news, and proxy servers as well as the results of attempts to verify connectivity to those servers using the **ping** command. This command lists information for and verifies connectivity with the TCP/IP loopback address. This command lists details about the computer and the operating system as well as all network adapters and clients. Connectivity is verified with network adapters and clients using the **ping** command, and the results are displayed.

Top of page

Examples

To show all information available to the Netsh diagnostic context, type:

show test /v

show version

Shows the version of the operating system and, optionally, the version of Windows Management Infrastructure (WMI). Used without parameters, the **show version** command lists the version number of the operating system installed on the local computer.

Top of page

Syntax

show version [{**/v** | **/p**}]

Top of page

Parameters

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- The first line of output for this command shows the version of the operating system. The second line shows the version of WMI. Typical output for this command follows:

```
Version (5.1.2416)
    Version = 5.1.2416
    BuildVersion = 2416.0000
```

Top of page

Examples

To show the version of the operating system and WMI, including fields with a null value, type:

show version /v

show wins

Lists WINS servers for the specified adapter or adapters. Used without parameters, the **show wins** command lists servers for all installed adapters.

Top of page

Syntax

show wins [{*IndexNumber* | *FullAdapterName* | *PartialAdapterName*}] [{**/v** | **/p**}]

Top of page

Parameters

***{IndexNumber* | *FullAdapterName* | *PartialAdapterName}* :** Specifies the adapter or adapters for which you want to list servers. The *IndexNumber* parameter specifies the adapter by its index number. The *FullAdapterName* parameter specifies the adapter by its name as registered in the operating system. The *PartialAdapterName* parameter specifies the adapter or adapters whose names contain the sequence of characters specified in the parameter. If the sequence of characters must appear at the beginning of the name, type an asterisk at the end of the parameter. If the sequence of characters must appear at the end of the name, type an asterisk at the beginning of the parameter. If the sequence of characters can appear anywhere in the name, type an asterisk at the beginning of the parameter and at the end of the parameter.

{/v | /p} : Specifies whether fields with a null value should appear. The **/v** parameter specifies that the fields should appear. The **/p** parameter specifies that the fields should not appear.

Top of page

Remarks

- You can use the **show adapter** command to display the index numbers and the full names of all logical adapters defined on the computer.
- If you specify an adapter by its index number, you do not need to type the leading zeroes.
- Typical output for this command is as follows:

```
WINS Servers
    2. [00000002] 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX
Compatible)
    WINSPrimaryServer = 172.16.44.17
    WINSSecondaryServer = 172.16.32.22
```

Top of page

Examples

To list the WINS servers defined for all adapters on the server, type:

show wins

To list the WINS servers specified for the adapter with index number 2, type:

show wins 2

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Netsh commands for Interface IP

You can use commands in the Netsh Interface IP context to configure the TCP/IP protocol (including addresses, default gateways, DNS servers, and WINS servers) and to display configuration and statistical information.

You can run these commands from the Windows XP command prompt or from the command prompt for the Netsh Interface IP context. For these commands to work at the Windows XP command prompt, you must type **netsh interface ip** before typing commands and parameters as they appear in the syntax below. To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP.

For more information on Netsh, see Using Netsh

To view the command syntax, click a command:

set address

Configures an IP address and a default gateway on a specified interface.

Syntax

set address [**name=**]*InterfaceName* [**source=**]{**dhcp** | **static** [**addr=**]*IPAddress* [**mask=**]*SubnetMask* [**gateway=**]{**none** | *DefaultGateway* [[**gwmetric=**]*GatewayMetric*]}}

⇡Top of page

Parameters

[name =] *InterfaceName* : Required. Specifies the name of the interface for which you want to configure address and gateway information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[source=]{dhcp | static [addr=]*IPAddress* [mask=]*SubnetMask* [gateway=]{none | *DefaultGateway* [[gwmetric=]*GatewayMetric*]}} : Required. Specifies whether the IP address to configure originates from a Dynamic Host Configuration Protocol (DHCP) server or is static. If the address is static, *IPAddress* specifies the address to configure, and *SubnetMask* specifies the subnet mask for the IP address being configured. If the address is static, you must also specify whether you want to leave the current default gateway (if any) in place or configure one for the address. If you configure a default gateway, *DefaultGateway* specifies the IP address of the default gateway to be configured, and *GatewayMetric* specifies the metric for the default gateway to be configured.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can use this command to switch an interface between DHCP configuration and static configuration. If the interface is configured statically, the DNS and WINS server addresses for this interface must also be statically configured.
- Omitting the *DefaultGateway* parameter does not clear gateway configuration but preserves whatever was previously configured.

add address

Adds an IP address and a default gateway on a specified interface configured with a static IP address.

Top of page

Syntax

add address [**name=**]*InterfaceName* [**addr=**]*IPAddress* [**mask=**]*SubnetMask* [[**gateway=**]*DefaultGateway* [**gwmetric=**]*GatewayMetric*]

Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to add address and gateway information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[addr=]*IPAddress* [mask=]*SubnetMask* : Required. Specifies the IP address to add and the subnet mask for that IP address.

[gateway=]*DefaultGateway* [gwmetric=]*GatewayMetric* : Specifies the IP address of the default gateway to add and the metric for that default gateway.

/?: Displays help at the command prompt.

delete address

Deletes an IP address or a default gateway on a statically configured interface.

⇡Top of page

Syntax

delete address [**name=**]*InterfaceName* [**addr=**] *IPAddress* [[**gateway=**]{*DefaultGateway* | **all**}]

⇡Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to delete address and gateway information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[addr=]*IPAddress* : Required. Specifies the IP address to delete.

[gateway=]{*DefaultGateway* | all} : Specifies whether to delete one default gateway or all default gateways. If only one default gateway should be deleted, *DefaultGateway* specifies the IP address of the default gateway to be deleted.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command will not delete the last IP address on the interface.

show address

Displays information about static IP addresses and default gateways on a specified interface. Used without parameters, **show address** displays address information for all interfaces.

Top of page

Syntax

show address [[**name=**]*InterfaceName*]

Top of page

Parameters

[name=]*InterfaceName* : Specifies the name of the interface for which you want to display address information. The *InterfaceName* must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

/?: Displays help at the command prompt.

Top of page

Remarks

- For interfaces configured using DHCP, use the **ipconfig** command to display the assigned IP addresses, lease parameters, and other DHCP information.

set dns

Configures a DNS server address for a specified interface.

Top of page

Syntax

set dns [**name=**]*InterfaceName* [**source=**]{**dhcp** | **static** [**addr=**]{*DNSAddress* | **none**}} [[**ddns=**]{**disabled** | **enabled**}] [[**suffix=**]{**interface** | **primary**}]

⇡Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to set DNS information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[source=]{dhcp | static [addr =]{*DNSAddress* | none}} : Required. Specifies whether the IP address of the DNS server is configured by DHCP or is static. If the IP address is static, *DNSAddress* specifies the IP address of the DNS server to configure, and **none** specifies that the DNS configuration should be removed.

[ddns=]{disabled | enabled} : Specifies whether the computer should attempt DNS dynamic update registration of the IP addresses of this connection and the full computer name. The full computer name is specified in the Control Panel item **System** on the **Computer Name** tab. The **enabled** parameter specifies that the attempt should occur. The **disabled** parameter specifies that the attempt should not occur.

[suffix=]{interface | primary} : Specifies whether the computer should attempt DNS dynamic update to register the IP addresses and the connection-specific domain name of the specified connection in addition to the full computer name. The connection-specific DNS name of this connection is the concatenation of the computer name (which is the first label of the full computer name) and the DNS suffix of this connection. The **interface** parameter specifies that both the full computer name and connection-specific name should be registered. The **primary** parameter specifies that only the full computer name should be registered.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- If the interface is already statically configured, the **static** parameter replaces the existing list of DNS server addresses with the one specified in the command.
- The default setting for **ddns=** is **enabled**.
- The default setting for **suffix=** is **primary**.

add dns

Adds a DNS server to a list of DNS servers for a specified interface.

↑Top of page

Syntax

add dns [**name=**]*InterfaceName* [**addr=**] *DNSAddress* [[**index=**]*DNSIndex*]

↑Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to add DNS information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[addr=]*DNSAddress* : Required. Specifies the IP address of the DNS server to add.

[index=]*DNSIndex* : Specifies the position of the added DNS server in the list of DNS servers for the interface.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- A lower number for the *DNSIndex* parameter corresponds to a higher preference. If no index is specified, then the server is added with the lowest preference.

delete dns

Deletes a DNS server or all DNS servers from a list of DNS servers for a specified interface or for all interfaces.

Top of page

Syntax

delete dns [**name=**]*InterfaceName* [**addr=**]{*DNSAddress* | **all**}

Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to delete DNS information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[addr=]{*DNSAddress* | all} : Required. Specifies whether to delete the address of one DNS server or all servers for all interfaces. If only one DNS server should be deleted, *DNSAddress* specifies the IP address of the DNS server to delete.

/?: Displays help at the command prompt.

show dns

Displays the DNS configuration of a specified interface. Used without parameters, **show dns** displays the DNS configurations of all interfaces.

Top of page

Syntax

show dns [[**name=**]*InterfaceName*]

Top of page

Parameters

[name=]*InterfaceName* : Specifies the name of the interface whose DNS configuration you want to display. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

/?: Displays help at the command prompt.

set wins

Sets WINS server configuration to either DHCP or static mode for a specified interface.

⇡Top of page

Syntax

set wins [**name=**]*InterfaceName* [**source=**]{**dhcp** | **static** [**addr=**]{*WINSAddress* | **none** }}

⇡Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to set WINS information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[source=]{dhcp | static [addr=]{*WINSAddress* | none }} : Required. Specifies whether the IP address of the WINS server to configure should be assigned by DHCP or is static. If the IP address is static, *WINSAddress* specifies the IP address of the WINS server to configure, and **none** specifies that the WINS configuration should be removed.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- If the interface is already statically configured, the **static** parameter replaces the existing WINS server address list with the one specified in the **set wins** command.

add wins

Adds a WINS server to a list of WINS servers for a specified interface.

⇡Top of page

Syntax

add wins [**name=**]*InterfaceName* [**addr=**] *WINSAddress* [[**index=**]*WINSIndex*]

⇡Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to add WINS information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[addr=]*WINSAddress* : Required. Specifies the IP address of the WINS server to add.

[index=]*WINSIndex* : Specifies the position of the added WINS server in the WINS server list for that interface.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- A lower number for the *WINSIndex* parameter corresponds to a higher preference. If no index is specified, then the server is added with the lowest preference.

delete wins

Deletes a WINS server or servers from a list of WINS servers for a specified interface or all interfaces.

⇡Top of page

Syntax

delete wins [**name=**]*InterfaceName* [**addr=**]{*WINSAddress* | **all**}

⇡Top of page

Parameters

[name=]*InterfaceName* : Required. Specifies the name of the interface for which you want to delete a WINS server or servers. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[addr=]{*WINSAddress* | all} : Required. Specifies whether to delete only one server for an interface or all servers for all interfaces. If only one server should be deleted, *WINSAddress* specifies the IP address of the WINS server to delete.

/?: Displays help at the command prompt.

show wins

Displays the WINS configuration for a specified interface. Used without parameters, **show wins** displays the WINS configuration for all interfaces.

⇡Top of page

Syntax

show wins [[**name=**]*InterfaceName*]

⇡Top of page

Parameters

[name=]*InterfaceName* : Specifies the name of the interface whose WINS information you want to display. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

/?: Displays help at the command prompt.

show icmp

Displays ICMP statistics. Used without parameters, **show icmp** displays the statistics only once.

⇡Top of page

Syntax

show icmp [[**rr=**]*RefreshRate*]

⇡Top of page

Parameters

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

show interface

Displays statistics for a specified interface. Used without parameters, **show interface** displays statistics for all interfaces only once.

⇡Top of page

Syntax

show interface [[**index=**]*InterfaceIndex*] [[**rr=**]*RefreshRate*]

⇡Top of page

Parameters

[index=]*InterfaceIndex* : Specifies the interface index, an integer that identifies the interface.

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

⇡Top of page

Remarks

•To obtain the interface index for an interface, review the output of the **show interface** command used without parameters.

show ipaddress

Displays information for a specified IP address. Used without parameters, **show ipaddress** displays information for all IP addresses on all interfaces once.

⇡Top of page

Syntax

show ipaddress [[**index=**]*IPAddress*] [[**rr=**]*RefreshRate*]

⇡Top of page

Parameters

[index=]*IPAddress* : Specifies an IP address of an interface.

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

⇡Top of page

Remarks

•To obtain the IP addresses of all interfaces, review the display of the **show ipaddress** command used without parameters.

show ipnet

Displays the contents of the Address Resolution Protocol (ARP) cache, which contains the hardware addresses of resolved next-hop IP addresses. Used without parameters, **show ipnet** displays the information once.

Top of page

Syntax

show ipnet [[**rr=**]*RefreshRate*]

Top of page

Parameters

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

show ipstats

Displays IP statistics. Used without parameters, **show ipstats** displays the statistics once.

Top of page

Syntax

show ipstats [[**rr=**]*RefreshRate*]

Top of page

Parameters

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

show joins

Displays IP multicast groups that have been joined for the specified IP address. Used without parameters, **show joins** displays information for all IP addresses.

⇡Top of page

Syntax

show joins [[**index=**]*IPAddress*]

⇡Top of page

Parameters

[index=]*IPAddress* : Specifies an IP address of an interface.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- To obtain the IP addresses for all interfaces, review the display of the **show ipaddress** command used without parameters.

show tcpconn

Displays information on a specified TCP connection. Used without parameters, **show tcpconn** displays information for all TCP connections once.

⇡Top of page

Syntax

show tcpconn [[**index=**]{*LocalIPAddress* | *LocalPort* | *RemoteIPAddress* | *RemotePort*}] [[**rr=**]*RefreshRate*]

⇡Top of page

Parameters

[index=]{*LocalIPAddress* | *LocalPort* | *RemoteIPAddress* | *RemotePort*} : Specifies the connection about which to display information. The *LocalIPAddress* parameter specifies an IP address of an interface. The *LocalPort* parameter specifies a TCP port for a local process. The *RemoteIPAddress* parameter specifies an IP address of a remote host. The *RemotePort* parameter specifies a TCP port for a remote process.

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the information).

/?: Displays help at the command prompt.

show tcpstats

Displays TCP statistics. Used without parameters, **show tcpstats** displays the statistics once.

⇡Top of page

Syntax

show tcpstats [[**rr=**]*RefreshRate*]

⇡Top of page

Parameters

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

show udpconn

Displays information about the UDP ports used for each IP address. Used without parameters, **show udpconn** displays UDP port information for all IP addresses once.

Top of page

Syntax

show udpconn [[**index=**]{*LocalIPAddress* | *LocalPort*}] [[**rr=**]*RefreshRate*]

Top of page

Parameters

[index=]{*LocalIPAddress* | *LocalPort*} : Specifies the connection about which to display information. The *LocalIPAddress* parameter specifies an IP address of an interface. The *LocalPort* parameter specifies a UDP port for a local process.

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

show udpstats

Displays UDP statistics. Used without parameters, **show udpstats** displays the statistics once.

Top of page

Syntax

show udpstats [[**rr=**]*RefreshRate*]

Top of page

Parameters

[rr=]*RefreshRate* : Specifies the refresh rate (the number of seconds between refreshing the display of the statistics).

/?: Displays help at the command prompt.

show config

Displays IP address and other configuration information for a specified interface. Used without parameters, **show config** displays configuration information for all interfaces.

⇡Top of page

Syntax

show config [[**name=**]*InterfaceName*]

⇡Top of page

Parameters

[name=]*InterfaceName* : Specifies the name of the interface for which you want to display configuration information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

/?: Displays help at the command prompt.

show offload

Displays the tasks that can be performed by the network adapter for the specified interface corresponding to installed network hardware. Used without parameters, **show offload** displays offload information for all interfaces corresponding to installed network hardware.

⇡Top of page

Syntax

show offload [[**name=**]*InterfaceName*]

⇡Top of page

Parameters

[name=]*InterfaceName* : Specifies the name of the interface for which you want to display offload information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

/?: Displays help at the command prompt.

delete arpcache

Removes the entries in the Address Resolution Protocol (ARP) cache for a specified interface. Used without parameters, **delete arpcache** removes the entries in the ARP caches of all interfaces.

Top of page

Syntax

delete arpcache [[**name=**]*InterfaceName*]

Top of page

Parameters

[name=]*InterfaceName* : Specifies the name of the interface for which you want to remove the ARP cache entries. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

/?: Displays help at the command prompt.

dump

Displays the current configuration as a series of Netsh Interface IP commands.

Top of page

Syntax

dump

Top of page

Parameters

none

Top of page

Remarks

- To create a configuration script file, redirect the output of this command to a file.

Top of page

Netsh Interface IP example

The following command configures the interface named **Local Area Connection** with the static IP address 10.0.5.99, the subnet mask of 255.255.255.0, and a default gateway of 10.0.5.1:

```
set address name="Local Area Connection" source=static addr=10.0.5.99 mask=255.255.255.0 gateway=10.0.5.1
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Netsh commands for RAS

You can administer remote access servers by typing commands at the command prompt for the Netsh RAS context. By using the Netsh RAS command prompt, you can administer servers more quickly over slow network connections, and you can create scripts that automate the administration of many servers.

To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP.

For more information on Netsh, see Using Netsh

To view the command syntax, click a command:

help

Displays command-line help for commands you can use in the Netsh RAS context.

Syntax

help

⇡Top of page

Parameters

none

⇡Top of page

Remarks

- You can also display command-line help by typing any of the following at the command prompt for the Netsh RAS context: **?**, **/?**, or **-?**.

show activeservers

Displays a list of remote access server advertisements.

⇧Top of page

Syntax

show activeservers

⇧Top of page

Parameters

none

show client

Lists remote access clients connected to this server.

⇧Top of page

Syntax

show client

⇧Top of page

Parameters

none

dump

Displays the configuration of the remote access server in script form.

Syntax

dump

⇡Top of page

Parameters

none

show tracing

Shows whether tracing is enabled for the specified component. Used without parameters, **show tracing** lists all installed components and whether tracing is enabled for each.

⇡Top of page

Syntax

show tracing [*Component*]

⇡Top of page

Parameters

Component **:** Specifies the component for which to display information.

set tracing

Enables or disables tracing for the specified component.

⇡Top of page

Syntax

set tracing *Component* {**enabled** | **disabled**}

⇡Top of page

Parameters

Component **:** Required. Specifies the component for which you want to enable or disable tracing. Use ***** to specify all components.

{enabled | disabled} : Required. Specifies whether to enable or disable tracing for the specified component.

Top of page

Remarks

- To see a list of all installed components, use the **show tracing** command without parameters.

Top of page

Examples

To set tracing for the PPP component, type:

set tracing ppp enabled

show authmode

Shows whether dial-up clients using certain types of devices should be authenticated.

Top of page

Syntax

show authmode

Top of page

Parameters

none

set authmode

Specifies whether dial-up clients using certain types of devices should be authenticated.

Top of page

Syntax

set authmode {**standard** | **nodcc** | **bypass**}

Top of page

Parameters

{standard | nodcc | bypass} : Required. Specifies whether dial-up clients using certain types of devices should be authenticated. The parameter **standard** specifies that clients using any type of device should be authenticated. The parameter **nodcc** specifies that clients using any type of device except a direct-connect device should be authenticated. The parameter **bypass** specifies that no clients should be authenticated.

add authtype

Adds an authentication type to the list of types through which the remote access server should attempt to negotiate authentication.

Top of page

Syntax

add authtype {**pap**|**spap**|**md5chap**|**mschap**|**mschapv2**|**eap**}

Top of page

Parameters

{pap|spap|md5chap|mschap|mschapv2|eap} : Required. Specifies which authentication type to add to the list of types through which the remote access server should attempt to negotiate authentication. The **pap** parameter specifies that the remote access server should use the Password Authentication Protocol (clear text). The **spap** parameter specifies that the remote access server should use the Shiva Password Authentication Protocol. The **md5chap** parameter specifies that the remote access server should use the Challenge Handshake Authentication Protocol (using the Message Digest 5 hashing scheme to encrypt the response). The **mschap** parameter specifies that the remote access server should use the Microsoft Challenge-Handshake Authentication Protocol. The **mschapv2** parameter specifies that the remote access server should use Version 2 of MSCHAP. The **eap** parameter specifies that the remote access server should use Extensible Authentication Protocol.

⇡Top of page

Remarks

- The remote access server will attempt to negotiate authentication by using protocols in order from the most secure to the least secure. After both the client and the server have agreed on an authentication type, PPP negotiation proceeds according to the appropriate RFCs.

delete authtype

Deletes an authentication type from the list of types through which the remote access server should attempt to negotiate authentication.

⇡Top of page

Syntax

delete authtype {**pap**|**spap**|**md5chap**|**mschap**|**mschapv2**|**eap**}

⇡Top of page

Parameters

{pap|spap|md5chap|mschap|mschapv2|eap} : Required. Specifies which authentication type to delete from the list of types through which the remote access server should attempt to negotiate authentication. The **pap** parameter specifies that the remote access server should not use the Password Authentication Protocol (clear text). The **spap** parameter specifies that the remote access server should not use the Shiva Password Authentication Protocol. The **md5chap** parameter specifies that the remote access server should not use the Challenge Handshake Authentication Protocol (using the Message Digest 5 hashing scheme to encrypt the response). The **mschap** parameter specifies that the remote access server should not use the Microsoft Challenge-Handshake Authentication Protocol. The **mschapv2** parameter specifies that the remote access server should not use Version 2 of MSCHAP. The **eap** parameter specifies that the remote access server should not use Extensible Authentication Protocol.

show authtype

Lists the authentication type (or types) that the remote access server uses to attempt to negotiate authentication.

⇡Top of page

Syntax

show authtype

⇡Top of page

Parameters

none

add link

Adds a link property to the list of link properties PPP should negotiate.

⇡Top of page

Syntax

add link {**swc** | **lcp**}

⇡Top of page

Parameters

{swc | lcp} : Required. Specifies which link property to add to the list of link properties PPP should negotiate. The parameter **swc** specifies that software compression (MPPC) should be added. The parameter **lcp** specifies that Link Control Protocol extensions from the PPP suite of protocols should be added.

delete link

Deletes a link property from the list of link properties PPP should negotiate.

Top of page

Syntax

delete link {**swc** | **lcp**}

Top of page

Parameters

{swc | lcp} : Required. Specifies which link property to delete from the list of link properties PPP should negotiate. The parameter **swc** specifies that software compression (MPPC) should be deleted. The parameter **lcp** specifies that Link Control Protocol extensions from the PPP suite of protocols should be deleted.

show link

Displays the link properties PPP should negotiate.

Top of page

Syntax

show link

Top of page

Parameters

none

add multilink

Adds a multilink type to the list of multilink types PPP should negotiate.

Top of page

Syntax

add multilink {**multi** | **bacp**}

Top of page

Parameters

{multi | bacp} : Required. Specifies which multilink type to add to the list of multilink types PPP should negotiate. The parameter **multi** specifies that multilink PPP sessions should be added. The parameter **bacp** specifies that Bandwidth Allocation Control Protocol should be added.

delete multilink

Deletes a multilink type from the list of multilink types PPP should negotiate.

Top of page

Syntax

delete multilink {**multi** | **bacp**}

Top of page

Parameters

{multi | bacp} : Required. Specifies which multilink type to delete from the list of multilink types PPP should negotiate. The parameter **multi** specifies that multilink PPP sessions should be deleted. The parameter **bacp** specifies that Bandwidth Allocation Control Protocol should be deleted.

show multilink

Shows the multilink types PPP should negotiate.

Top of page

Syntax

show multilink

Top of page

Parameters

none

add registeredserver

Registers the specified server as a remote access server in the specified Active Directory domain. Used without parameters, **add registeredserver** registers the computer from which you type the command in its primary domain.

Top of page

Syntax

add registeredserver [[**domain=**]*DomainName*] [[**server=**]*ServerName*]

Top of page

Parameters

[domain=]*DomainName* : Specifies, by domain name, the domain in which to register the server. If you do not specify a domain, the server is registered in its primary domain.

[server=]*ServerName* : Specifies, by DNS name or IP address, the server to register. If you do not specify a server, the computer from which you type the command is registered.

delete registeredserver

Deletes the registration of the specified server as a remote access server from the specified Active Directory domain. Used without parameters, **delete registeredserver** deletes the registration of the computer from which you type the command from its primary domain.

Top of page

Syntax

delete registeredserver [[**domain=**]*DomainName*] [[**server=**]*ServerName*]

Top of page

Parameters

[domain=]*DomainName* : Specifies, by domain name, the domain from which to remove the registration. If you do not specify a domain, the registration is removed from the primary domain of the computer from which you type the command.

[server=]*ServerName* : Specifies, by IP address or DNS name, the server whose registration you want to remove. If you do not specify a server, the registration is removed for the computer from which you type the command.

show registeredserver

Displays status information about the specified server registered as a remote access server in the specified Active Directory domain. Used without parameters, **show registeredserver** displays information about the computer from which you type the command in its primary domain.

Top of page

Syntax

show registeredserver [[**domain=**]*DomainName*] [[**server=**]*ServerName*]

Top of page

Parameters

[domain=]*DomainName* : Specifies, by domain name, the domain in which the server about which you want to display information is registered. If you do not specify a domain, information is displayed about the server as it is registered in the primary domain of the computer from which you type the command.

[server=]*ServerName* : Specifies, by IP address or DNS name, the server about which you want to display information. If you do not specify a server, information about the computer from which you typed the command is displayed.

show user

Displays the properties of a specified remote access user or users. Used without parameters, **show user** displays the properties of all remote access users.

Top of page

Syntax

show user [**name=***UserName*] [[**mode=**] {**permit** | **report**}]

Top of page

Parameters

name=*UserName* : Specifies, by logon name, the user whose properties you want to display. If you do not specify a user, the properties of all users are displayed.

mode= {permit | report} : Specifies whether to show properties for all users or only those whose dial-up permission is set to permit. The **permit** parameter specifies that properties should be displayed only for users whose dial-up permission is permit. The **report** parameter specifies that properties should be displayed for all users.

set user

Sets the properties of the specified remote access user.

Top of page

Syntax

set user [**name=**]*UserName* [**dialin**] {**permit** | **deny** | **policy**} [**cbpolicy**] {**none** | **caller** | **admin** [**cbnumber=**]*CallbackNumber*}

Top of page

Parameters

name=*UserName* : Required. Specifies, by logon name, the user for which you want to set properties.

[dialin] {permit | deny | policy} : Required. Specifies under what circumstances the user should be allowed to connect. The **permit** parameter specifies that the user should always be allowed to connect. The **deny** parameter specifies that the user should never be allowed to connect. The **policy** parameter specifies that remote access policies should determine whether the user is allowed to connect.

[cbpolicy] {none | caller | admin [cbnumber=]*CallbackNumber*} : Required. Specifies the callback policy for the user. The callback feature saves the user the cost of the phone call used to connect to a remote access server. The **none** parameter specifies that the user should not be called back. The **caller** parameter specifies that the user should be called back at a number specified by the user at connection time. The **admin** parameter specifies that the user should be called back at the number specified by the *CallbackNumber* parameter.

Top of page

Remarks

- For users in a mixed-mode domain, the **policy** parameter and the **deny** parameter are equivalent.

Top of page

Examples

To allow GuestUser to connect and be called back at 4255551212, type:

set user guestuser permit admin 4255551212

Top of page

Netsh RAS IP context commands

The following commands are specific to the **RAS IP** context within the Netsh environment.

To view the command syntax, click a command:

dump

Creates a script that contains the IP configuration of a remote access server. If you save this script to a file, you can use it to restore IP configuration settings.

Syntax

dump

Parameters

none

show config

Displays the current IP configuration of the remote access server.

Syntax

show config

Parameters

none

set negotiation

Specifies whether the remote access server should allow IP to be configured for any client connections the server accepts.

Syntax

set negotiation {**allow** | **deny**}

Parameters

{allow | deny} : Required. Specifies whether to permit IP over client connections. The **allow** parameter allows IP over client connections. The **deny** parameter prevents IP over client connections.

set access

Specifies whether IP network traffic from any client should be forwarded to the network or networks to which the remote access server is connected.

Syntax

set access {**all** | **serveronly**}

Parameters

{all | serveronly} : Required. Specifies whether clients should be able to reach the remote access server and any networks to which it is connected. The **all** parameter allows clients to reach networks through the server. The **serveronly** parameter allows clients to reach only the server.

set addrassign

Sets the method by which the remote access server should assign IP addresses to its clients.

Syntax

set addrassign {**auto** | **pool**}

Parameters

{auto | pool} : Required. Specifies whether IP addresses should be assigned by using DHCP or from a pool of addresses held by the remote access server. The **auto** parameter specifies that addresses should be assigned by using DHCP. If no DHCP server is available, a random, private address is assigned. The **pool** parameter specifies that addresses should be assigned from a pool.

set addrreq

Specifies whether dial-up clients should be able to request their own IP addresses.

Syntax

set addrreq {**allow** | **deny**}

Parameters

{allow | deny} : Required. Specifies whether clients should be able to request their own IP addresses. The **allow** parameter allows clients to request addresses. The **deny** parameter prevents clients from requesting addresses.

set broadcastnameresolution

Enables or disables broadcast name resolution using NetBIOS over TCP/IP.

Syntax

set broadcastnameresolution {**enabled** | **disabled**}

Parameters

{enabled | disabled} : Required. Specifies whether to enable or disable broadcast name resolution using NetBIOS over TCP/IP.

add range

Adds a range of addresses to the pool of static IP addresses that the remote access server can assign to clients.

Syntax

add range [**from**=]*StartingIPAddress* [**to**=]*EndingIPAddress*

Parameters

[from=]*StartingIPAddress* [to=]*EndingIPAddress* : Required. Specifies the range of IP addresses to add. The *StartingIPAddress* parameter specifies the first IP address in the range. The *EndingIPAddress* parameter specifies the last IP address in the range.

Examples

To add the range of IP addresses 10.2.2.10 to 10.2.2.20 to the static pool of IP addresses that the remote access server can assign, type:

add range from=10.2.2.10 to=10.2.2.20

delete range

Deletes a range of addresses from the pool of static IP addresses that the remote access server can assign to clients.

Syntax

delete range [**from**=]*StartingIPAddress* [**to**=]*EndingIPAddress*

Parameters

[from=]*StartingIPAddress* [to=]*EndingIPAddress* : Required. Specifies the range of IP addresses to delete. The *StartingIPAddress* parameter specifies the first IP address in the range. The *EndingIPAddress* parameter specifies the last IP address in the range.

Examples

To delete the range of IP addresses 10.2.2.10 to 10.2.2.20 from the pool of static IP addresses that the remote access server can assign, type:

delete range from=10.2.2.10 to=10.2.2.20

delete pool

Deletes all addresses from the pool of static IP addresses that the remote access server can assign to clients.

Syntax

delete pool

Parameters

⇡Top of page

Netsh RAS IPX context commands

The following commands are specific to the **RAS IPX** context within the Netsh environment. The IPX/SPX protocol is not available on Windows XP 64-Bit Edition.

To view the command syntax, click a command:

dump

Creates a script that contains the IPX configuration of the remote access server. If you save this script to a file, you can use it to restore IPX configuration settings.

Syntax

dump

Parameters

none

show config

Displays the current IPX configuration of the remote access server.

Syntax

show config

Parameters

none

set negotiation

Specifies whether the remote access server should allow IPX to be configured for any client connections it accepts.

Syntax

set negotiation {**allow** | **deny**}

Parameters

{allow | deny} : Required. Specifies whether to allow IPX configuration. The **allow** parameter allows IPX configuration. The **deny** parameter prevents IPX configuration.

set access

Specifies whether IPX network traffic from any client should be forwarded to the network or networks to which the remote access server is connected.

Syntax

set access {**all** | **serveronly**}

Parameters

{all | serveronly} : Required. Specifies whether IPX traffic should be forwarded. The **all** parameter allows IPX traffic to be forwarded. The **serveronly** parameter prevents traffic from being forwarded.

set netassign

Specifies the method by which the remote access server assigns IPX addresses to its clients.

Syntax

set netassign [**method**] {**auto** | **pool** | **autosame** | **poolsame**}

Parameters

[method] {auto | pool | autosame | poolsame} : Required. Specifies the method by which the remote access server assigns IPX addresses to clients. The **auto** parameter specifies that the remote access server should assign a different random IPX network number to each client. The **pool** parameter specifies that the remote access server should assign a different IPX address from a pool of addresses to each client. The **autosame** parameter specifies that the remote access server should generate a random IPX network number and assign it to all clients. The **poolsame** parameter specifies that the remote access server should assign an address from a pool to all clients.

Remarks

- For most configurations, either the **autosame** parameter or the **poolsame** parameter is recommended because they conserve network numbers and reduce network traffic.
- Before the remote access server assigns a network number to a client, the server must verify whether the number is already in use on the intranet to which the remote access server is connected. As a result, some addresses in the pool might not be assigned.

set pool

Specifies the pool of IPX addresses from which the remote access server can assign addresses to clients.

Syntax

set pool [**firstnet**=]*IPXAddress* [**size**=]{*PoolSize* | **0**}

Parameters

[firstnet=]*IPXAddress* : Required. Specifies, in hexadecimal notation, the first IPX address in the pool.

[size=]{*PoolSize* | 0} : Required. Specifies the size of the pool or that the pool should grow as needed. The *PoolSize* parameter specifies, in decimal notation, the number of IPX addresses in the pool. The **0** parameter specifies that the pool should grow as needed.

Examples

To specify that the pool of IPX addresses should start at AAAAAAA and grow as needed, type:

set pool AAAAAAA 0

set nodereq

Specifies whether dial-up clients should be permitted to request their own IPX node numbers.

Syntax

set nodereq {**allow** | **deny**}

Parameters

{allow | deny} : Required. Specifies whether to allow clients to request their own IPX node numbers. The **allow** parameter grants such requests. The **deny** parameter ignores such requests.

Top of page

Netsh RAS AppleTalk context commands

The following commands are specific to the **RAS AppleTalk** context within the Netsh environment. The AppleTalk protocol is not available on Windows XP Home Edition or Windows XP Professional.

To view the command syntax, click a command:

dump

Creates a script that contains the AppleTalk configuration of the remote access server. If you save this script to a file, you can use it to restore AppleTalk configuration settings.

Syntax

dump

Parameters

none

show config

Displays the current AppleTalk configuration of the remote access server.

Syntax

show config

Parameters

none

set negotiation

Specifies whether the remote access server should allow AppleTalk to be configured for any client connections the server accepts.

Syntax

set negotiation {**allow** | **deny**}

Parameters

{allow | deny} : Required. Specifies whether to allow AppleTalk configuration. The **allow** parameter allows configuration. The **deny** parameter prevents configuration.

set access

Specifies whether AppleTalk network traffic from any client should be forwarded to the network or networks to which the remote access server is connected.

Syntax

set access {**all** | **serveronly**}

Parameters

{all | serveronly} : Required. Specifies whether to forward AppleTalk network traffic. The **all** parameter allows traffic to be forwarded. The **serveronly** parameter prevents traffic from being forwarded.

Top of page

Netsh RAS AAAA context commands

The following commands are specific to the **RAS AAAA** context within the Netsh environment.

To view the command syntax, click a command:

dump

Displays the AAAA configuration of a remote access server in script form.

Syntax

dump

Parameters

none

add acctserv

Specifies the IP address or the DNS name of a RADIUS server to use for accounting and specifies accounting options.

Syntax

add acctserv [**name**=]*ServerID* [[**secret**=]*SharedSecret*] [[**init-score**=]*ServerPriority*] [[**port**=]*Port*] [[**timeout**=]*Seconds*] [[**messages**] {**enabled** | **disabled**}]

Parameters

[name=]*ServerID* : Required. Specifies, by IP address or DNS name, the RADIUS server.

[secret=]*SharedSecret* : Specifies the shared secret.

[init-score=]*ServerPriority* : Specifies the initial score (server priority).

[port=]*Port* : Specifies the port to which accounting requests should be sent.

[timeout=]*Seconds* : Specifies the timeout period, in seconds, during which the RADIUS server can be idle before it should be marked unavailable.

[messages] {enabled | disabled} : Specifies whether to send accounting on/off messages. The **enabled** parameter specifies that messages should be sent. The **disabled** parameter specifies that messages should not be sent.

delete acctserv

Deletes a RADIUS accounting server.

Syntax

delete acctserv [**name**=]*ServerID*

Parameters

[name=]*ServerID* : Required. Specifies, by DNS name or IP address, which server to delete.

set acctserv

Provides the IP address or the DNS name of a RADIUS server to use for accounting.

Syntax

add acctserv [**name=**]*ServerID* [[**secret=**]*SharedSecret*] [[**init-score=**]*ServerPriority*] [[**port=**]*Port*] [[**timeout=**]*Seconds*] [[**messages**] {**enabled** | **disabled**}]

Parameters

[name=]*ServerID* : Required. Specifies, by IP address or DNS name, the RADIUS server.

[secret=]*SharedSecret* : Specifies the shared secret.

[init-score=]*ServerPriority* : Specifies the initial score (server priority).

[port=]*Port* : Specifies the port on which to send the authentication requests.

[timeout=]*Seconds* : Specifies, in seconds, the amount of time that should elapse before the RADIUS server is marked unavailable.

[messages=] {enabled | disabled} : Specifies whether accouting on/off messages should be sent.

show acctserv

Displays detailed information about an accounting server. Used without parameters, **show acctserv** displays information about all configured accounting servers.

Syntax

show acctserv [[**name**=]*ServerID*]

Parameters

[name=]*ServerID* : Specifies, by DNS name or IP address, the RADIUS server about which to display information.

add authserv

Provides the IP address or the DNS name of a RADIUS server to which authentication requests should be passed.

Syntax

add authserv [**name**=]*ServerID* [[**secret**=]*SharedSecret*] [[**init-score**=]*ServerPriority*] [[**port**=]*Port*] [[**timeout**=]*Seconds*] [[**signature**] {**enabled** | **disabled**}]

Parameters

[name=]*ServerID* : Required. Specifies, by IP address or DNS name, the RADIUS server.

[secret=]*SharedSecret* : Specifies the shared secret.

[init-score=]*ServerPriority* : Specifies the initial score (server priority).

[port=]*Port* : Specifies the port to which authentication requests should be sent.

[timeout=]*Seconds* : Specifies the timeout period, in seconds, during which the RADIUS server can be idle before it should be marked unavailable.

[signature] {enabled | disabled} : Specifies whether to use digital signatures. The **enabled** parameter specifies that digital signatures should be used. The **disabled** parameter specifies that digital signatures should not be used.

delete authserv

Deletes a RADIUS authentication server.

Syntax

delete authserv [**name**=]*ServerID*

Parameters

[name=]*ServerID* : Required. Specifies, by DNS name or IP address, which server to delete.

set authserv

Provides the IP address or the DNS name of a RADIUS server to which authentication requests should be passed.

Syntax

set authserv [**name=**]*ServerID* [[**secret=**]*SharedSecret*] [[**init-score=**]*ServerPriority*] [[**port=**]*Port*] [[**timeout=**]*Seconds*] [[**signature**] {**enabled** | **disabled**}]

Parameters

[name=]*ServerID* : Required. Specifies, by IP address or DNS name, the RADIUS server.

[secret=]*SharedSecret* : Specifies the shared secret.

[init-score=]*ServerPriority* : Specifies the initial score (server priority).

[port=]*Port* : Specifies the port on which to send the authentication requests.

[timeout=]*Seconds* : Specifies the amount of time, in seconds, that should elapse before the RADIUS server is marked unavailable.

[signature=] {enabled | disabled} : Specifies whether digital signatures should be used.

show authserv

Displays detailed information about an authentication server. Used without parameters, **show authserv** displays information about all configured authentication servers.

Syntax

show authserv [[**name**=]*ServerID*]

Parameters

[name=]*ServerID* : Specifies, by DNS name or IP address, the RADIUS server about which to display information.

set acco

Specifies the accounting provider.

Syntax

set acco provider {**windows** | **radius** | **none**}

Parameters

provider {windows | radius | none} : Required. Specifies whether accounting should be performed and by which server. The **windows** parameter specifies that Windows security should perform accounting. The **radius** parameter specifies that a RADIUS server should perform accounting. The **none** parameter specifies that no accounting should be performed.

show acco

Displays the accounting provider.

Syntax

show acco

Parameters

none

set authe

Specifies the authentication provider.

Syntax

set authe provider {**windows** | **radius**}

Parameters

provider {windows | radius} : Required. Specifies which technology should perform authentication. The **windows** parameter specifies that Windows security should perform authentication. The **radius** parameter specifies that a RADIUS server should perform authentication.

show authe

Displays the authentication provider.

Syntax

show authe

Parameters

none

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Netsh commands for Routing

You can use the Netsh commands for Routing to administer routing servers from the command line, rather than through the Routing and Remote Access console. Administering servers from the command line is most useful in the following situations:

- You can administer routing servers more quickly from the command line than you can using a console. The ability is especially important if you are using a slow-speed network link, for example, a connection over a wide area network (WAN).
- You can administer routing servers more consistently by incorporating these commands into a script that you can run on many servers.

You can use the following commands at the command prompt for the appropriate Netsh context (as distinguished from the command prompt for Windows XP) to administer routing servers. To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP.

For more information on Netsh, see

Netsh Routing IP IGMP commands

The following commands are available for use in the Netsh Routing IP IGMP context.

To view the command syntax, click a command:

add interface

Configures IGMP on the specified interface.

Syntax

add interface [**InterfaceName=**]*InterfaceName* [[**IgmpPrototype=**]{**igmprtrv1** | **igmprtrv2** | **igmprtrv3** | **igmpproxy**}] [[**IfEnabled=**]{**enable** | **disable**}] [[**RobustVar**=]*Integer*] [[**GenQueryInterval=**]*Integer*] [[**GenQueryRespTime=**]*Integer*] [[**StartUpQueryCount=**]*Integer*] [[**StartUpQueryInterval=**]*Integer*] [[**LastMemQueryCount=**]*Integer*]

[[**LastMemQueryInterval=***Integer*] [[**AccNonRtrAlertPkts=**]{**yes** | **no**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies the name of the interface on which you want to configure IGMP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[IgmpPrototype=]{igmprtrv1 | igmprtrv2 | igmprtrv3 | igmpproxy} : Specifies the type of protocol to be configured on the interface. The default type is igmprtrv2.

[IfEnabled=]{enable | disable} : Specifies whether the protocol you are configuring should be enabled or disabled. The default is enabled.

[RobustVar=]=*Integer* : Specifies the robustness variable. The value should be greater than zero (0). The default value is 2.

[GenQueryInterval=]*Integer* : Specifies the interval, in seconds, at which general queries should be sent on the interface. The default value is 125 seconds.

[GenQueryRespTime=]*Integer* : Specifies the maximum response time, in seconds, by which hosts should respond to a general query. The default value is 10 seconds.

[StartUpQueryCount=]*Integer* : Specifies the number of general queries that will be sent during startup. The default value is 2.

[StartUpQueryInterval=]*Integer* : Specifies the interval, in seconds, between general queries sent during startup. The default value is 31 seconds.

[LastMemQueryCount=]*Integer* : Specifies the number of group-specific queries sent when notice of a host leaving the group is received. The default value is 2.

[LastMemQueryInterval=]*Integer* : Specifies the interval, in milliseconds, between group-specific queries. The default value is 1000 milliseconds.

[AccNonRtrAlertPkts=]{yes | no} : Specifies whether to accept IGMP packets that might not have the router alert option set. The default is **yes**.

Examples

To modify the default startup query interval to 21 seconds for use with IGMP configuration of the interface named Local Area Connection, type:

add interface "Local Area Connection" startupqueryinterval=21

install

Installs an IGMP router/proxy and sets global logging. Used without parameters, **install** installs the IGMP router/proxy and specifies that errors related to IGMP are logged but no other IGMP-related events are logged.

Syntax

install [[**LogLevel**=]{**none** | **error** | **warn** | **info**}]

Parameters

[LogLevel=]{none | error | warn | info} : Specifies which events should be logged. The **none** parameter specifies no events related to IGMP should be logged. The **error** parameter specifies that only errors related to IGMP should be logged. The **warn** parameter specifies that only warnings related to IGMP should be logged. The **info** parameter specifies that all events related to IGMP should be logged.

delete interface

Removes an IGMP router or proxy from the specified interface.

Syntax

delete interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, an interface from which you want to remove an IGMP router or proxy. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

Examples

delete interface "Local Area Connection"

uninstall

Removes IGMP router/proxy from all interfaces.

Syntax

uninstall

Parameters

none

set interface

Changes interface configuration parameters.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [[**IgmpPrototype=**]{**igmprtrv1** | **igmprtrv2** | **igmprtrv3** | **igmpproxy**}] [[**IfEnabled=**]{**enable** | **disable**}] [[**RobustVar=**]*Integer*] [[**GenQueryInterval=**]*Integer*] [[**GenQueryRespTime=**]*Integer*] [[**StartUpQueryCount=**]*Integer*] [[**StartUpQueryInterval=**]*Integer*] [[**LastMemQueryCount=**]*Integer*] [[**LastMemQueryInterval=**]*Integer*] [[**AccNonRtrAlertPkts=**]{**yes** | **no**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies the name of the interface on which you want to configure IGMP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[IgmpPrototype=]{igmprtrv1 | igmprtrv2 | igmprtrv3 | igmpproxy} : Specifies the type of protocol to be configured on the interface. The default type is igmprtrv2.

[IfEnabled=]{enable | disable} : Specifies whether the protocol you are configuring should be enabled or disabled. The default is enabled.

[RobustVar=]*Integer* : Specifies the robustness variable. The value should be greater than zero (0). The default value is 2.

[GenQueryInterval=]*Integer* : Specifies the interval, in seconds, at which general queries should be sent on the interface. The default value is 125 seconds.

[GenQueryRespTime=]*Integer* : Specifies the maximum response time, in seconds, by which hosts should respond to a general query. The default value is 10 seconds.

[StartUpQueryCount=]*Integer* : Specifies the number of general queries that will be sent during startup. The default value is 2.

[StartUpQueryInterval=]*Integer* : Specifies the interval, in seconds, between general queries sent during startup. The default value is 31 seconds.

[LastMemQueryCount=]*Integer* : Specifies the number of group-specific queries sent when notice of a host leaving the group is received. The default value is 2.

[LastMemQueryInterval=]*Integer* : Specifies the interval, in milliseconds, between group-specific queries. The default value is 1,000 milliseconds.

[AccNonRtrAlertPkts=]{yes | no} : Specifies whether to accept IGMP packets that do not have the router alert option set. The default is to accept those packets.

Examples

To modify the default startup query interval to 21 seconds for use with IGMP configuration of an interface named Local Area Connection, type:

set interface "Local Area Connection" startupqueryinterval=21

set global

Sets IGMP global parameters. Used without parameters, **set global** specifies that warnings are logged but that no other events related to IGMP are logged.

Syntax

set global [**LogLevel**=]{**none** | **error** | **warn** | **info**}

Parameters

[LogLevel=]{none | error | warn | info} : Required. Specifies which events should be logged. The **none** parameter specifies that no events related to IGMP should be logged. The **error** parameter specifies that only errors related to IGMP should be logged. The **warn** parameter specifies that only warnings related to IGMP should be logged. The **info** parameter specifies that all events related to IGMP should be logged.

show interface

Displays the IGMP configuration of the specified interface. Used without parameters, **show interface** displays IGMP configuration information for all interfaces.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

show global

Displays the global IGMP parameters.

Syntax

show global

Parameters

none

show ifstats

Displays IGMP statistics for the specified interface. Used without parameters, **show ifstats** shows IGMP statistics once for all interfaces.

Syntax

show ifstats [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

Parameters

[Index=]*Integer* : Specifies a number identifying the interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Examples

To show IGMP statistics every 5 seconds for an interface with an index value of 1001, type:

show ifstats index=1001 rr=5

show iftable

Shows IGMP host groups for a specified interface. Used without parameters, **show iftable** shows IGMP host groups for all interfaces once.

Syntax

show iftable [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

Parameters

[Index=]*Integer* : Specifies, by index number, the interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Examples

To show IGMP host groups every 5 seconds for an interface with an index value of 1001, type:

show iftable index=1001 rr=5

show grouptable

Shows the IGMP hosts group table for a multicast group. Used without parameters, **show**

grouptable shows the IGMP hosts group tables once for all multicast groups on all interfaces associated with the router.

Syntax

show grouptable [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

Parameters

[Index=]*Integer* : Specifies, by IP address, the multicast group for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Examples

To display the IGMP hosts group table every 5 seconds for an IGMP multicast group identified by the IP address of 192.168.100.239, type:

show grouptable index=192.168.100.239 rr=5

show rasgrouptable

Displays the hosts group table for a remote access client interface. Used without parameters, **show rasgrouptable** displays the hosts group tables once for all remote access client interfaces.

Syntax

show rasgrouptable [[**Index**=]*IPAddress*] [[**Rr**=]*Integer*]

Parameters

[Index=]*IPAddress* : Specifies, by IP address, the remote access client interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Examples

To display the hosts group table for a remote access client interface with an IP address of

10.10.1.100 and to refresh the information every five seconds, type:

show rasgrouptable index=10.10.1.100 rr=5

show proxygrouptable

Displays the IGMP hosts group table for an IGMP proxy interface. Used without parameters, **show proxygrouptable** displays information for all interfaces once.

Syntax

show proxygrouptable [[**InterfaceName=**]*InterfaceName*] [[**Rr**=]*Integer*]

Parameters

[InterfaceName=]*InterfaceName* : Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Examples

To display IGMP hosts group tables for all IGMP proxy interfaces and to refresh the information every five seconds, type:

show proxygrouptable rr=5

⇡Top of page

Netsh Routing IP Routerdiscovery commands

The following commands are available for use in the Netsh Routing IP Routerdiscovery context.

To view the command syntax, click a command:

add interface

Configures router discovery for the specified interface.

Syntax

add interface [**InterfaceName=**]*InterfaceName* [[**Disc=**]{**enable** | **disable**}] [[**MinInt=**]*Integer*] [[**MaxInt=**]*Integer*] [[**Life=**]*Integer*] [[**Level=**]*Integer*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to configure router discovery. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[Disc=]{enable | disable} : Specifies whether router discovery should occur on this interface. The **enable** parameter specifies that router discovery should occur, and the **disable** parameter specifies that router discovery should not occur.

[MinInt=]*Integer* : Specifies a minimum interval, in minutes, between when router discovery packets are sent.

[MaxInt=]*Integer* : Specifies a maximum interval, in minutes, between when router discovery packets are sent.

[Life=]*Integer* : Specifies the number of minutes for which a router discovery packet sent from this router or server is valid.

[Level=]*Integer* : Specifies a number indicating preference for using this interface as the default gateway. Where multiple interfaces are enabled for router discovery, a higher number used here indicates a higher level of preference.

Remarks

- When added, router discovery advertisements are sent at random times bounded by the set minimum and maximum intervals.

Examples

To enable router discovery for the Local Area Connection interface, specifying a minimum interval of 7 minutes, a maximum interval of 10 minutes, a life of 30 minutes for router discovery packets, and a zero (0) or no preference level, type:

add interface "Local Area Connection" enable 7 10 30 0

set interface

Configures router-discovery for an interface.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [[**Disc=**]{**enable** | **disable**}] [[**MinInt=**]*Integer*] [[**MaxInt=**]*Integer*] [[**Life=**]*Integer*] [[**Level=**]*Integer*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to configure router discovery. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[Disc=]{enable | disable} : Specifies whether router discovery should occur on this interface. The **enable** parameter specifies that router discovery should occur, and the **disable** parameter specifies that router discovery should not occur.

[MinInt=]*Integer* : Specifies, in minutes, a minimum interval between when router discovery packets are sent.

[MaxInt=]*Integer* : Specifies, in minutes, a maximum interval between when router discovery packets are sent.

[Life=]*Integer* : Specifies, in minutes, how long a router discovery packet sent from this router or server is valid.

[Level=]*Integer* : Specifies a number indicating preference for using this interface as the default gateway. A higher number used here specifies a higher level of preference.

Examples

To enable router discovery for the Local Area Connection interface, specifying a minimum interval of 7 minutes, a maximum interval of 10 minutes, a life of 30 minutes for router discovery packets, and a zero (0) or no preference level, type:

set interface "Local Area Connection" enable 7 10 30 0

delete interface

Deletes router-discovery configuration.

Syntax

delete interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to delete router-discovery configuration. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

Examples

To delete the router-discovery configuration for an interface named Local Area Connection, type:

delete interface name="Local Area Connection"

show interface

Displays router-discovery information. Used without parameters, **show interface** displays information for all interfaces.

Syntax

show interface [[**InterfaceName=**]*InterfaceName*]

Parameters

[InterfaceName=]*InterfaceName* : Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

Examples

To display router-discovery information for an interface named Local Area Connection, type:

show interface "Local Area Connection"

⇡Top of page

Netsh Routing IP RIP commands

The following commands are available for use in the Netsh Routing IP RIP context.

To view the command syntax, click a command:

add peerfilter

Adds a filter for servers that can be accepted as peers.

Syntax

add peerfilter [**Server**=]*IPAddress*

Parameters

[Server=]*IPAddress* : Required. Specifies, by IP address, the peer server whose traffic you want to filter.

add acceptfilter

Adds an acceptance filter for routes received on an interface.

Syntax

add acceptfilter [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1 IPAddress2*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[Addr=]*IPAddress1 IPAddress2* : Required. Specifies an IP address pairing. The first address is for the low IP address, and the second address is for the high IP address to use in building the filter.

Examples

To add an acceptance filter for an interface named Local Area Connection for an IP range of 10.0.0.2 to 10.11.21.154, type:

add acceptfilter "Local Area Connection" 10.0.0.2 10.11.21.154

add announcefilter

Adds a filter for routes announced on an interface.

Syntax

add announcefilter [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1 IPAddress2*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies the name of an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[Addr=]*IPAddress1 IPAddress2* : Required. Specifies an IP address pairing. The first address is for the low IP address, and the second address is for the high IP address to use in building the filter.

Examples

To add a filter for routes announced on an interface named Local Area Connection for a range of IP addresses from 10.0.0.2 to 10.11.21.154, type:

add announcefilter Local Area Connection 10.0.0.2 10.11.21.154

add neighbor

Adds a RIP neighbor on an interface.

Syntax

add neighbor [**InterfaceName=**]*InterfaceName* [**addr=**]*IPAddress*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[addr=]*IPAddress* : Required. Specifies the IP address of the RIP neighbor you want to add.

Examples

To add a RIP neighbor located at an IP address of 10.0.0.2 for an interface named Local Area Connection, type:

add neighbor "Local Area Connection" 10.0.0.2

add interface

Adds RIP to a specified interface.

Syntax

add interface [**InterfaceName=**]*InterfaceName* [[**Metric=**]*Integer*] [[**UpdateMode=**]{**demand** | **periodic**}] [[**Announce=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Accept=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Expire=**]*Integer*] [[**Remove=**]*Integer*] [[**Update=**]*Integer*] [[**AuthMode=**]{**authnone** | **authsimplepassword**}] [[**Tag=**]*Integer*] [[**Unicast=**]{**also** | **only** | **disable**}] [[**AccFiltMode=**]{**include** | **exclude** | **disable**}] [[**AnnFiltMode=**]{**include** | **exclude** | **disable**}] [[**Password=**]*String*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which to add RIP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[Metric=]*Integer* : Specifies a metric value for routes based on this interface.

[UpdateMode=]{demand | periodic} : Specifies when updates will occur. The **demand** parameter specifies that updates should occur on demand. The **periodic** parameter specifies that updates should occur at regular intervals.

[Announce=]{none | rip1 | rip1compat | rip2} : Specifies which routes should be announced. The **none** parameter specifies that all routes should be announced. The **rip1** parameter specifies that only RIP version 1 routes should be announced. The **rip1compat** parameter specifies that announcements should be made in RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be announced.

[Accept=]{none | rip1 | rip1compat | rip2} : Specifies which routes should be accepted. The **none** parameter specifies that all routes should be accepted. The **rip1** parameter specifies that only RIP version 1 routes should be accepted. The **rip1compat** parameter specifies that routes should be accepted in RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be accepted.

[Expire=]*Integer* : Specifies, in seconds, the route expiration interval.

[Remove=]*Integer* : Specifies, in seconds, the route removal interval.

[Update=]*Integer* : Specifies, in seconds, the full update interval.

[AuthMode=]{authnone | authsimplepassword} : Specifies whether authorization should be required. The **authnone** parameter specifies that no authorization should be required. The **authsimplepassword** parameter specifies that password authorization should be required.

[Tag=]*Integer* : Specifies the route tag.

[Unicast=]{also | only | disable} : Specifies the unicast mode. The **also** parameter specifies use also unicast mode. The **only** parameter specifies use only unicast mode. The **disable** parameter specifies that unicast mode should be disabled.

[AccFiltMode=]{include | exclude | disable} : Specifies whether acceptance filters should be included or disabled. The **include** parameter specifies that acceptance filters should be included. The **exclude** parameter specifies that acceptance filters should not be included. The **disable** parameter specifies that acceptance filters should be disabled.

[AnnFiltMode=]{include | exclude | disable} : Specifies whether announcement filters should be included or disabled. The **include** parameter specifies that announcement filters should be included. The **exclude** parameter specifies that announcement filters should not be included. The

disable parameter specifies that announcement filters should be disabled.

[Password=]*String* : Specifies a plaintext password no longer than 16 characters.

add helper

Installs a helper DLL under RIP.

Syntax

add helper *DLLName*

Parameters

***DLLName* :** Required. Specifies a DLL that supports this context.

delete peerfilter

Deletes a filter for an accepted peer server.

Syntax

delete peerfilter [**Server**=]*IPAddress*

Parameters

[Server=]*IPAddress* : Required. Specifies, by IP address, the peer server whose traffic you do not want to filter.

delete acceptfilter

Deletes an acceptance filter for routes received on a specified interface.

Syntax

delete acceptfilter [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1 IPAddress2*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[Addr=]*IPAddress1 IPAddress2* : Required. Specifies an IP address pairing. The first address is for the low IP address, and the second address is for the high IP address used in building the filter.

Examples

To delete an acceptance filter for the IP ranges of 10.0.0.2 to 10.11.21.154 on an interface named Local Area Connection, type:

delete acceptfilter "Local Area Connection" 10.0.0.2 10.11.21.154

delete announcefilter

Deletes an announcement filter set for an interface.

Syntax

delete announcefilter [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1 IPAddress2*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies the name of an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[Addr=]*IPAddress1 IPAddress2* : Required. Specifies an IP address pairing. The first address specifies the low IP address, and the second address specifies the high IP address used in building the filter.

Examples

To delete an announcement filter for the IP range 10.0.0.2 to 10.11.21.154 for an interface named Local Area Connection, type:

delete announcefilter "Local Area Connection" 10.0.0.2 10.11.21.154

delete neighbor

Deletes a RIP neighbor from an interface.

Syntax

delete neighbor [**InterfaceName=**]*InterfaceName IPAddress*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***IPAddress* :** Required. Specifies, by IP address, the RIP neighbor to delete.

Examples

To delete a RIP neighbor located at 10.0.0.2 for an interface named Local Area Connection, type:

delete neighbor "Local Area Connection" 10.0.0.2

delete interface

Removes RIP from the specified interface.

Syntax

delete interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, an interface from which you want to remove RIP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

delete helper

Removes a specified helper DLL from Netsh.

Syntax

delete helper *HelperName*

Parameters

***HelperName* :** Required. Specifies, by file name, the helper DLL to remove.

set interface

Configures RIP on a specified interface.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [[**Metric=**]*Integer*] [[**UpdateMode=**]{**demand** | **periodic**}] [[**Announce=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Accept=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Expire=**]*Integer*] [[**Remove=**]*Integer*] [[**Update=**]*Integer*] [[**AuthMode=**]{**authnone** | **authsimplepassword**}] [[**Tag=**]*Integer*] [[**Unicast=**]{**also** | **only** | **disable**}] [[**AccFiltMode=**]{**include** | **exclude** | **disable**}] [[**AnnFiltMode=**]{**include** | **exclude** | **disable**}] [[**Password=**]*String*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which to configure RIP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[Metric=]*Integer* : Specifies a metric value for routes based on this interface.

[UpdateMode=]{demand | periodic} : Specifies when updates should occur. The **demand** parameter specifies that updates should occur on demand. The **periodic** parameter specifies that updates should occur at regular intervals.

[Announce=]{none | rip1 | rip1compat | rip2} : Specifies which routes should be announced. The **none** parameter specifies that all routes should be announced. The **rip1** parameter specifies that only RIP version 1 routes should be announced. The **rip1compat** parameter specifies that announcements should be made in RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be announced.

[Accept=]{none | rip1 | rip1compat | rip2} : Specifies which routes should be accepted. The **none** parameter specifies that all routes should be accepted. The **rip1** parameter specifies that only RIP version 1 routes should be accepted. The **rip1compat** parameter specifies that acceptance should occur using RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be accepted.

[Expire=]*Integer* : Specifies, in seconds, how much time should elapse before the route expires.

[Remove=]*Integer* : Specifies, in seconds, how much time should elapse before the route is removed.

[Update=]*Integer* : Specifies, in seconds, how often full updates occur.

[AuthMode=]{authnone | authsimplepassword} : Specifies whether authorization should be required. The **authnone** parameter specifies that no authorization should be required. The **authsimplepassword** parameter specifies that password authorization should be required.

[Tag=]*Integer* : Specifies the route tag.

[Unicast=]{also | only | disable} : Specifies the unicast mode. The **also** parameter specifies use also unicast mode. The **only** parameter specifies use only unicast mode. The **disable** parameter specifies that unicast mode should be disabled.

[AccFiltMode=]{include | exclude | disable} : Specifies whether acceptance filters should be included. The **include** parameter specifies that acceptance filters should be included. The **exclude** parameter specifies that acceptance filters should not be included. The **disable** parameter specifies that acceptance filters should be disabled.

[AnnFiltMode=]{include | exclude | disable} : Specifies whether announcement filters should be included or disabled. The **include** parameter specifies that announcement filters should be included. The **exclude** parameter specifies that announcement filters should not be included. The **disable** parameter specifies that announcement filters should be disabled.

[Password=]*String* : Specifies a plaintext password no longer than 16 characters long.

Examples

To modify the RIP configuration on the interface named Local Area Connection, specifying a metric value of 2 and that updates should occur on demand, type:

```
set interface "Local Area Connection" metric=2 updatemode=periodic
```

set flags

Sets RIP-related flags for a specified interface.

Syntax

set flags [**InterfaceName=**]*InterfaceName* [**clear,**][**splithorizon,**][**poisonreverse,**][**triggeredupdates,**][**cleanupupdates**][**accepthostroutes,**][**acceptdefaultroutes,**][**senddefaultroutes,**][**nosubnetsummary**]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which to set flags. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[clear,][splithorizon,][poisonreverse,][triggeredupdates,][cleanupupdates,][accepthostroutes,][acceptdefaultroutes,][senddefaultroutes,][nosubnetsummary] : Specifies policies for RIP-related flags. The **clear** parameter specifies that all set RIP flags should be cleared. The **splithorizon** parameter specifies that RIP for split horizon should be enabled. The **poisonreverse** parameter specifies that RIP for poison reverse should be enabled. The **triggeredupdates** parameter specifies that RIP triggered updates should occur. The **cleanupupdates** parameter specifies that cleanup updates for RIP should occur. The **accepthostroutes** parameter specifies that host routes should be accepted. The **acceptdefaultroutes** parameter specifies that default routes should be accepted. The **senddefaultroutes** parameter specifies that default routes should be sent. The **nosubnetsummary** parameter specifies that subnet summarization should not occur.

Remarks

- The last policy parameter should not have a comma after it.

set global

Sets global RIP parameters.

Syntax

set global {[[**loglevel**=]{**none** | **error** | **warn** | **info**}] | [[**mintrig**=]*Integer*] | [[**peermode**=]{**include** | **exclude** | **disable**}]}

Parameters

[loglevel=]{none | error | warn | info} : Specifies which RIP events should be logged. The **none** parameter specifies that no RIP events should be logged. The **error** parameter specifies that only errors related to RIP should be logged. The **warn** parameter specifies that only warnings related to RIP should be logged. The **info** parameter specifies that all events related to RIP should be logged.

[mintrig=]*Integer* : Specifies, in minutes, the minimum amount of time that should elapse between triggers.

[peermode=]{include | exclude | disable} : Specifies whether peers should be included and whether peer mode should be disabled. The **include** parameter specifies that peers should be included.The **exclude** parameter specifies that peers should not be included.The **disable** parameter specifies that peer mode should be disabled.

show interface

Shows RIP configuration for the specified interface. Used without parameters, **show interface** displays configuration information for all interfaces.

Syntax

show interface [[**InterfaceName=**]*InterfaceName*]

Parameters

[InterfaceName=]*InterfaceName* : Specifies, by name, the interface whose information you want to display. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

show flags

Displays RIP flags set for a specified interface. Used without parameters, **show flags** displays information for all interfaces.

Syntax

show flags [[**InterfaceName=**]*InterfaceName*]

Parameters

[InterfaceName=]*InterfaceName* : Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

show global

Shows RIP global parameters.

Syntax

show global

Parameters

none

show ifstats

Displays RIP statistics for a specified interface. Used without parameters, **show ifstats** displays information once for all interfaces.

Syntax

show ifstats [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

Parameters

[Index=]*Integer* : Specifies, by index number, the interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Remarks

- To list the index value for each interface, use the **show interface** command in the Netsh Interface IP context.

Examples

To display RIP statistics every 5 seconds for an interface with an index value of 1, type:

show ifstats index=1 rr=5

show ifbinding

Displays RIP interface IP address binding. Used without parameters, **show ifbinding** displays information once for all interfaces.

Syntax

show ifbinding [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

Parameters

[Index=]*Integer* : Specifies, by index number, the interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Remarks

- To list the index value for each interface, use the **show interface** command in the Netsh Interface IP context.

Examples

To display the bindings every 5 seconds for an interface with an index value of 1, type:

show ifbinding index=1 rr=5

⇡Top of page

Netsh Routing IP Relay commands

The following commands are available for use in the Netsh Routing IP Relay context.

To view the command syntax, click a command:

add dhcpserver

Adds a DHCP server to the list of DHCP servers whose messages will be forwarded by DHCP Relay Agent.

Syntax

add dhcpserver *IPAddress*

Parameters

IPAddress **:** Required. Specifies, by IP address, the DHCP server you want to add.

add interface

Enables DHCP Relay Agent on the specified interface.

Syntax

add interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* **:** Required. Specifies, by name, the interface on which you want to enable DHCP Relay Agent. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

delete interface

Disables DHCP Relay Agent on the specified interface.

Syntax

delete interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which you want to disable DHCP Relay Agent. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

delete dhcpserver

Removes a DHCP server from the list of DHCP servers whose messages will be forwarded by DHCP Relay Agent.

Syntax

delete dhcpserver *IPAddress*

Parameters

***IPAddress* :** Required. Specifies, by IP address, the DHCP server to be deleted.

set global

Sets global logging parameters for DHCP Relay Agent configuration.

Syntax

set global [**LogLevel=**]{**none** | **error** | **warn** | **info**}

Parameters

[LogLevel=]{none | error | warn | info} : Required. Specifies which DHCP Relay Agent events should be logged. The **none** parameter specifies that no events related to DHCP Relay Agent should be logged. The **error** parameter specifies that only errors related to DHCP Relay Agent should be logged. The **warn** parameter specifies that only warnings related to DHCP Relay Agent should be logged. The **info** parameter specifies that all events related to DHCP Relay Agent should be logged.

set interface

Updates the DHCP Relay Agent configuration on an interface.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [**relaymode=**]{**enable** | **disable**} [[**maxhop=**]*Integer*] [[**minsecs=**]*Integer*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which you want to update the configuration of DHCP Relay Agent. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[relaymode=]{enable | disable} : Required. Specifies whether the DHCP Relay Agent is enabled or disabled. The **enable** parameter specifies that DHCP Relay Agent should be enabled for this interface. The **disable** parameter disables DHCP Relay Agent for this interface.

[maxhop=]*Integer* : Specifies the number of hops a DHCP packet can make before it should be dropped and should no longer be relayed.

[minsecs=]*Integer* : Specifies the minimum number of seconds from boot time that must appear in a DHCP packet before it should be sent to a DHCP server in the list of servers whose messages will be forwarded by DHCP Relay Agent.

Examples

To update the DHCP Relay Agent configuration on the interface named Local Area Connection with a maximum hop count of 16 and a value of 4 seconds for the minimum seconds from boot time, type:

set interface "Local Area Connection" enable 16 4

show interface

Displays configuration information for DHCP Relay Agent on the specified interface.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

show global

Displays DHCP Relay Agent global configuration.

Syntax

show global [[**Rr=**]*Integer*]

Parameters

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

show ifconfig

Displays DHCP Relay Agent configuration information for the specified interface. Used without parameters, **show ifconfig** displays information once for all interfaces where DHCP Relay Agent is enabled.

Syntax

show ifconfig [[**Index**=]*Integer*] [[**Rr=**]*Integer*]

Parameters

[Index=]*Integer* : Specifies, by index value, the interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

show ifbinding

Displays IP address bindings for the specified interface. Used without parameters, **show ifbinding** displays information once for all interfaces where DHCP relay is enabled for use.

Syntax

show ifbinding [[**Index**=]*Integer*] [[**Rr=**]*Integer*]

Parameters

[Index=]*Integer* : Specifies, by index value, the interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

show ifstats

Displays DHCP Relay Agent statistics for the specified interface. Used without parameters, **show ifstats** displays information once for all interfaces where DHCP relay is enabled for use.

Syntax

show ifstats [[**Index**=]*Integer*] [[**Rr=**]*Integer*]

Parameters

[Index=]*Integer* : Specifies, by index value, the interface for which you want to display information.

[Rr=]*Integer* : Specifies, in seconds, how often you want the information to be refreshed.

Top of page

Netsh Routing IP OSPF commands

The following commands are available for use in the Netsh Routing IP OSPF context.

To view the command syntax, click a command:

Note

- The OSPF protocol is not available on Windows XP 64-Bit Edition.

add range

Adds the IP address and mask pair that define a range of addresses belonging to this area.

Syntax

add range [**Areaid=**]*IPAddress* [**Range=**]*IPAddress SubnetMask*

Parameters

[Areaid=]*IPAddress* : Required. Specifies the IP address of the area to which the range is added.

[Range=]*IPAddress SubnetMask* : Required. Specifies the IP address and the mask of the range to add.

Remarks

- OSPF ranges are used to summarize the routes within the OSPF area.

add area

Adds an ID for the area.

Syntax

add area [**Areaid=**]*IPAddress*

Parameters

[Areaid=]*IPAddress* : Required. Specifies the IP address of the area to which you want to add the range.

Remarks

- The **Areaid** parameter is a 32-bit number expressed in dotted decimal notation that identifies the OSPF area. The area ID does not need to coincide with an IP address or an IP network ID. The area ID of 0.0.0.0 is reserved for the backbone. If the area represents a subnetted network, you can use the IP network number of the subnetted network for the area ID.

add virtif

Adds the virtual interface for a router.

Syntax

add virtif [**transareaid=**]*TransitAreaID* [**virtnbrid=**]*VirtualNeighborID*

Parameters

[transareaid=]*TransitAreaID* : Required. Specifies the IP address of the transit area used to exchange routing information between the backbone area border router and the area border router of the noncontiguous area.

[virtnbrid=]*VirtualNeighborID* : Required. Specifies, by IP address, the router ID of the virtual neighbor to add.

add neighbor

Adds an OSPF neighbor to the specified interface.

Syntax

add neighbor [**InterfaceName=**]*InterfaceName* [**addr**=]*IPAddress* [**nbraddr**=]*IPAddress* [**nbrprio=**]*Priority*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to which you want to add a neighbor. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[addr=]*IPAddress* : Required. Specifies the OSPF interface, by IP address.

[nbraddr=]*IPAddress* : Required. Specifies the IP address of the neighbor you want to add.

[nbrprio=]*Priority* : Required. Specifies the OSPF router priority of the neighbor you want to add.

add interface

Configures OSPF on the specified interface.

Syntax

add interface [**InterfaceName=**]*InterfaceName* [**areaid**=]*IPAddress* [**addr**=]*IPAddress* [**mask**=] *SubnetMask*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which you want to add OSPF. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[areaid=]*IPAddress* : Required. Specifies the IP address of the area to which the interface belongs.

[addr=]*IPAddress* : Required. Specifies the IP address of the interface to add.

[mask=]*SubnetMask* : Required. Specifies the subnet mask of the interface to add.

add routefilter

Adds an OSPF route filter.

Syntax

add routefilter [**Filter=**]*IPAddress SubnetMask*

Parameters

[Filter=]*IPAddress SubnetMask* : Required. Specifies, by IP address and subnet mask, the route you want to filter.

add protofilter

Adds an OSPF protocol filter.

Syntax

add protofilter [filter=]{**autostatic** | **local** | **rip** | **snmp** | **nondod** | **static**}

Parameters

[filter=]{autostatic | local | rip | snmp | nondod | static} : Required. Specifies the protocol filter you want to add. The **autostatic** parameter specifies a RIP autostatic route. The **local** parameter specifies a route added by the network to which the computer is connected. The **rip** parameter specifies that the filter is added using RIP. The **snmp** parameter specifies that the route is added using SNMP. The **nondod** parameter specifies an RTM route that is present only when a DOD interface is connected. The **static** parameter specifies an RTM route.

Remarks

- The **add protofilter** command is used on an OSPF router acting as an Autonomous System Boundary Router. The filters you add filter other route sources by protocol source, such as static routes and RIP, so that they are not redistributed into the OSPF Autonomous System.

install

Installs OSPF.

Syntax

install

Parameters

none

delete range

Deletes the specified range for the area ID.

Syntax

delete range [**Areaid=**]*IPAddress* [**Range=**]*IPAddress SubnetMask*

Parameters

[Areaid=]*IPAddress* : Required. Specifies, by IP address, the area you want to delete.

[Range=]*IPAddress SubnetMask* : Required. Specifies the IP address and the subnet mask of the area range you want to delete.

delete area

Deletes the specified area.

Syntax

delete area [**Areaid=**]*IPAddress*

Parameters

[Areaid=]*IPAddress* : Required. Specifies, by IP address, the area you want to delete.

Remarks

The area ID is a 32-bit number expressed in dotted decimal notation that identifies the OSPF area. The area ID does not need to coincide with an IP address or an IP network ID. The area ID of 0.0.0.0 is reserved for the backbone. If the area represents a subnetted network, you can use the IP network number of the subnetted network for the area ID.

delete virtif

Deletes the specified virtual interface.

Syntax

delete virtif [**transareaid=**]*TransitAreaID* [**virtnbrid=**]*VirtualNeighborID*

Parameters

[transareaid=]*TransitAreaID* : Required. Specifies the IP address of the transit area used to exchange routing information between the backbone area border router and the area border router of the noncontiguous area.

[virtnbrid=]*VirtualNeighborID* : Required. Specifies the router ID of the virtual neighbor to delete.

delete neighbor

Deletes the specified OSPF neighbor.

Syntax

delete neighbor [**InterfaceName=**]*InterfaceName* [**addr**=]*IPAddress* [**nbraddr**=]*IPAddress* [**nbrprio=**]*Priority*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from which you want to delete a neighbor. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[addr=]*IPAddress* : Required. Specifies, by IP address, the interface from which you want to delete a neighbor.

[nbraddr=]*IPAddress* : Required. Specifies the IP address of the neighbor to delete.

[nbrprio=]*Priority* : Required. Specifies the OSPF router priority of the neighbor to delete.

delete interface

Deletes OSPF from the specified interface.

Syntax

delete interface [**InterfaceName=**]*InterfaceName* [**addr**=]*IPAddress* [**mask**=]*SubnetMask*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from which you want to delete OSPF. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[addr=]*IPAddress* : Required. Specifies, by IP address, the interface to delete.

[mask=]*SubnetMask* : Required. Specifies, by subnet mask, the interface to delete.

delete routefilter

Deletes the specified OSPF route filter.

Syntax

delete routefilter [**Filter=**]*IPAddress SubnetMask*

Parameters

[Filter=]*IPAddress SubnetMask* : Required. Specifies the IP address and the subnet mask of the route filter you want to delete.

delete protofilter

Deletes the specified OSPF protocol filter.

Syntax

delete protofilter [**filter=**]{**autostatic** | **local** | **rip** | **snmp** | **nondod** | **static**}

Parameters

[filter=]{autostatic | local | rip | snmp | nondod | static} : Required. Specifies the protocol filter you want to delete. The **autostatic** parameter specifies a RIP autostatic route. The **local** parameter specifies a route added by the network to which the computer is connected. The **rip** parameter specifies that the filter is added using RIP. The **snmp** parameter specifies that the route is added using SNMP. The **nondod** parameter specifies an RTM route that is present only when a DOD interface is connected. The **static** parameter specifies an RTM route.

Remarks

- The **delete protofilter** command is used on an OSPF router acting as an Autonomous System Boundary Router. The filters you add filter other route sources by protocol source, such as static routes and the Routing Information Protocol (RIP), so that they are not redistributed into the OSPF Autonomous System.

uninstall

Deletes OSPF.

Parameters

none

set area

Changes parameters for the specified area.

Syntax

set area [**areaid=**]*AreaID* [[**auth=**]{**none** | **password**}] [[**stubarea**=]{**yes** | **no**}] [[**metric**=]*StubMetric*] [[**sumadv**=]{**yes** | **no**}]

Parameters

[areaid=]*AreaID* : Required. Specifies, by IP address, the OSPF area for which you want to change parameters.

[auth=]{none | password} : Specifies whether passwords are required for the area. If passwords are required for an area, all interfaces that are in the same area and on the same network segment must use the same password, and interfaces that are in the same area but on different network segments can have different passwords. By default, passwords are required and the password is 12345678. Passwords are transmitted in plaintext format, so this option is for identification, not security.

[stubarea=]{yes | no} : Specifies whether this area is configured as a stub area, which is an OSPF area that does not enumerate external routes. Routes from outside the OSPF Autonomous System (AS) are not flooded into or through stub areas. Routing to AS external destinations in these areas is based only on a summary default route. This reduces the overhead for the routers in the stub area.

[metric=]*StubMetric* : Specifies the cost of the summary default route that the router advertises into the stub area.

[sumadv=]{yes | no} : Specifies whether to import summary advertisements.

set virtif

Configures parameters for the specified virtual interface.

Syntax

set virtif [**transareaid=**]*TransitAreaID* [**virtnbrid=**]*VirtualNeighborID* [[**transdelay=**]*TransitDelay*] [[**retrans=**]*ReTransmitInterval*] [[**hello=**]*HelloInterval*] [[**dead=**]*DeadInterval*] [[**password=**]*Password*]

Parameters

[transareaid=]*TransitAreaID* : Required. Specifies the IP address of the transit area.

[virtnbrid=]*VirtualNeighborID* : Required. Specifies the router ID of the virtual neighbor.

[transdelay=]*TransitDelay* : Specifies the estimated number of seconds it takes to transmit a link state update packet over this interface. If you do not specify this parameter, the estimated delay is set to 1 second.

[retrans=]*ReTransmitInterval* : Specifies the number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface. A typical value for a local area network is 5 seconds.

[hello=]*HelloInterval* : Specifies, in seconds, the interval between transmissions of hello packets by the router on the interface. This setting must be the same for all routers that are connected to the same network. A typical value for a local area network is 10 seconds.

[dead=]*DeadInterval* : Specifies, in seconds, how long a router can fail to respond before a neighboring router should consider this router is malfunctioning. This setting should be an integral multiple of the hello interval (commonly 4). This value must be the same for all OSPF router interfaces that are connected to the same network segment.

[password=]*Password* : Specifies the password for this interface, if passwords are required for the area. All interfaces that are in the same area and on the same network must use the same password. By default, passwords are required and the password is 12345678.

set interface

Configures OSPF parameters for an interface.

Syntax

set interface [**interfacename=**]*InterfaceName* [**addr=**]*IPAddress* [**mask=**]*SubnetMask* [[**state=**]{**enable** | **disable**}] [[**areaid=**]*AreaID*] [[**type=**]{**nbma** | **p2p** | **broadcast**}] [[**prio=**]*Priority*] [[**transdelay=**]*TransitDelay*] [[**retrans=**]*ReTransmitInterval*] [[**hello=**]*HelloInterval*] [[**dead=**] *DeadInterval*] [[**poll=**]*PollInterval*] [[**metric=**]*Metric*] [[**password=**]*Password*] [[**mtu=**]*MTUSize*]

Parameters

[interfacename=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[addr=]*IPAddress* : Required. Specifies the IP address of the interface.

[mask=]*SubnetMask* : Required. Specifies the subnet mask of the interface.

[state=]{enable | disable} : Specifies whether the interface is enabled. The **enable** parameter enables the interface. The **disable** parameter disables the interface.

[areaid=]*AreaID* : Specifies the IP address of the interface area.

[type=]{nbma | p2p | broadcast} : Specifies the type of OSPF interface for which you want to configure parameters. The **nbma** parameter specifies a non-broadcast multiple access (NBMA) OSPF interface. The **p2p** parameter specifies a point-to-point OSPF interface. The **broadcast** parameter specifies a broadcast OSPF interface.

[prio=]*Priority* : Specifies the interface priority.

[transdelay=]*TransitDelay* : Specifies the estimated number of seconds it takes to transmit a link state update packet over this interface. If you do not specify this parameter, the estimated delay is set to 1 second.

[retrans=]*ReTransmitInterval* : Specifies the number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface. A typical value for a local area network is 5 seconds.

[hello=]*HelloInterval* : Specifies, in seconds, the interval between transmissions of hello packets by the router on the interface. This setting must be the same for all routers that are connected to the same network. A typical value for a local area network is 10 seconds.

[dead=]*DeadInterval* : Specifies, in seconds, how long a router can fail to respond before a neighboring router should consider this router to be malfunctional. This setting should be an integral multiple of the hello interval (commonly 4). This value must be the same for all OSPF router interfaces connected to the same network segment.

[poll=]*PollInterval* : Specifies the number of seconds between OSPF network polls for non-broadcast multiple access (NBMA) interfaces only. You should set the poll interval to be at least twice as long as the dead interval. A typical value for an X.25 network is two minutes.

[metric=]*Metric* : Specifies the cost of sending a packet out of this interface. The value set here is advertised as the link cost for this interface in the link state advertisement of the router. Faster interfaces usually have lower costs. The maximum setting is 32,767.

[password=]*Password* : Specifies the password for this interface, if passwords are required for the area. All interfaces that are in the same area and on the same network must use the same password. By default, passwords are required and the password is 12345678.

[mtu=]*MTUSize* : Specifies, in bytes, the maximum size of IP packets carrying OSPF information that can be sent without fragmentation. The default IP MTU for an Ethernet network is 1,500 bytes.

set global

Configures global OSPF parameters.

Syntax

set global [**routerid=**]*IPAddress* [**asborder=**]{**yes** | **no**} [[**loglevel=**]{**none** | **error** | **warn** | **info**}]

Parameters

[routerid=]*IPAddress* : Required. Specifies, by IP address, the router for which you want to configure parameters.

[asborder=]{yes | no} : Required. Specifies whether the router acts as an Autonomous System (AS) boundary router.

[loglevel=]{none | error | warn | info} : Specifies which events should be logged. The **none** parameter specifies that no events related to OSPF should be logged. The **error** parameter specifies that only errors related to OSPF should be logged. The **warn** parameter specifies that only warnings related to OSPF should be logged. The **info** parameter specifies that all events related to OSPF should be logged.

set routefilter

Configures the OSPF route filter action.

Syntax

set routefilter [**action=**]{**drop** | **accept**}

Parameters

[action=]{drop | accept} : Required. Specifies whether the route filter drops or accepts packets. The **drop** parameter specifies that packets should be dropped. The **accept** parameter specifies that packets should be accepted.

set protofilter

Configures OSPF protocol filter action.

Syntax

set protofilter [**action=**]{**drop** | **accept**}

Parameters

[action=]{drop | accept} : Required. Specifies whether the protocol filter accepts or drops packets. The **drop** parameter specifies that packets should be dropped. The **accept** parameter specifies that packets should be accepted.

show global

Displays global OSPF parameters.

Syntax

show global

Parameters

none

show area

Displays area parameters.

Syntax

show area

Parameters

none

show virtif

Displays parameters for all virtual interfaces.

Syntax

show virtif

Parameters

none

show interface

Displays OSPF configuration information for the specified interface.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

show routefilter

Displays information about the OSPF route filter.

Syntax

show routefilter

Parameters

none

show protofilter

Displays information about the OSPF protocol filter.

Syntax

show protofilter

Parameters

none

show area

Displays information about OSPF areas.

Syntax

show area

Parameters

none

show virtif

Displays information about OSPF virtual interfaces.

Syntax

show virtif

Parameters

none

⇡Top of page

Netsh Routing IP Autodhcp commands

The following commands are available for use in the Netsh Routing IP Autodhcp context.

To view the command syntax, click a command:

add exclusion

Adds an exclusion to the DHCP allocator scope.

Syntax

add exclusion *IPAddress*

Parameters

IPAddress **:** Required. Specifies an IP address to exclude from the DHCP allocator scope.

delete exclusion

Deletes an exclusion from the DHCP allocator scope.

Syntax

delete exclusion *IPAddress*

Parameters

IPAddress **:** Required. Specifies an IP address already excluded from the DHCP allocator scope.

install

Installs the routing protocol under IP.

Syntax

install

Parameters

none

set global

Configures global parameters for DHCP allocation.

Syntax

set global {[[**scopenetwork=**]*IPAddress*] | [[**scopemask=**]*SubnetMask*] | [[**leasetime=**]*Minutes*] | [[**loglevel=**]{**none** | **error** | **warn** | **info**}]}

Parameters

[scopenetwork=]*IPAddress* : Specifies the IP address for the DHCP allocator scope.

[scopemask=]*SubnetMask* : Specifies the subnet mask associated with the IP address for the scope.

[leasetime=]*Minutes* : Specifies, in minutes, the lease duration time.

[LogLevel=]{none | error | warn | info} : Specifies which events should be logged. The **none** parameter specifies that no events related to DHCP should be logged. The **error** parameter specifies that only errors related to DHCP should be logged. The **warn** parameter specifies that only warnings related to DHCP should be logged. The **info** parameter specifies that all events related to DHCP should be logged.

Examples

To set the IP network for DHCP allocation to 10.10.10.0, with an associated scope subnet mask of 255.255.255.0, a lease time of 11520 minutes (8 days), and logging of information for all related events, type:

set global 10.10.10.0 255.255.255.0 11520 info

To adjust only the lease time globally for all scopes on all interfaces to a value of 4320 minutes (3 days), type:

set global leasetime=4320

set interface

Configures DHCP allocator parameters for the specified interface.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [**mode=**]{**enable** | **disable**}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[mode=]{enable | disable} : Required. Specifies whether DHCP allocation is enabled or disabled for the interface.

show global

Displays the DHCP allocator global configuration.

Syntax

show global

Parameters

none

show interface

Displays the DHCP allocator configuration for the specified interface.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

uninstall

Removes the routing protocol under IP.

Syntax

uninstall

Parameters

none

⇡Top of page

Netsh Routing IP Dnsproxy commands

The following commands are available for use in the Netsh Routing IP Dnsproxy context.

To view the command syntax, click a command:

set global

Sets global DNS proxy parameters.

Syntax

set global [**querytimeout=**]*Integer* [[**dnsmode=**]{**enable** | **disable**}] [[**winsmode=**]{**enable** | **disable**}] [[**loglevel=**]{**none** | **error** | **warn** | **info**}]

Parameters

[querytimeout=]*Integer* : Required. Specifies a timeout, in seconds, for proxied DNS queries.

[dnsmode=]{enable | disable} : Specifies whether DNS proxy operation is enabled or disabled.

[winsmode=]{enable | disable} : Specifies whether WINS proxy operation is enabled or disabled.

[loglevel=]{none | error | warn | info} : Specifies which events should be logged. The **none** parameter specifies that no events related to DNS proxy should be logged. The **error** parameter specifies that only errors related to DNS proxy should be logged. The **warn** parameter specifies that only warnings related to DNS proxy should be logged. The **info** parameter specifies that all events related to DNS proxy should be logged.

Examples

To specify that queries should time out after 10 seconds, to enable DNS proxy operation and WINS proxy operation, and to prevent logging of DNS proxy events, type:

set global 10 enable enable none

install

Installs the routing protocol under IP.

Syntax

install

Parameters

none

set interface

Configures DNS proxy parameters for an interface.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [[**mode=**]{**enable** | **disable** | **default**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{enable | disable | default} : Specifies whether DNS proxy is enabled, disabled, or set to use the default for the specified interface. The **enable** parameter specifies that DNS proxy is enabled. The **disable** parameter specifies that DNS proxy is disabled. The **default** parameter specifies that DNS proxy is set according to the default for the interface.

show global

Displays the DNS proxy global configuration.

Syntax

show global

Parameters

none

show interface

Displays the DNS proxy configuration for the specified interface.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

uninstall

Removes the routing protocol under IP.

Syntax

uninstall

Parameters

none

⇡Top of page

Netsh Routing IP NAT commands

The following commands are available for use in the Netsh Routing IP NAT context.

To view the command syntax, click a command:

add addressmapping

Adds an IP address mapping to the network address translation (NAT) address pool for the specified interface.

Syntax

add addressmapping [**InterfaceName=**]*InterfaceName* [**public=**]*IPAddress* [**private=**] *IPAddress* [**inboundsessions=**]{**enable** | **disable**}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which you want to use network address translation. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[public=]*IPAddress* : Required. Specifies the IP address used on a public network.

[private=]*IPAddress* : Required. Specifies the IP address in the pooled range of addresses available for use on the private network.

[inboundsessions=]{enable | disable} : Required. Specifies whether inbound sessions are enabled or disabled. The **enable** parameter specifies that inbound sessions are enabled. The **disable** parameter specifies that inbound sessions are disabled.

Examples

To map the public IP address 10.11.11.1 to the private IP address 10.10.10.1 and to disable inbound sessions for this NAT table mapping, type:

add addressmapping "Local Area Connection" 10.11.11.1 10.10.10.1 disable

add addressrange

Adds an address range to the NAT address pool for the specified interface.

Syntax

add addressrange [**InterfaceName=**]*InterfaceName* [**start=**]*IPAddress* [**end=**]*IPAddress* [**mask=**]*SubnetMask*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to whose address pool you want to add an address range.

[start=]*IPAddress* : Required. Specifies the starting IP address for the address range.

[end=]*IPAddress* : Required. Specifies the ending IP address for the address range.

[mask=]*SubnetMask* : Required. Specifies the IP subnet mask associated with the network range bounded by the start and end IP addresses.

add interface

Configures NAT on the specified interface.

Syntax

add interface [**InterfaceName=**]*InterfaceName* [[**mode=**]{**full** | **addressonly** | **private**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which you want to configure NAT. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{full | addressonly | private} : Specifies whether the interface uses full, addressonly, or private mode.

add portmapping

Adds a protocol port mapping on the NAT interface.

Syntax

add portmapping [**InterfaceName=**]*InterfaceName* [**proto=**]{**tcp** | **udp**} [**publicip=**]{*IPAddress* | **0.0.0.0**} [**publicport=**]*Integer* [**privateip=**]*IPAddress* [**privateport=**]*Integer*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to add a port mapping. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[proto=]{tcp | udp} : Required. Specifies whether the protocol type is set to TCP or UDP.

[publicip=]{*IPAddress* | 0.0.0.0} : Required. Specifies an external IP address on the public network or 0.0.0.0 to indicate any IP address not specified within the private network address range.

[publicport=]*Integer* : Required. Specifies the public protocol port by using a number from 0 to 9999.

[privateip=]*IPAddress* : Required. Specifies an IP address within the private network range.

[privateport=]*Integer* : Required. Specifies the private protocol port by using a number from 0 to

9999.

delete addressmapping

Deletes an address mapping from the NAT address pool for the specified interface.

Syntax

delete addressmapping [**InterfaceName=**]*InterfaceName* [[**public=**]*IPAddress*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to delete an address mapping. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[public=]*IPAddress* : Specifies the IP address to be deleted. If you do not specify an address, all address mappings from the NAT address pool are deleted from the interface.

delete addressrange

Deletes an address range from the NAT address pool for the specified interface.

Syntax

delete addressrange [**InterfaceName=**]*InterfaceName* [**start=**]*IPAddress*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from which you want to delete an address range. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[start=]*IPAddress* : Required. Specifies the starting IP address of the range you want to delete.

delete interface

Removes network address translation (NAT) from the specified interface.

Syntax

delete interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to remove the use of NAT. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

delete portmapping

Deletes a protocol port mapping from the specified NAT-enabled interface.

Syntax

delete portmapping [**InterfaceName=**]*InterfaceName* [[**proto=**]{**tcp** | **udp**}] [**publicip=**]{*IPAddress* | **0.0.0.0**}] [[**publicport=**]*Integer*] [[**privateip=**]*IPAddress*] [**privateport=**]*Integer*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to delete a port mapping. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[proto=]{tcp | udp} : Specifies the protocol type. The **tcp** parameter specifies that the protocol type is Transmission Control Protocol. The **udp** parameter specifies that the protocol type is User Datagram Protocol.

[publicip=]{*IPAddress* | 0.0.0.0} : Specifies an external IP address on the public network or 0.0.0.0 to indicate any IP address not specified within the private network address range.

[publicport=]*Integer* : Specifies the public protocol port by using a number from 0 to 9999.

[privateip=]*IPAddress* : Specifies an IP address within the private network range.

[privateport=]*Integer* : Specifies the private protocol port by using a number from 0 to 9999.

set global

Sets global parameters for NAT.

Syntax

set global {[[**tcptimeoutmins=**]*Integer*] | [[**udptimeoutmins=**]*Integer*] | [[**LogLevel**=]{**none** | **error** | **warn** | **info**}]}

Parameters

[tcptimeoutmins=]*Integer* : Specifies, in minutes, the timeout value for TCP mappings.

[udptimeoutmins=]*Integer* : Specifies, in minutes, the timeout value for UDP mappings.

[LogLevel=]{none | error | warn | info} : Specifies which events should be logged. The **none** parameter specifies that no events related to NAT should be logged. The **error** parameter specifies that only errors related to NAT should be logged. The **warn** parameter specifies that only warnings related to NAT should be logged. The **info** parameter specifies that all events related to NAT should be logged.

set interface

Configure NAT parameters for the specified interface.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [**mode=**]{*Full* | *AddressOnly* | *Private*}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{*Full* | *AddressOnly* | *Private*} : Required. Specifies whether the interface should be enabled for full, addressonly, or private mode.

show global

Displays network address translation (NAT) global configuration.

Syntax

show global

Parameters

none

show interface

Displays network address translation (NAT) configuration for the specified interface.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

⇡Top of page

Netsh Routing IPX commands

The following commands are available for use in the Netsh Routing IPX context.

To view the command syntax, click a command:

show interface

Displays configuration information for the specified interface.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

add interface

Adds interface entries to the IPX configuration.

Syntax

add interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface whose entries you want to add. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

delete interface

Deletes interface entries from the IPX configuration.

Syntax

delete interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface whose entries you want to delete. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

set interface

Updates the IPX configuration on the specified interface.

Syntax

set interface [**InterfaceName=**]*InterfaceName* {[[**admstate=**]{**enabled** | **disabled**}] | [[**wanprotocol=**]{**ppp** | **ipxwan**}]}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to update IPX configuration. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[admstate=]{enabled | disabled} : Specifies whether admstate is enabled or disabled on the interface.

[wanprotocol=]{ppp | ipxwan} : Specifies the protocol of the interface, PPP or ipxwan.

show route

Displays route information for a specified network address.

Syntax

show route *NetworkAddress*

Parameters

***NetworkAddress* :** Required. Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

show staticroute

Displays configuration information.

Syntax

show staticroute [**InterfaceName=**]*InterfaceName NetworkAddress*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***NetworkAddress* :** Required. Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

add staticroute

Adds entries to the IPX configuration of the specified interface.

Syntax

add staticroute [**InterfaceName=**]*InterfaceName* {[*NetworkAddress*] | [[**nexthopmacaddress=**]*MacAddress*] | [[**ticks=**]*Integer*] | [[**hops=**]*Integer*]}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to whose IPX configuration you want to add entries. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***NetworkAddress* :** Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[nexthopmacaddress=]*MacAddress* : Specifies a 6-byte next hop mac address no longer than 12 hexadecimal digits. Leading 0s are optional.

[ticks=]*Integer* : Specifies the tick count as a decimal integer.

[hops=]*Integer* : Specifies the hop count as a decimal integer.

delete staticroute

Deletes an entry from the IPX configuration of the specified interface.

Syntax

delete staticroute [**InterfaceName=**]*InterfaceName* {[*NetworkAddress*] | [[**nexthopmacaddress=**]*MacAddress*] | [[**ticks=**]*Integer*] | [[**hops=**]*Integer*]}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from which you want to delete an entry. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

***NetworkAddress* :** Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[nexthopmacaddress=]*MacAddress* : Specifies a 6-byte next hop mac address no longer than 12 hexadecimal digits. Leading 0s are optional.

[ticks=]*Integer* : Specifies the tick count as a decimal integer.

[hops=]*Integer* : Specifies the hop count as a decimal integer.

set staticroute

Updates the IPX configuration of the specified interface.

Syntax

set staticroute [**InterfaceName=**]*InterfaceName* {[*NetworkAddress*] | [[**nexthopmacaddress=**] *MacAddress*] | [[**ticks=**]*Integer*] | [[**hops=**]*Integer*]}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface whose IPX configuration you want to update. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

***NetworkAddress* :** Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[nexthopmacaddress=]*MacAddress* : Specifies a 6-byte next hop mac address no longer than 12 hexadecimal digits. Leading 0s are optional.

[ticks=]*Integer* : Specifies the tick count as a decimal integer.

[hops=]*Integer* : Specifies the hop count as a decimal integer.

show service

Displays service name configuration information. Used without parameters, **show service** displays all service types and service names.

Syntax

show service [*ServiceType*] [*ServiceName*]

Parameters

***ServiceType* :** Specifies a 2-byte service type no longer than 4 hexadecimal digits. Leading 0s are optional.

***ServiceName* :** Specifies, by name, the service for which you want to display information.

show staticservice

Displays configuration information about static route service.

Syntax

show staticservice [**InterfaceName=**]*InterfaceName* [*ServiceType*] [*ServiceName*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***ServiceType* :** Specifies a 2-byte service type no longer than 4 hexadecimal digits. Leading 0s are optional.

***ServiceName* :** Specifies a service name.

add staticservice

Adds entries to the IPX configuration of a specified interface.

Syntax

add staticservice [**InterfaceName=**]*InterfaceName* {[*ServiceType*] | [*ServiceName*] | [*NetworkAddress*] | [[**node=**]*NodeAddress*] | [[**socket=**]*SocketAddress*] | [[**hops=**]*Integer*]}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to whose configuration you want to add entries. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

***ServiceType* :** Specifies a 2-byte service type no longer than 4 hexadecimal digits. Leading 0s are optional.

***ServiceName* :** Specifies a service name.

***NetworkAddress* :** Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[node=]*NodeAddress* : Specifies a 6-byte node address no longer than 12 hexadecimal digits. Leading 0s are optional.

[socket=]*SocketAddress* : Specifies a 2-byte socket address no longer than 4 hexadecimal digits. Leading 0s are optional.

[hops=]*Integer* : Specifies the hop count as a decimal integer.

delete staticservice

Deletes an entry from the IPX configuration of a specified interface.

Syntax

delete staticservice [**InterfaceName=**]*InterfaceName* {[*ServiceType*] | [*ServiceName*] |

[*NetworkAddress*] | [[**node=**]*NodeAddress*] | [[**socket=**]*SocketAddress*] | [[**hops=**]*Integer*]}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from whose IPX configuration you want to delete an entry. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***ServiceType* :** Specifies a 2-byte service type no longer than 4 hexadecimal digits. Leading 0s are optional.

***ServiceName* :** Specifies a service name.

***NetworkAddress* :** Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[node=]*NodeAddress* : Specifies a 6-byte node address no longer than 12 hexadecimal digits. Leading 0s are optional.

[socket=]*SocketAddress* : Specifies a 2-byte socket address no longer than 4 hexadecimal digits. Leading 0s are optional.

[hops=]*Integer* : Specifies the hop count as a decimal integer.

set staticservice

Updates the IPX static route configuration.

Syntax

set staticservice [**InterfaceName=**]*InterfaceName* {[*ServiceType*] | [*ServiceName*] | [*NetworkAddress*] | [[**node=**]*NodeAddress*] | [[**socket=**]*SocketAddress*] | [[**hops=**]*Integer*]}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface whose configuration you want to update. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***ServiceType* :** Specifies a 2-byte service type no longer than 4 hexadecimal digits. Leading 0s are optional.

***ServiceName* :** Specifies a service name.

***NetworkAddress* :** Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[node=]*NodeAddress* : Specifies a 6-byte node address no longer than 12 hexadecimal digits. Leading 0s are optional.

[socket=]*SocketAddress* : Specifies a 2-byte socket address no longer than 4 hexadecimal digits. Leading 0s are optional.

[hops=]*Integer* : Specifies the hop count as a decimal integer.

set filter

Updates the IPX router filter configuration.

Syntax

set filter [**InterfaceName=**]*InterfaceName* [[**mode=**]{**input** | **output**}] [[**action=**]{**permit** | **deny**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface whose configuration you want to update. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Specifies the mode, input or output.

[action=]{permit | deny} : Specifies the action, permit or deny.

show filter

Displays IPX filter configuration information for the specified interface.

Syntax

show filter [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

add filter

Adds router filter entries to the IPX configuration.

Syntax

add filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [[**srcnet=**]*SourceSubnet*] [[**srcnode=**]*SourceNode*] [[**srcsocket=**]*SourceSocket*] [[**dstnet=**]*DestinationSubnet*] [[**dstnode=**]*DestinationNode*] [[**dstsocket=**]*DestinationSocket*] [[**pkttype=**]*PacketType*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to which you want to add filter entries. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Required. Specifies the mode.

[srcnet=]*SourceSubnet* : Specifies a 4-byte network mask no longer than 8 hexadecimal digits. Leading 0s are optional.

[srcnode=]*SourceNode* : Specifies a 6-byte node address no longer than 8 hexadecimal digits. Leading 0s are optional.

[srcsocket=]*SourceSocket* : Specifies a 2-byte socket address no longer than 8 hexadecimal digits. Leading 0s are optional.

[dstnet=]*DestinationSubnet* : Specifies a 4-byte network mask no longer than 8 hexadecimal digits. Leading 0s are optional.

[dstnode=]*DestinationNode* : Specifies a 6-byte node address no longer than 8 hexadecimal digits. Leading 0s are optional.

[dstsocket=]*DestinationSocket* : Specifies a 2-byte socket address no longer than 8 hexadecimal digits. Leading 0s are optional.

[pkttype=]*PacketType* : Specifies a 1-byte packet type no longer than 8 hexadecimal digits. Leading 0s are optional.

delete filter

Deletes router filter entries from the IPX configuration.

Syntax

delete filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [[**srcnet=**]*SourceSubnet*] [[**srcnode=**]*SourceNode*] [[**srcsocket=**]*SourceSocket*] [[**dstnet=**]*DestinationSubnet*] [[**dstnode=**]*DestinationNode*] [[**dstsocket=**]*DestinationSocket*] [[**pkttype=**]*PacketType*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from which you want to delete a filter entry. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Required. Specifies the mode.

[srcnet=]*SourceSubnet* : Specifies a 4-byte network mask no longer than 8 hexadecimal digits. Leading 0s are optional.

[srcnode=]*SourceNode* : Specifies a 6-byte node address no longer than 8 hexadecimal digits. Leading 0s are optional.

[srcsocket=]*SourceSocket* : Specifies a 2-byte socket address no longer than 8 hexadecimal digits. Leading 0s are optional.

[dstnet=]*DestinationSubnet* : Specifies a 4-byte network mask no longer than 8 hexadecimal digits. Leading 0s are optional.

[dstnode=]*DestinationNode* : Specifies a 6-byte node address no longer than 8 hexadecimal digits. Leading 0s are optional.

[dstsocket=]*DestinationSocket* : Specifies a 2-byte socket address no longer than 8 hexadecimal digits. Leading 0s are optional.

[pkttype=]*PacketType* : Specifies a 1-byte packet type no longer than 8 hexadecimal digits. Leading 0s are optional.

show global

Displays global IPX configuration information.

Syntax

show global

Parameters

none

set global

Configures global IPX settings.

Syntax

set global [**LogLevel**=]{**none** | **error** | **warn** | **info**}

Parameters

[LogLevel=]{none | error | warn | info} : Required. Specifies which events should be logged. The **none** parameter specifies that no events related to IPX should be logged. The **error** parameter specifies that only errors related to IPX should be logged. The **warn** parameter specifies that only warnings related to IPX should be logged. The **info** parameter specifies that all events related to IPX should be logged.

⇡Top of page

Netsh Routing IPX NetBIOS commands

The following commands are available for use in the Netsh Routing IPX NetBIOS context.

To view the command syntax, click a command:

show interface

Displays propagation of IPX NetBIOS broadcasts and traffic on router interfaces.

Syntax

show interface [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

set interface

Configures propagation of IPX NetBIOS broadcasts and traffic on router interfaces.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [[**bcastaccept=**]{**enabled** | **disabled**}] [[**bcastdeliver=**]{**enabled** | **disabled** | **staticonly** | **onlywhenup**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface on which you want to configure propagation. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[bcastaccept=]{enabled | disabled} : Specifies whether the router interface should accept NetBIOS broadcasts and traffic.

[bcastdeliver=]{enabled | disabled | staticonly | onlywhenup} : Specifies whether and how the router interface should deliver NetBIOS broadcasts and traffic. When **bcastdeliver** is enabled, all broadcasts and traffic are forwarded. When **bcastdeliver** is disabled, no broadcasts or traffic are forwarded. The **onlywhenup** parameter specifies that broadcasts and traffic are forwarded only when the interface is up. The **staticonly** parameter specifies that broadcasts and traffic are forwarded only for static routes.

show nbname

Displays static NetBIOS names on the specified router interface.

Syntax

show nbname [**InterfaceName=**]*InterfaceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

add nbname

Adds a static NetBIOS name to the specified router interface.

Syntax

add nbname [**InterfaceName=**]*InterfaceName NetBIOSName* [[**nbtype=**]*Type*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to which you want to add a name. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***NetBIOSName* :** Required. Specifies the NetBIOS name you want to add.

[nbtype=]*Type* : Specifies a 1-byte NetBIOS name type no longer than 2 hexadecimal digits. Leading 0s are optional.

delete nbname

Deletes a static NetBIOS name from the specified router interface.

Syntax

delete nbname [**InterfaceName=**]*InterfaceName NetBIOSName* [**nbtype=**]*Type*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from which you want to delete a name. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***NetBIOSName* :** Required. Specifies the NetBIOS name you want to delete.

[nbtype=]*Type* : Specifies a 1-byte NetBIOS name type no longer than 2 hexadecimal digits. Leading 0s are optional.

Top of page

Netsh Routing IPX RIP commands

The following commands are available for use in the Netsh Routing IPX RIP context.

To view the command syntax, click a command:

show interface

Displays IPX RIP status of the specified interface. Used without parameters, **show interface** displays information for all interfaces.

Syntax

show interface [[**InterfaceName=**]*InterfaceName*]

Parameters

[InterfaceName=]*InterfaceName* : Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

set interface

Configures IPX RIP interfaces.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [[**admstate=**]{**enabled** | **disabled**}] [[**advertise=**]{**enabled** | **disabled**}] [[**listen=**]{**enabled** | **disabled**}] [[**gnsreply=**]{**enabled** | **disabled**}] [[**updatemode=**]{**standard** | **none** | **autostatic**}] [[**interval=**]*Integer*] [[**agemultiplier=**]*Integer*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface you want to configure. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[admstate=]{enabled | disabled} : Specifies whether admstate is enabled or disabled.

[advertise=]{enabled | disabled} : Specifies whether advertising is enabled or disabled.

[listen=]{enabled | disabled} : Specifies whether listening is enabled or disabled.

[gnsreply=]{enabled | disabled} : Specifies whether gnsreply is enabled or disabled.

[updatemode=]{standard | none | autostatic} : Specifies when and how updates occur.

[interval=]*Integer* : Specifies, in seconds, the update interval.

[agemultiplier=]*Integer* : Specifies the service aging multiplier.

show global

Displays global IPX RIP settings.

Syntax

show global

Parameters

none

set global

Configures IPX RIP global settings.

Syntax

set global [**LogLevel**=]{**none** | **error** | **warn** | **info**}

Parameters

[LogLevel=]{none | error | warn | info} : Required. Specifies which events should be logged. The **none** parameter specifies that no IPX RIP-related events should be logged. The **error** parameter specifies that only errors related to IPX RIP should be logged. The **warn** parameter specifies that only warnings related to IPX RIP should be logged. The **info** parameter specifies that all events related to IPX RIP should be logged.

set filter

Configures IPX RIP on the specified interface.

Syntax

set filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [**action=**]{**permit** | **deny**}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface whose configuration you want to update. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[mode=]{input | output} : Required. Specifies the mode, input or output.

[action=]{permit | deny} : Required. Specifies the action, permit or deny.

show filter

Displays IPX RIP configuration information.

Syntax

show filter [**InterfaceName=**]*InterfaceName* [[**mode=**]{**input** | **output**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Specifies the mode, input or output.

add filter

Adds a routing filter entry to the IPX RIP configuration of the specified interface.

Syntax

add filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [**net=**]*NetworkAddress* [**mask=**]*SubnetMask*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to whose IPX RIP configuration you want to add entries. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Required. Specifies the mode, input or output.

[net=]*NetworkAddress* : Required. Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[mask=]*SubnetMask* : Required. Specifies a 4-byte network mask no longer than 8 hexadecimal digits. Leading 0s are optional.

delete filter

Deletes a routing filter entry from the IPX RIP configuration of the specified interface.

Syntax

delete filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [**net=**]*NetworkAddress* [**mask=**]*SubnetMask*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from whose IPX RIP configuration you want to delete an entry. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Required. Specifies the mode, input or output.

[net=]*NetworkAddress* : Required. Specifies a 4-byte network address no longer than 8 hexadecimal digits. Leading 0s are optional.

[mask=]*SubnetMask* : Required. Specifies a 4-byte network mask no longer than 8 hexadecimal digits. Leading 0s are optional.

⇡Top of page

Netsh Routing IPX SAP commands

The following commands are available for use in the Netsh Routing IPX SAP context.

To view the command syntax, click a command:

show interface

Displays IPX SAP status for the specified interface. Used without parameters, **show interface** displays information for all interfaces.

Syntax

show interface [[**InterfaceName=**]*InterfaceName*]

Parameters

[InterfaceName=]*InterfaceName* : Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

set interface

Configures IPX SAP interfaces.

Syntax

set interface [**InterfaceName=**]*InterfaceName* [[**admstate=**]{**enabled** | **disabled**}] [[**advertise=**]{**enabled** | **disabled**}] [[**listen=**]{**enabled** | **disabled**}] [[**gnsreply=**]{**enabled** | **disabled**}] [[**updatemode=**]{**standard** | **none** | **autostatic**}] [[**interval=**]*Integer*] [[**agemultiplier=**]*Integer*]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to configure IPX SAP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[admstate=]{enabled | disabled} : Specifies whether admstate is enabled or disabled.

[advertise=]{enabled | disabled} : Specifies whether advertising is enabled or disabled.

[listen=]{enabled | disabled} : Specifies whether listening is enabled or disabled.

[gnsreply=]{enabled | disabled} : Specifies whether gnsreply is enabled or disabled.

[updatemode=]{standard | none | autostatic} : Specifies when and how updates occur.

[interval=]*Integer* : Specifies the update interval in seconds.

[agemultiplier=]*Integer* : Specifies the service aging multiplier.

show global

Displays global settings for IPX SAP.

Syntax

show global

Parameters

none

set global

Configures global settings for IPX SAP.

Syntax

set global [**LogLevel=**]{**none** | **error** | **warn** | **info**}

Parameters

[LogLevel=]{none | error | warn | info} : Required. Specifies which events should be logged. The **none** parameter specifies that no IPX SAP-related events should be logged. The **error** parameter specifies that only errors related to IPX SAP should be logged. The **warn** parameter specifies that only warnings related to IPX SAP should be logged. The **info** parameter specifies that all events related to IPX SAP should be logged.

set filter

Updates the IPX SAP configuration.

Syntax

set filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [**action=**]{**permit** | **deny**}

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface whose configuration you want to update. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[mode=]{input | output} : Required. Specifies the mode, input or output.

[action=]{permit | deny} : Required. Specifies the action, permit or deny.

show filter

Displays IPX SAP configuration information for the specified interface.

Syntax

show filter [**InterfaceName=**]*InterfaceName* [[**mode=**]{**input** | **output**}]

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Specifies the mode, input or output.

add filter

Adds entries to the IPX SAP configuration of the specified interface.

Syntax

add filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [**svtype=**]*ServiceType* [**svname=**]*ServiceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface to whose configuration you want to add entries. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

[mode=]{input | output} : Required. Specifies the mode, input or output.

[svtype=]*ServiceType* : Required. Specifies a 2-byte service type no longer than 4 hexadecimal digits. Leading 0s are optional.

[svname=]*ServiceName* : Required. Specifies a service name.

delete filter

Deletes filter configuration entries from the IPX SAP configuration of the specified interface.

Syntax

delete filter [**InterfaceName=**]*InterfaceName* [**mode=**]{**input** | **output**} [**svtype=**]*ServiceType* [**svname=**]*ServiceName*

Parameters

[InterfaceName=]*InterfaceName* : Required. Specifies, by name, the interface from whose configuration you want to delete entries. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

[mode=]{input | output} : Required. Specifies the mode, input or output.

[svtype=]*ServiceType* : Required. Specifies a 2-byte service type no longer than 4 hexadecimal digits. Leading 0s are optional.

[svname=]*ServiceName* : Required. Specifies a service name.

Top of page

Netsh Routing example

To enable router discovery for the interface named Local Area Connection, specifying a minimum interval of 7 minutes, a maximum interval of 10 minutes, a life of 30 minutes for router discovery packets, and a zero (0) or no preference level, type:

add interface "Local Area Connection" enable 7 10 30 0

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Netsh commands for WINS

The Netsh commands for WINS offer a command-line tool for administering WINS servers that can be used as an equivalent alternative to console-based management. This is useful in the following situations:

- When managing WINS servers in wide area networks (WANs), commands can be used in interactive mode at the Netsh command prompt to better manage slow-speed network links.
- When managing a large number of WINS servers, commands can be used in batch mode to script and automate recurring administrative tasks that need to be performed for all WINS servers.

The following commands are available at the Netsh command prompt (as distinguished from the Windows XP command prompt). For more information about how to use the Netsh commands for WINS, see Use WINS command-line tools

Note

Top of page

Netsh WINS

The following command are available at the **wins>** prompt, which is rooted within the netsh environment.

To view the command syntax, click a command:

server

Switches to the specified server context. Used without parameters, the local WINS server is assumed as the default.

Syntax

server [{\\ServerName | *IPAddress*}]

Parameters

{\\*ServerName* | *IPAddress*} : Specifies the remote (non-local) WINS server to which to switch the Netsh WINS command-line context. Identifies the server by server name or IP address.

Examples

In the first example, this command changes the Netsh context to the local WINS server.

In the second example, this command changes the Netsh context to a remote server named \\WINS-SRV1.

In the third example, this command changes the Netsh context to a remote server with IP address 10.0.0.1.

server

server \\WINS-SRV1

server 10.0.0.1

Top of page

Netsh WINS server

The following commands are available at the **wins server>** prompt, which is rooted within the netsh environment.

To view the command syntax, click a command:

add

add filter

Adds a user defined filter to the specified WINS server.

Syntax

add filter [**ID=**]*ID* [**Desc=**]*FilterDescription*

Parameters

[ID=]*ID* : Required. Specifies the ID of the filter that you want to add, in hexadecimal.

[Desc=]*FilterDescription* : Required. Specifies a description of the filter that you want to add.

add name

Adds a name record to the database on the specified WINS server.

Syntax

add name [**Name=**]*ComputerName* [[**EndChar=**]*16thCharInHex*] [[**Scope=**]*ScopeName*] [[**RecType=**]{**0** | **1**}] [[**Group=**]{**0** | **1** | **2** | **3** | **4**}] [[**Node=**]{**0** | **1** | **3**}] [**IP=**]**{***IPAddress1*[, *IPAddress2*,*IPAddress3*]**}**

Parameters

[Name=]*ComputerName* : Required. Specifies the name to add and register in the WINS server database.

[EndChar=]*16thCharInHex* : Indicates the 16th character of the name in hexadecimal. If unspecified, record types are added based on the value of **Group=**. For more information, see Remarks.

[Scope=]*ScopeName* : Specifies the name of the NetBIOS scope.

[RecType=]{0 | 1} : Indicates the record type: **0**-Static (default), **1**-Dynamic.

[Group=]{0 | 1 | 2 | 3 | 4} : Indicates the group type: **0**-Unique (default), **1**-Group, **2**-Internet **3**-Multihomed, **4**-Domain Name.

[Node=]{0 | 1 | 3} : Indicates the NetBIOS node type: **0**-B Node, **1**-P Node (default), **3**-H Node.

[IP=]{*IPAddress1*[,*IPAddress2*,*IPAddress3*]} : Required. Specifies the IP address of the name record to add. If the computer is multihomed, more than one IP address can be specified. Use braces ({}) to enclose one or more IP addresses, and separate multiple IP addresses with commas.

Remarks

•If **EndChar=** and **Group=** are both used, **EndChar=** overrides **Group=**.

- When **EndChar=** is not specified, the following record types are added to the WINS database based on the value of **Group=**:

Group parameter value	Record type added
Unique group (0)	03h, 20h, 00h
Normal group (1)	1eh
Special group/Internet (2)	20h
Multihomed group (3)	03h, 20h, 00h
Special group/domain (4)	1ch

Examples

In the first example, this command adds a default NetBIOS record set of the 00h, 03h, and 20h records for a computer named HOSTA with a mapped IP address of 10.0.0.1.

In the second example, this command adds a 20h record for HOSTB, a multihomed computer with IP addresses 10.0.0.2 and 10.0.0.3, on SUBNETA, with a static record type and P-Node configuration.

add name Name=HOSTA IP={10.0.0.1}

add name HOSTB 20 SUBNETA 0 3 1{10.0.0.2,10.0.0.3}

add partner

Adds a replication partner to the list of replication partners on the specified WINS server.

Syntax

add partner [**Server=**]*IPAddress* [[**NetBIOS=**]*ServerNetBIOSName*] [[**Type=**]{**0** | **1** | **2**}]

Parameters

[Server=]*IPAddress* : Required. Specifies the replication partner to add to the WINS server. Identifies the partner server by IP address.

[NetBIOS=]*ServerNetBIOSName* : Specifies the replication partner to add to the WINS server.

Identifies the partner server by NetBIOS name.

[Type=]{0 | 1 | 2} : Indicates the type of partner to add: **0**-Pull, **1**-Push, **2**-Both (default).

Remarks

- If the IP address does not resolve, the NetBIOS name parameter is used to create the partner record. Otherwise, the NetBIOS parameter is not used.

Examples

In the following example, this command adds a push/pull replication partner with an IP address of 10.2.2.2 to the current WINS server.

add partner 10.2.2.2

add pngserver

Adds a list of Persona Non Grata servers to the specified WINS server.

Syntax

add pngserver [**Servers=**]**{***ListOfServerIPAddresses***}**

Parameters

[Servers=]{*ListOfServerIPAddresses*} : Required. Specifies the Persona Non Grata server or servers to add. Identifies each server by its IP address, enclosed by braces ({}). If a you provide a list, IP addresses must be separated by commas and enclosed by braces ({}).

Remarks

- This command adds the new server or servers after checking for duplicates. The Persona Non Grata (PNG) list is used to specify owners (other WINS servers) whose records are to be blocked during replication.

Examples

In the following example, this command adds the WINS servers 10.2.2.2 and 10.2.2.3 to the Persona Non Grata list for the current WINS server.

add pngserver {10.2.2.2,10.2.2.3}

add server

Adds a WINS server to the WINS console.

Syntax

add server [**Server=**]{\\ServerName | *ServerIPAddress*}

Parameters

[Server=]{\\***ServerName* | *ServerIPAddress*}** **:** Required. Specifies the WINS server to add. Identifies the server by NetBIOS name or IP address.

Examples

In the following example, this command adds a server with IP address 172.17.32.28 to the WINS console.

add server Server=172.17.32.28

check

check database

Checks the consistency of the WINS database. Used without parameters, a consistency check is performed on all replicas whose verification interval has expired. The consistency check is not be performed immediately if overload conditions exist, but occurs after the configured verification interval period has passed.

Syntax

check database [[**All=**]{**1** | **0**}] [[**Force=**]{**1** | **0**}]

Parameters

[All=]{1 | 0} : Indicates whether to perform a consistency check on all replicas or to perform a consistency check on the replicas whose verification interval has expired: **1**-All, **0**-Otherwise (default).

[Force=]{1 | 0} : Indicates whether to override the WINS consistency check under overload conditions: **1**-Override, **0**-Otherwise (default).

Remarks

- It is a network and resource intensive operation to check all replicas for consistency.
- Overload condition occurs when the consistency check command is repeated within a duration of one hour.

Examples

In the following example, this command does a consistency check for all replicas. If overload conditions exist, the consistency check is performed after the configured verification interval has passed.

check database 1

check name

Checks a list of name records against a specified set of WINS servers.

Syntax

check name {**NameFile=***FileName* | **NameList={***NamesList***}**} {**ServerFile=***FileName* | **ServerList={***ListOfServerIPAddresses***}**}[**InclPartner=**{**Y** | **y**}]

Parameters

{NameFile=*FileName* | NameList={*NamesList*}} : Required. Specifies the file that contains the list of names to check or specifies a list of names to check. If you provide a list, names must be separated by commas and enclosed by braces({}).

{ServerFile=*FileName* | ServerList={*ListOfServerIPAddresses*}} : Required. Specifies the file with the list of server IP addresses or specifies a list of the server IP addresses to check. If you provide a list, IP addresses must be separated by commas and enclosed by braces ({}).

InclPartner={Y | y} : Specifies that all partners will be included in the list of servers. Partners are not included by default.

Remarks

- Tags are required for this command. Whenever inconsistencies are found, repair is attempted.
- The format for *Name* in both *NameFile* and *NameList* is *Name*16thchar*, where *Name* is the record name and 16thchar is the 16th character for the record.
- In the file, name entries are separated by commas or carriage returns/line feeds.

Examples

In the first example, this command checks WINS data against the Names.txt and Servers.txt files, assuming those files to be located in the specified path location (the root of drive D).

In the second example, this command checks the names Name1 with 16th character 03, 00 and 20h respectively against the servers with IP addresses 10.2.2.2 and 10.2.2.3.

check name NameFile=D:\Names.txt ServerFile=D:\Servers.txt

check name NameList={Name1*03,Name1*00,Name1*20} ServerList={10.2.2.2,10.2.2.3}

check version

Checks the consistency of version ID numbers for WINS record owners in the WINS database.

Syntax

check version [**Server=**]*IPAddress* [[**File=**]*FileName*]

Parameters

[Server=]*IPAddress* : Required. Specifies the IP address of the start WINS server.

[File=]*FileName* : Specifies the file in which to store the index to IP address table. If omitted, **check version** creates the file Wins.rec in the current directory and stores the index to IP address table there.

Remarks

- If you provide an invalid file name, *FileName* is set to Wins.rec in the current directory.
- You might need to increase the width of the command window to make it wide enough to view the table correctly.

Examples

In the following example, this command verifies the version number consistency of the records for a WINS server with the IP address 10.2.2.2 at its replication partners.

check version 10.2.2.2

delete

delete name

Deletes a registered name from the WINS server database.

Syntax

delete name [**Name=**]*RegisteredName* [**EndChar=**]*16thCharInHex* [[**Scope=**]*ScopeName*]

Parameters

[Name=]*RegisteredName* : Required. Specifies the registered name to be deleted.

[EndChar=]*16thCharInHex* : Required. Indicates the 16th character of the name of the record to be deleted in hexadecimal.

[Scope=]*ScopeName* : Specifies the NetBIOS scope identifier for the record to be deleted.

Examples

In the following example, this command deletes the messenger (03h) records with the record name HOSTA from the WINS database.

delete name Name=HOSTA EndChar=03

delete owners

Deletes or tombstones a list of owners and their records from the database on the specified WINS server.

Syntax

delete owners [**Servers=**]**{***ListOfServerIPAddresses***}** [[**Op=**]{**0** | **1**}]

Parameters

[Servers=]{*ListOfServerIPAddresses*} : Required. Specifies a list of the owner servers with records to delete. Identifies the servers by IP addresses that must be separated by commas and enclosed by braces ({}).

[Op=]{0 | 1} : Indicates your option: **0**-Tombstone (default), **1**-Delete.

Examples

In the following example, this command deletes all records owned by the WINS server with the IP address 10.0.0.1.

delete owners Servers={10.0.0.1} Op=1

delete partner

Deletes either all replication partners or a specified replication partner from the list of replication partners on the specified WINS server. When used without parameters, all replication partners are deleted from both the pull and the push partner lists without a confirmation prompt.

Syntax

delete partner [[**Server=**]*IPAddress*] [[**Type=**]{**0** | **1** | **2**}] [[**Confirm=**]{**Y**|**y**}]

Parameters

[Server=]*IPAddress* : Specifies the partner to delete. Identifies the partner by IP address.

[Type=]{0 | 1 | 2} : Indicates the type of partner list to delete: **0**-Pull, **1**-Push, **2**-Both (default).

[Confirm=]{Y|y} : Initiates and completes the operation without prompting for confirmation.

Examples

In the following example, this command deletes a push/pull type partner with an IP address of 10.0.0.1 from the current WINS server without prompting for confirmation.

delete partner Server=10.0.0.1 Type=2 Confirm=Y

delete pngserver

Deletes all Persona Non Grata servers or specific Persona Non Grata servers from the list of Persona Non Grata servers on the specified WINS server.

Syntax

delete pngserver [**Servers=**]**{***ListOfServerIPAddresses***}**

Parameters

[Servers=]{*ListOfServerIPAddresses***} :** Required. Specifies a list of Persona Non Grata servers to delete. Identifies Persona Non Grata servers by IP addresses separated by commas and enclosed by braces ({}).

Remarks

- The Persona Non Grata (PNG) list specifies owners (other WINS servers) whose records are to be blocked during replication. If you use an empty set of braces ({}), **delete pngserver** deletes all of the servers on the PNG server list.

Examples

In the following example, this command deletes the WINS servers 10.2.2.2 and 10.2.2.3 from the Persona Non Grata list for this WINS server.

delete pngserver Servers={10.2.2.2,10.2.2.3}

delete records

Deletes all records or a set of records from the current WINS server or marks records as tombstoned.

Syntax

delete records [**MinVer=**]**{***High***,***Low***}** [**MaxVer=**]**{***High***,***Low***}** [[**Op=**]{**0** | **1**}]

Parameters

[MinVer=]{*High*,*Low*} : Required. Specifies the minimum high and the minimum low record versions from which to begin deletion. High and low values are separated by a comma and enclosed by braces ({}).

[MaxVer=]{*High*,*Low*} : Required. Specifies the maximum high and maximum low record versions at which to end deletion. High and low values are separated by a comma and enclosed by braces ({}).

[Op=]{0 | 1} : Indicates your option: **0**-Delete (default), **1**-Tombstone.

Remarks

•If you set both **MinVer=** and **MaxVer=** to {0,0}, **delete records** deletes all records.

Examples

In the first example, this command marks all records on the WINS server as tombstoned.

In the second example, this command deletes all records between and including 2de2d and 2de3f.

delete records MinVer={0,0} MaxVer={0,0} Op=1

delete records MinVer={0,2de2d} MaxVer={0,2de3f}

delete server

Deletes a WINS server from the WINS console.

Syntax

delete server [**Server=**]{\\ServerName | *ServerIPAddress*}

Parameters

[Server=]{\\***ServerName* | *ServerIPAddress*****} :** Required. Specifies the WINS server to delete. Identifies the server by NetBIOS name or IP address.

Examples

In the following example, this command deletes the server with IP address 172.31.255.254 from the WINS console.

delete server Server=172.31.255.254

init

init backup

Initiates backup of the WINS database to a specified directory. Used without parameters, this command provides a full backup to the default backup path that is currently set.

Syntax

init backup [[**Dir=**]*BackupDir*] [[**Type=**]{**0** | **1**}]

Parameters

[Dir=]*BackupDir* : Specifies the directory in which backups are stored. If unspecified, the default backup path that is currently set is used.

[Type=]{0 | 1} : Indicates the type of backup: **0**-Full backup (default), **1**-Incremental backup.

Remarks

- WINS backup can only be used for a local backup on the same server. Backup of remote WINS servers is not supported.
- Backup files will automatically be created under the wins_bak subdirectory of the **Dir=** directory.

Examples

In the following example, this command initiates a full backup of the WINS database on the local WINS server to the folder C:\WINSfiles\wins_bak. WINS will create the wins_bak subdirectory.

init backup Dir=C:\WINSfiles Type=0

init compact

Initiates the compaction of the WINS server database.

Syntax

init compact

Parameters

none

init export

Initiates the export of the WINS server database to a file.

Syntax

init export

Parameters

none

init import

Initiates the import of static mappings from an Lmhosts file.

Syntax

init import [**File=**]*ImportFile*

Parameters

[File=]*ImportFile* : Required. Specifies the Lmhosts-formatted text file used for importing static mappings into the WINS database.

Examples

In the following example, this command imports the static mappings of the Fixedip.sam file, located at the C:\WINSfiles path, as new static mappings in the local WINS database.

init import File=C:\WINSfiles\Fixedip.sam

init pull

Initiates and sends a pull trigger to another WINS server.

Syntax

init pull [**Server=**]{*PullServerName* | *IPAddress*}

Parameters

[Server=]{\\*PullServerName* | *IPAddress*} : Required. Specifies the WINS server to which to send a pull trigger. Identifies the server by server name or IP address.

Remarks

•If you provide a server name that cannot be resolved to an IP address, the operation fails.

Examples

In the following example, this command initiates and sends a pull trigger to WINS server 10.2.2.2.

init pull Server=10.2.2.2

init pullrange

Initiates and pulls a range of records from another WINS server.

Syntax

init pullrange [**Owner=**]*OwnerServerIP* [**Server=**]*PullServerIP* [**MaxVer=**]**{***High***,***Low***}** [**MinVer=**]**{***High***,***Low***}**

Parameters

[Owner=]*OwnerServerIP* : Required. Specifies the WINS server with records to be pulled. Identifies the server by IP address.

[Server=]*PullServerIP* : Required. Specifies the WINS server that pulls the range of records. Identifies the server by IP address.

[MaxVer=]{*High*,*Low*} : Required. Specifies the maximum high and the maximum low version number of the records to be pulled. High and low values are separated by a comma and enclosed by braces ({}).

[MinVer=]{*High*,*Low*} : Required. Specifies the minimum high and the minimum low version number of the records to be pulled. High and low values are separated by a comma and enclosed by braces ({}).

Remarks

•If you set both **MaxVer=** and **MinVer=** to {0,0}, **init pullrange** pulls the entire range of records.

Examples

In the following example, this command initiates and sends a pull trigger to WINS server 10.2.2.3 for all records owned by WINS server 10.2.2.2.

init pullrange Owner=10.2.2.2 Server=10.2.2.3 MaxVer={0,0} MinVer={0,0}

init push

Initiates and sends a push trigger to another WINS server.

Syntax

init push [**Server=**]{\\PushServerName | *IPAddress*} [[**PropReq=**]{**0** | **1**}]

Parameters

[Server=]{\\***PushServerName* | *IPAddress*}** :** Required. Specifies the WINS server to which to send a push trigger. Identifies the server by name or IP address.

[PropReq=]{0 | 1} : Indicates your propagation request: **0**-Push to the specified server only (default), **1**-Have server propagate the push trigger to all of its partners.

Remarks

•If you provide a server name that cannot be resolved to an IP address, the operation fails.

Examples

In the following example, this command initiates and sends a push trigger to WINS server 10.2.2.2 and has it propagate the trigger to all of its replication partners.

init push Server=10.2.2.2 PropReq=1

init replicate

Initiates and forces an immediate replication of the database with replication partners.

Syntax

init replicate

Parameters

none

Remarks

- This command initiates push/pull replication with all replication partners for this server.

init restore

Initiates the restoration of the WINS database from a directory and file at the specified WINS server.

Syntax

init restore [**Dir=**]*RestoreDir*

Parameters

[Dir=]*RestoreDir* : Required. Specifies the directory from which to restore files.

Remarks

- WINS backups can only be restored locally on the same server. You cannot restore the WINS database from a remote computer.

- **Dir=** should contain a subdirectory wins_bak, which will contain the database file, but this subdirectory should not be included in the **Dir=** parameter.

Examples

In the following example, this command restores the WINS database from the backup database file location of C:\WINSfiles.

init restore Dir=C:\WINSfiles

init scavenge

Initiates the scavenging of the WINS database for the specified WINS server.

Syntax

init scavenge

Parameters

none

Remarks

- This command removes outdated information from the WINS database.

init search

Initiates a search for the specified record name in the WINS database.

Syntax

init search [**Name=**]*RecordName* [[**EndChar=**]*EndCharacter*] [[**Case=**]{**0** | **1**}] [[**File=**]*FileName*]

Parameters

[Name=]*RecordName* : Required. Specifies the name of the record for which to search.

[EndChar=]*EndCharacter* : Indicates the 16th character of the NetBIOS name record. If unspecified, the search returns all records that match the string that you entered as a value for **Name=**. For more information, see Remarks.

[Case=]{0 | 1} : Indicates name case preference: **0**-Ignore name case (default), **1**-Use name case.

[File=]*FileName* : Specifies the file in which to store ASCII plain text output. If unspecified, output is displayed in the command prompt window.

Remarks

•If you do not supply a value for **EndChar=**, the search returns all records that match the string that is entered as a value for **Name=**. For example, if the database contains a record named MYPC-1, a search that uses MYPC as the value of the name parameter returns all records starting with this name pattern, regardless of record type:

Record Name	Record Type
MYPC-1	[00h]Workstation
MYPC-1	[01h]Other
MYPC-1	[03h]Messenger

If you specify a value for **EndChar=**, only records that match the record name and record type are returned. For example, if the value for **Name=** is MYPC and the value for EndChar is 00, the following search results return:

Record Name	Record Type
MYPC-1	[00h]Workstation

Examples

In the following example, this command searches for records named dc1 with an end character of 1c, ignores name case, and outputs the information to the names.txt file located in C:\WINSfiles.

init search Name=dc1 EndChar=1c Case=0 File=C:\WINSfiles \names.txt

reset

reset counter

Resets the statistics for the local WINS server.

Syntax

reset counter

Parameters

none

set

set autopartnerconfig

Sets the automatic replication partner configuration information for the server. Used without parameters, automatic partner configuration is reset (disabled).

Syntax

set autopartnerconfig [[**State=**]{**0** | **1**}] [[**Interval=**]*Value*] [[**TTL=**]*Value*]

Parameters

[State=]{0 | 1} : Indicates the automatic replication partner configuration information setting: **0**-Reset (default), **1**-Set.

[Interval=]*Value* : Specifies the multicast interval value, in seconds.

[TTL=]*Value* : Specifies the Time-To-Live value for multicast datagrams between partner WINS servers. The TTL value must be within the range of 1 to 32.

Examples

In the following example, this command sets the automatic configuration information for this WINS server for a multicast interval of 3,600 seconds and a Time to Live (TTL) value of 20.

set autopartnerconfig State=0 Interval=3600 TTL=20

set backuppath

Sets the backup parameters for the specified WINS server.

Syntax

set backuppath [**Dir=**]*NewPath* [[**Shutdown=**]{**0** | **1**}]

Parameters

[Dir=]*NewPath* : Required. Specifies the backup path for the server database. If unspecified, the default path is C:\.

[Shutdown=]{0 | 1} : Indicates the setting for backup during shutdown: **0**-Disable backup of the WINS database during shutdown (default), **1**-Enable backup of the WINS database during shutdown.

Examples

In the following example, this command sets the WINS server to automatically back up the WINS database to C:\WINSfiles each time that the server is shut down.

set backuppath Dir=C:\WINSfiles Shutdown=1

set burstparam

Sets burst handling parameters for the specified WINS server.

Syntax

set burstparam [**State=**]{**0** | **1**} [[**Value=**]*QueueSize*]

Parameters

[State=]{0 | 1} : Required. Indicates the setting for burst handling: **0**-Reset or disable burst handling for the WINS server, **1**-Set or enable burst handling for the WINS server.

[Value=]*QueueSize* : Specifies the value of the burst queue size. The queue size must be within the range of 50 to 5,000.

Examples

In the following example, this command sets the size of the burst queue to permit 1,000 client registration and renewal requests before the WINS server uses burst handling.

set burstparam State=1 Value=1000

set defaultparam

Sets the default parameters for WINS server configuration.

Syntax

set defaultparam

Parameters

none

Remarks

- This command sets all the configuration parameters for the WINS server to their default values. It is recommended that you run this command after WINS service installation to configure the server with default parameter settings.

set logparam

Sets or resets database and detailed event logging parameters. Used without parameters, the current values are maintained.

Syntax

set logparam [[**DBChange=**]{**0** | **1**}] [[**Event=**]{**0** | **1**}]

Parameters

[DBChange=]{0 | 1} : Indicates whether you want to log database changes: **1**-Enable logging of database changes to JET log files (default), **0**-Reset this option.

[Event=]{0 | 1} : Indicates the detail of events: **1**-Log detailed events to the system event log , **0**-Reset this option (default).

Remarks

- The WINS database engine can log the database changes to separate files. Because this option improves the performance of database operations, it is turned on by default.
- The WINS server can log detailed event information in the system event log. Because this option is resource-intensive, it is turned off by default.

Examples

In the following example, this command sets logging parameters for this WINS server to enable the logging of database changes to JET files and to log detailed events to the system event log.

set logparam Event=1

set migrateflag

Sets the migration flag for the specified WINS server.

Syntax

set migrateflag [**State=**]{**0** | **1**}

Parameters

[State=]{0 | 1} : Required. Indicates the setting of the migrate flag: **0**-Disable (migrate off), **1**-Enable (migrate on)

Remarks

- The migrate flag determines if unique static mappings can be overwritten by dynamic records.

Examples

In the following example, this command sets the migrate flag to permit the overwriting of unique static mappings by dynamic records for the specified WINS server.

set migrateflag State=1

set namerecord

Sets the intervals that are used to determine when WINS name records are renewed, deleted, and verified at the specified WINS server. Used without parameters, current values are maintained.

Syntax

set namerecord [[**Renew=**]*Value*] [[**Extinction=**]*Value*] [[**ExTimeout=**]*Value*] [[**Verification=**]*Value*]

Parameters

[Renew=]*Value* : Specifies the interval renewal rate, in seconds.

[Extinction=]*Value* : Specifies the extinction interval, in seconds.

[ExTimeout=]*Value* : Specifies the extinction time-out, in seconds.

[Verification=]*Value* : Specifies the verification interval, in seconds.

Remarks

- All parameters are optional. For parameters not used, previously set or default values are maintained.

Examples

In the following example, this command sets each of the values for name records on this WINS server.

set namerecord Renew=3600 Extinction=7200 ExTimeout=86400 Verification=2073600

set periodicdbchecking

Sets the periodic database consistency checking parameters for the specified WINS server. Used without parameters, current values are maintained.

Syntax

set periodicdbchecking [[**State=**]{**0** | **1**}] [[**MaxRec=**]*Value*] [[**CheckAgainst=**]{**0** | **1**}] [[**CheckEvery=**]*Value*] [[**Start=**]*Value*]

Parameters

[State=]{0 | 1} : Required. Indicates whether to check the state of the WINS database: **0**-Disable, **1**-Enable(default). If you set the value of this parameter to 0, all checks are disabled regardless of the values of other parameters.

[MaxRec=]*Value* : Specifies the maximum number of records that are checked each period. If unspecified, the default is 30,000.

[CheckAgainst=]{0 | 1} : Indicates the server to check against: **0**-Owner server (default), **1**-Randomly selected server.

[CheckEvery=]*Value* : Specifies how often the WINS database should be checked for consistency, in hours. If unspecified, the default is 24 hours.

[Start=]*Value* : Specifies the time, in seconds, at which to start the consistency check.

Remarks

- In order to set periodic consistency check intervals, it is recommended that you specify values for the **CheckEvery** and **Start** parameters. In order to adjust existing periodic consistency check settings, specify values only for **State** and the parameter or parameters that you want to change. Parameters for which you do not supply a value remain unchanged.

Examples

In the following example, this command checks the WINS database against a random replication partner every hour starting at 3,600 seconds.

set periodicdbchecking State=1 MaxRec=1000 CheckAgainst=1 CheckEvery=1 Start=3600

set pullparam

Sets the default pull partner parameters for the specified WINS server. Used without parameters, previously set or default values are maintained.

Syntax

set pullparam [[**State=**]{**0** | **1**}] [[**Strtup=**]{**0** | **1**}] [[**Start=**]*Value*] [[**Interval=**]*Value*] [[**Retry=**]*Value*]

Parameters

[State=]{0 | 1} : Required. Indicates whether you want persistent connections to be used with pull partners: **0-**Do not use persistent connections with pull partners, **1-**Use persistent connections with pull partners .

[Strtup=]{0 | 1} : Indicates whether pull replication should start at startup: **0**-Do not start pull replication at startup (default), **1-**Start pull replication at startup.

[Start=]*Value* : Specifies the start time for the pull trigger, in seconds.

[Interval=]*Value* : Specifies the replication interval, in seconds.

[Retry=]*Value* : Specifies the number of times to retry replication.

Remarks

•If you do not specify a value for **Interval=**, only one replication occurs.
•If you do not specify a value for **Start=**, execution of the command starts the pull trigger.

Examples

In the following example, this command sets the default pull partner parameters for this WINS server to use a persistent connection with its pull partners, begin replication at startup, start the pull trigger in 23,600 seconds at an interval of 259,200 seconds, and retry replication seven times.

set pullparam State=1 Strtup=1 Start=23600 Interval=259200 Retry=7

set pullpartnerconfig

Sets the configuration parameters for the specified pull partner.

Syntax

set pullpartnerconfig [[**State=**]{**0** | **1**}] [**Server=**]*ServerName* [[**Start=**]*Value*] [[**Interval=**] *Value*]

Parameters

[State=]{0 | 1} : Indicates the state of the persistent connection flag: **0**-Disable, **1**-Enable (default).

[Server=]*ServerName* : Required. Specifies the pull replication partner by server name or IP address.

[Start=]*Value* : Specifies the start time for replication, in seconds.

[Interval=]*Value* : Specifies the interval between pull replications, in seconds.

Remarks

- If you provide a server name for **Server=** that cannot be resolved to an IP address, the function fails.

Examples

In the following example, this command sets the pull partner configuration for the current WINS server for the partner WINS server 10.2.2.2. Pull replication starts in 3,600 seconds and runs every 4,320 seconds.

set pullpartnerconfig State=1 Server=10.2.2.2 Start=3600 Interval=4320

set pushparam

Sets the default push partner parameters for the specified WINS server.

Syntax

set pushparam [**State=**]{**0** | **1**} [[**Strtup=**]{**0** | **1**}] [[**AddChange=**]{**0** | **1**}] [[**Update=**]*Value*]

Parameters

[State=]{0 | 1} : Required. Indicates whether you want persistent connections to be used with push partners: **0**-Do not use persistent connections with push partners, **1**-Use persistent connections with push partners.

[Strtup=]{0 | 1} : Indicates whether you want push replication to start at startup: **0**-Do not start push replication at startup (default), **1**-Start push replication at startup.

[AddChange=]{0 | 1} : Indicates whether you want replication to be triggered upon address change: **0**-Do not trigger replication upon address change (default), **1**-Trigger replication upon address change.

[Update=]*Value* : Specifies the number of changes in version ID that will occur before replication.

Examples

In the following example, this command sets the default push partner parameters for this WINS server to use persistent connections with its push partners, start push replication at startup and when addresses change, and push to partners after the version ID has been updated twice for records owned at this WINS server.

set pushparam State=1 Strtup=1 AddChange=1 Update=2

set pushpartnerconfig

Sets the configuration parameters for the specified push partner.

Syntax

set pushpartnerconfig [[**State=**]{**0** | **1**}] [**Server=**]\\ServerName [**Update=**]*Value*

Parameters

[State=]{0 | 1} : Indicates the state of the persistent connection flag: **0**-Disable, **1**-Enable (default).

[Server=]\\*ServerName* : Required. Specifies the push partner server for which you want to set configuration parameters. Identifies the server by server name or IP address.

[Update=]*Value* : Required. Specifies the threshold for updates.

Remarks

- If you provide a name in the server parameter cannot be resolved to an IP address, the function fails.

Examples

In the following example, this command sets the push partner configuration for the current WINS server for a partner WINS server with IP address 10.2.2.2 and sets two as the limit of changes in version ID allowed before push partners are informed about them.

set pushpartnerconfig State=1 Server=10.2.2.2 Update=2

set replicateflag

Sets the replication flag for the specified WINS server.

Syntax

set replicateflag [**State=**]{**0** | **1**}

Parameters

[State=]{0 | 1} : Required. Indicates the setting of the replication flag: **0**-Disable (replication flag off), **1**-Enable (replication flag on).

Examples

In the following example, this command enables the replication flag for this server.

set replicateflag State=1

set startversion

Sets the value of the beginning (start) version ID number for the database.

Syntax

set startversion [**Version=**]**{***High***,***Low***}**

Parameters

[Version=]{*High*,*Low*} : Required. Specifies the start version ID number for the WINS database. High and low values are separated by a comma and enclosed by braces ({}).

Examples

In the following example, this command sets the beginning version ID number for the WINS database to 0.

set startversion Version={0,0}

show

show browser

Displays all active domain master browser [1Bh] records for the specified WINS server.

Syntax

show browser

Parameters

none

show database

Displays the database and records for a partial or complete list of specified owner servers.

Syntax

show database [**Servers=**]**{***IPAddresses***}** [[**RecType=**]{**0** | **1** | **2**}] [[**Count=**]*Count*] [[**Start=**]{**0** | **1**}] [[**EndChar=**]*16thCharInHex*] [[**File=**]*FileName*]

Parameters

[Servers=]{*IPAddresses*} : Required. Specifies a list of the owner servers whose records are to be displayed. Identifies owner servers by IP addresses, which must be separated by commas and enclosed in braces ({}).

[RecType=]{0 | 1 | 2} : Indicates the type of record: **0**-Both (default), **1**-Static, **2**-Dynamic.

[Count=]*Count* : Specifies the number of records that you want to display. If unspecified, all records that match the query parameters are be returned.

[Start=]{0 | 1} : Indicates where to start sorting the display: **0**-Start from the beginning of the database (default), **1**-Start from the end of the database.

[EndChar=]*16thCharInHex* : Indicates the 16th character of the records, in hexadecimal. This number filters the records that are retrieved from the database. If unspecified, all record types are returned.

[File=]*FileName* : Specifies the file in which formatted output is stored. If unspecified, output is displayed in the command prompt window.

Remarks

- If you specify an empty set ({}) for **Servers=**, **show database** records are displayed for all owner servers.

Examples

In the following example, this command outputs the first 1000 dynamic records in the WINS database for the server 10.2.2.2 to the text file output.txt.

show database Servers={10.2.2.2} RecType=2 Count=1000 Start=0 Endchar=0 File=C:\WINSfiles\output.txt

show info

Displays configuration information for the specified WINS server.

Syntax

show info

Parameters

none

Remarks

- If set, this command displays the following WINS configuration information for the current server:

  - Backup path
  - Name record setting
  - Periodic database consistency checking
  - Burst handling
  - Logging parameters
  - Start version count

show name

Queries and displays detailed information for a specified record in the current WINS server database.

Syntax

show name [**Name=**]*RecordName* [**EndChar=**]*16thCharInHex* [[**Scope=**]*ScopeName*]

Parameters

[Name=]*RecordName* : Required. Specifies the record for which to query.

[EndChar=]*16thCharInHex* : Required for all queries except unique type records. Indicates the 16th character in hexadecimal.

[Scope=]*ScopeName*] : Specifies the NetBIOS scope, if used.

Examples

In the following example, this command shows currently stored information for name record HOSTA, of File Server type, [20h] that exists in the WINS database on the specified WINS server.

show name Name=HOSTA EndChar=20

show partner

Displays all pull partners, push partners, or pull and push partners for the specified WINS server. When used without parameters, this command displays all push partners, pull partners, and push/pull partners for the specified WINS server.

Syntax

show partner [[**Type=**]{**0** | **1** | **2** | **3**}]

Parameters

[Type=]{0 | 1 | 2 | 3} : Indicates the type of partner to display: **0**-All (default), **1**-Pull, **2**-Push, **3**-Both push and pull.

Examples

In the following example, this command displays the server name, IP address, and replication type of each replication partner for this WINS server.

show partner

show partnerproperties

Displays default partner configuration information for the specified WINS server.

Syntax

show partnerproperties

Parameters

none

Remarks

- This command displays the auto partner configuration and default push/pull parameters for the current WINS server.

show pullpartnerconfig

Displays configuration information for a pull partner.

Syntax

show pullpartnerconfig [**Server=**]{*Name* | *IPAddress*}

Parameters

[Server=]{*Name* | *IPAddress*} : Required. Specifies the pull partner for which you want to display configuration information. Identifies the partner by name or IP address.

Remarks

•If you provide a name that cannot be resolved to an IP address, the function fails.

Examples

In the following example, this command displays the pull partner configuration information for WINS server 10.2.2.2.

show pullpartnerconfig Server=10.2.2.2

show pushpartnerconfig

Displays configuration information for a push partner.

Syntax

show pushpartnerconfig [**Server=**]{*Name* | *IPAddress}*

Parameters

[Server=]{*Name* | *IPAddress}* : Required. Specifies the push partner for which you want to display configuration information. Identifies the push partner by name or IP address.

Remarks

•If you provide a name that cannot be resolved to an IP address, the function fails.

Examples

In the following example, this command displays the push partner configuration information for WINS server 10.2.2.2.

show pushpartnerconfig Server= 10.2.2.2

show recbyversion

Displays a set of records for an owner, specified as a range of version IDs, and sorts the records by version.

Syntax

show recbyversion [**Server=**]*IPAddress* [**MaxVer=**]**{***High***,***Low***}** [**MinVer=**]**{***High***,***Low***}** [[**Name=**]*RecordName*] [[**EndChar=**]*16thChar*] [**Case=**]{**0** | **1**}

Parameters

[Server=]*IPAddress* : Required. Specifies the owner server for which you want to display a set of records. Identifies the server by IP address.

[MaxVer=]{*High*,*Low*} : Required. Specifies the range of records to return by the maximum high and the maximum low version number. High and low values are separated by a comma and enclosed by braces ({}).

[MinVer=]{*High*,*Low*} : Required. Specifies the range of records to return by minimum high and the minimum low version number. High and low values are separated by a comma and enclosed by braces ({}).

[Name=]*RecordName* : Specifies the records for which to look.

[EndChar=]*16thChar* : Indicates the character for which to look, in hexadecimal.

[Case=]{0 | 1} : Indicates whether to create a case-sensitive comparison: **0**-Ignore case (default), **1**-Create a case-sensitive comparison.

Remarks

- If **Name=** is not specified, **EndChar=** and **Case=** are ignored.
- If you set both **MinVer=** and **MaxVer=** to {0,0}, **show recbyversion** displays the entire range of records.

Examples

In the following example, this command displays the total number of records (by record version) owned by WINS server 10.2.2.2.

show recbyversion Server=10.2.2.2 MaxVer={0,0} MinVer={0,0}Name=dc1 EndChar=1c Case=0

show reccount

Displays the number of records owned by a specific WINS server.

Syntax

show reccount [**Server=**]*IPAddress* [**MaxVer=**]**{***High***,***Low***}** [**MinVer=**]**{***High***,***Low***}**

Parameters

[Server=]*IPAddress* : Required. Specifies the owner server for which you want to display the number of records. Identifies the server by IP address.

[MaxVer=]{*High*,*Low*} : Required. Specifies the maximum high and the maximum low version of records to show. High and low values are separated by a comma and enclosed by braces ({}).

[MinVer=]{*High*,*Low*} : Required. Specifies the minimum high and the minimum low version of records to show. High and low values are separated by a comma and enclosed by braces ({}).

Remarks

- If you set both **MinVer=** and **MaxVer=** to {0,0}, **show reccount** counts the entire range of records.

Examples

In the following example, this command shows the total record count for WINS server 10.2.2.2.

show reccount Server=10.2.2.2 MaxVer={0,0} MinVer={0,0}

show server

Displays information for the specified WINS server.

Syntax

show server

Parameters

none

Remarks

•This command displays the name and IP address of the specified WINS server.

show serverstatistics

Displays statistics for the specified WINS server.

Syntax

show statistics

Parameters

none

show version

Displays the current maximum version counter value for the specified WINS server.

Syntax

show version

Parameters

none

show versionmap

Displays the owner ID to maximum version number mappings table for the specified WINS server.

Syntax

show versionmap

Parameters

none

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

⇡Top of page

Related Topics

- Unless otherwise specified, field tags (such as **Name=** and **EndChar=**) are optional. If tags are used, they must be applied to all of the parameters in the command. If tags are omitted, a value must be provided for each of the fields that are supported in this command.

  The term Persona Non Grata refers to the Windows 2000 WINS server features **Block records for these owners**.

  For more information about **netsh**, see Using Netsh

  For an example of how to use Netsh commands for WINS in batch files and scripts, see Netsh WINS example

You cannot run these Netsh commands locally on Windows XP Professional or Windows XP Home Edition. To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal Server. You can run these commands from the Windows 2000 command prompt or from the command prompt for the Netsh WINS context. In order for these commands to work at the Windows 2000 command prompt, you must type **netsh dhcp** before typing the commands and parameters as they appear in the syntax below. There might be functional differences between Netsh context commands on Windows 2000 and Windows XP.

Top of page

Manage Your Profile |Legal |Contact Us

Netstat

Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, **netstat** displays active TCP connections.

Syntax

netstat [**-a**] [**-e**] [**-n**] [**-o**] [**-p** *Protocol*] [**-r**] [**-s**] [*Interval*]

Top of page

Parameters

-a : Displays all active TCP connections and the TCP and UDP ports on which the computer is listening.

-e : Displays Ethernet statistics, such as the number of bytes and packets sent and received. This parameter can be combined with **-s**.

-n : Displays active TCP connections, however, addresses and port numbers are expressed numerically and no attempt is made to determine names.

-o : Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the **Processes** tab in Windows Task Manager. This parameter can be combined with **-a**, **-n**, and **-p**.

-p *Protocol* : Shows connections for the protocol specified by *Protocol*. In this case, the *Protocol* can be **tcp**, **udp**, **tcpv6**, or **udpv6**. If this parameter is used with **-s** to display statistics by protocol, *Protocol* can be **tcp**, **udp**, **icmp**, **ip**, **tcpv6**, **udpv6**, **icmpv6**, or **ipv6**.

-s : Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. If the IPv6 protocol for Windows XP is installed, statistics are shown for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The **-p** parameter can be used to specify a set of protocols.

-r : Displays the contents of the IP routing table. This is equivalent to the **route print** command.

***Interval* :** Redisplays the selected information every *Interval* seconds. Press CTRL+C to stop the redisplay. If this parameter is omitted, **netstat** prints the selected information only once.

/? : Displays help at the command prompt.

↑Top of page

Remarks

- Parameters used with this command must be prefixed with a hyphen (**-**) rather than a slash (**/**).
- **Netstat** provides statistics for the following:

  - Proto

    The name of the protocol (TCP or UDP).
  - Local Address

    The IP address of the local computer and the port number being used. The name of the local computer that corresponds to the IP address and the name of the port is shown unless the **-n** parameter is specified. If the port is not yet established, the port number is shown as an asterisk (*).
  - Foreign Address

    The IP address and port number of the remote computer to which the socket is connected. The names that corresponds to the IP address and the port are shown unless the **-n** parameter is specified. If the port is not yet established, the port number is shown as an asterisk (*).
  - (state)

    Indicates the state of a TCP connection. The possible states are as follows:

    CLOSE_WAIT

    CLOSED

    ESTABLISHED

    FIN_WAIT_1

FIN_WAIT_2

LAST_ACK

LISTEN

SYN_RECEIVED

SYN_SEND

TIMED_WAIT

For more information about the states of a TCP connection, see RFC 793.

- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To display both the Ethernet statistics and the statistics for all protocols, type the following command:

netstat -e -s

To display the statistics for only the TCP and UDP protocols, type the following command:

netstat -s -p tcp udp

To display active TCP connections and the process IDs every 5 seconds, type the following command:

nbtstat -o 5

To display active TCP connections and the process IDs using numerical form, type the following command:

nbtstat -n -o

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Nslookup

Displays information that you can use to diagnose Domain Name System (DNS) infrastructure. Before using this tool, you should be familiar with how DNS works. The Nslookup command-line tool is available only if you have installed the TCP/IP protocol.

Syntax

nslookup [**-***SubCommand* ...] [{*ComputerToFind*| [**-***Server*]}]

⇡Top of page

Parameters

-*SubCommand* ... : Specifies one or more **nslookup** subcommands as a command-line option. For a list of subcommands, see Related Topics.

***ComputerToFind* :** Looks up information for *ComputerToFind* using the current default DNS name server, if no other server is specified. To look up a computer not in the current DNS domain, append a period to the name.

-*Server* : Specifies to use this server as the DNS name server. If you omit -*Server*, the default DNS name server is used.

{help|?} : Displays a short summary of **nslookup** subcommands.

⇡Top of page

Remarks

- If *ComputerToFind* is an IP address and the query is for an A or PTR resource record type, the name of the computer is returned. If *ComputerToFind* 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 following **set** subcommands: **domain**, **srchlist**, **defname**, and **search**.
- If you type a hyphen (-) instead of *ComputerToFind*, the command prompt changes to **nslookup** interactive mode.

- The command-line length must be less than 256 characters.
- **Nslookup** has two modes: interactive and noninteractive.

  If you need to look up only a single piece of data, use noninteractive mode. For the first parameter, type the name or IP address of the computer that you want to look up. For the second parameter, type the name or IP address of a DNS name server. If you omit the second argument, **nslookup** uses the default DNS name server.

  If you need to look up more than one piece of data, you can use interactive mode. Type a hyphen (-) for the first parameter and the name or IP address of a DNS name server for the second parameter. Or, omit both parameters and **nslookup** uses the default DNS name server. Following are some tips about working in interactive mode:

  - To interrupt interactive commands at any time, press CTRL+B.
  - To exit, type **exit**.
  - To treat a built-in command as a computer name, precede it with the escape character (\).
  - An unrecognized command is interpreted as a computer name.
- If the lookup request fails, **nslookup** prints an error message. The following table lists possible error messages.

Error message	Description
Timed out	The server did not respond to a request after a certain amount of time and a certain number of retries. You can set the time-out period with the **set timeout** subcommand. You can set the number of retries with the **set retry** subcommand.
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.
Nonexistent 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**.

•For more information about the **nslookup** command and DNS, see the following resources:

- Microsoft Windows Resource Kits Web site
- Lee, T., Davies, J. 2000. *Microsoft Windows 2000 TCP/IP Protocols and Services Technical Reference*. Redmond, Washington: Microsoft Press.
- Albitz, P., Loukides, M. and C. Liu. 1998. *DNS and BIND, Third Edition*. Sebastopol, California: O'Reilly and Associates, Inc.

Top of page

Examples

Each command-line 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 time-out to 10 seconds, type:

nslookup -querytype=hinfo -timeout=10

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one

Courier font	Code or program output

Command-line reference A-Z

Nslookup subcommands

Top of page

Manage Your Profile |Legal |Contact Us

Nslookup subcommands

For more information, click a command:

- Nslookup: exit
- Nslookup: finger
- Nslookup: help
- Nslookup: ls
- Nslookup: lserver
- Nslookup: root
- Nslookup: server
- Nslookup: set
- Nslookup: set all
- Nslookup: set class
- Nslookup: set d2
- Nslookup: set debug
- Nslookup: set defname
- Nslookup: set domain
- Nslookup: set ignore
- Nslookup: set port
- Nslookup: set querytype
- Nslookup: set recurse
- Nslookup: set retry
- Nslookup: set root
- Nslookup: set search
- Nslookup: set srchlist
- Nslookup: set timeout
- Nslookup: set type
- Nslookup: set vc
- Nslookup: view

Top of page

Manage Your Profile |Legal |Contact Us

Ntbackup

Perform backup operations at a command prompt or from a batch file using the **ntbackup** command followed by various parameters.

Syntax

ntbackup backup [systemstate] "*@bks file name*" **/J** {"*job name*"} [**/P** {"*pool name*"}] [**/G** {"*guid name*"}] [**/T** { "*tape name*"}] [**/N** {"*media name*"}] [**/F** {"*file name*"}] [**/D** {"*set description*"}] [**/DS** {"*server name*"}] [**/IS** {"*server name*"}] [**/A**] [**/V:**{**yes**|**no**}] [**/R:**{**yes**|**no**}] [**/L:**{**f**|**s**|**n**}] [**/M** {*backup type*}] [**/RS:**{**yes**|**no**}] [**/HC:**{**on**|**off**}] [**/SNAP:**{**on**|**off**}]

Top of page

Parameters

systemstate : Specifies that you want to back up the System State data. When you select this option, the backup type will be forced to normal or copy

***@bks file name* :** Specifies the name of the backup selection file (.bks file) to be used for this backup operation. The at (@) character must precede the name of the backup selection file. A backup selection file contains information on the files and folders you have selected for backup. You have to create the file using the graphical user interface (GUI) version of Backup.

/J {"*job name*"} : Specifies the job name to be used in the log file The job name usually describes the files and folders you are backing up in the current backup job as well as the date and time you backed up the files.

/P {"*pool name*"} : Specifies the media pool from which you want to use media. This is usually a subpool of the Backup media pool, such as 4mm DDS. If you select this you cannot use the **/A**, **/G**, **/F**, or **/T** command-line options.

/G {"*guid name*"} : Overwrites or appends to this tape. Do not use this switch in conjunction with **/P**.

/T {"*tape name*"} : Overwrites or appends to this tape. Do not use this switch in conjunction with **/P**.

/N {"*media name*"} : Specifies the new tape name. You must not use **/A** with this switch.

/F {"*file name*"} : Logical disk path and file name. You must not use the following switches with this switch: **/P /G /T**.

/D {"*set description*"} : Specifies a label for each backup set

/DS {"*server name*"} : Backs up the directory service file for the specified Microsoft Exchange Server.

/IS {"*server name*"} : Backs up the Information Store file for the specified Microsoft Exchange Server.

/A : Performs an append operation. Either **/G** or **/T** must be used in conjunction with this switch. Do not use this switch in conjunction with **/P**.

/V:{yes|no} : Verifies the data after the backup is complete.

/R:{yes|no} : Restricts access to this tape to the owner or members of the Administrators group

/L:{f|s|n} : Specifies the type of log file: **f**=full, **s**=summary, **n**=none (no log file is created).

/M {*backup type*} : Specifies the backup type. It must be one of the following: normal, copy, differential, incremental, or daily

/RS:{yes|no} : Backs up the migrated data files located in Remote Storage. The **/RS** command-line option is not required to back up the local Removable Storage database (that contains the Remote Storage placeholder files). When you backup the %systemroot% folder, Backup automatically backs up the Removable Storage database as well.

/HC:{on|off} : Uses hardware compression, if available, on the tape drive.

/SNAP:{on|off} : Specifies whether or not the backup is a volume shadow copy

/M {*backup type*} : Specifies the backup type. It must be one of the following: normal, copy, differential, incremental, or daily

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- You cannot restore files from the command line using the **ntbackup** command.
- The following command-line options default to what you have already set using the graphical user interface (GUI) version of Backup unless they are changed by a command-line option: **/V /R /L /M /RS /HC**. For example, if hardware compression is turned on in the **Options** dialog box in Backup, it will be used if **/HC** is not specified on the command line. However, if you specify **/HC:off** at the command line, it overrides the **Option** dialog box setting and compression is not used.
- If you have Windows Media Services running on your computer, and you want to back up the files associated with these services, see "Running Backup with Windows Media Services" in the Windows Media Services online documentation. You must follow the procedures outlined in the Windows Media Services online documentation before you can back up or restore files associated with Windows Media Services.
- You can only back up the System State data on a local computer You cannot back up the System State data on a remote computer
- If you are using Removable Storage to manage media, or you are using the Remote Storage to store data, then you should regularly back up the files that are in the following folders:

  Systemroot\System32\Ntmsdata

  Systemroot\System32\Remotestorage

  This ensures that all Removable Storage and Remote Storage data can be restored.

⇡Top of page

Examples

Example 1

The following example performs a normal backup named "My Job 1" of the remote share \\iggy-multi\c$. This example pulls a tape from the Backup media pool, and name the tape "Command Line Backup 1." The description of the backup job is "Command Line Functionality." The backup is verified after the backup job is complete, access is not restricted to the owner/administrator, the logging level is set to summary only, Remote Storage data is not backed up, and hardware compression is enabled.

```
ntbackup backup \\iggy-multi\c$ /m normal /j "My Job 1" /p "Backup" /n "Command Line Backup 1" /d "Command Line Functionality" /v:yes /r:no /l:s /rs:no /hc:on
```

Example 2

The following example performs a copy backup named "My Job 2" of the local drive D:\. The backed up files and folders are appended to the tape named "Command Line Backup 1." All other options default to those specified in the Backup program.

```
ntbackup backup d:\ /j "My Job 2" /a /t "Command Line Backup 1" /m copy
```

Example 3

The following example performs a backup using the backup type that is specified in the Backup program. It uses the backup selection file named Commandline.bks, located in the C:\Program Files\Windows NT\ntbackup\data\ directory to choose which files to backup. The backup job is named "My Job 3" and it overwrites the tape named "Command Line Backup 1" with the new name "Command Line Backup 2."

```
ntbackup backup "@C:\Program Files\Windows NT\ntbackup\data\commandline.bks" /j "My Job 3" /t "Command Line Backup 1" /n "Command Line Backup 2"
```

Example 4

The following examples show how to perform a backup to a file from the command line. All three examples use the Backup program's default values for the backup type, verification setting, logging level, hardware compression, and any other restrictions. The first example shows how to backup \\iggy-multi\d$ to the file D:\Backup.bkf. The second example shows how to append the same backup to the same file. The third example shows how to overwrite the file with the same backup. In all three examples a complete UNC name could be substituted for the drive letter (that is, instead of d:\backup.bkf, the user could specify \\iggy-multi\d$\backup.bkf as the backup destination).

```
ntbackup backup \\iggy-multi\d$ /j "Command Line Backup 4" /f "D:\backup.bkf"
```

```
ntbackup backup \\iggy-multi\d$ /j "Command Line Backup 5" /f "D:\backup.bkf" /a
```

```
ntbackup backup \\iggy-multi\d$ /j "Command Line Backup 6" /f "D:\backup.bkf"
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Backup overview

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ntcmdprompt

Runs the command interpreter Cmd.exe, rather than Command.com, after running a TSR or after starting the command prompt from within an MS-DOS application.

Syntax

ntcmdprompt

⇡Top of page

Parameters

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- When Command.com is running, some features of Cmd.exe, such as the **doskey** display of command history, are not available. If you would prefer to run the Cmd.exe command interpreter after you have started a TSR or started the command prompt from within an application based on MS-DOS, 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 program information file (PIF).

⇡Top of page

Examples

To include **ntcmdprompt** in your Config.nt file, or the configuration startup file specified in the PIF, type:

ntcmdprompt

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Doskey

Top of page

Manage Your Profile |Legal |Contact Us

Ntsd

Ntsd is included as a courtesy to software developers. Only system developers should use this command. For more information, see the help file included with NTSD.

Top of page

Manage Your Profile |Legal |Contact Us

Openfiles

Queries or displays open files. Also queries, displays, or disconnects files opened by network users.

To view the command syntax, click a command:

openfiles disconnect

Disconnects one or more remote users connected to open shared files.

Syntax

openfiles.exe /disconnect [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] {[**/id** *OpenFileID*]|[**/a** *UserName*]|[**/o** *OpenMode*]} [**/se** *SessionName*] [**/op** *OpenFileName*]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer. (Do not use backslashes.) The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/id *OpenFileID* : Disconnects the file opened with the specified numeric *OpenFileID* on the computer specified by the **/s** parameter. Use **openfiles.exe /query** to learn the file ID. The wildcard (*) can be used to disconnect all open files on the specified computer.

/a *UserName* : Disconnects all open files that were accessed by the specified user on the computer specified by the **/s** parameter. The wildcard (*) can be used to disconnect all open files on the specified computer.

/o ***OpenMode*** **:** Disconnects all open files with the specified *OpenMode* on the computer specified by the **/s** parameter. The *OpenMode* parameter includes the Read/Write and Read modes. The wildcard (*) can be used to disconnect all open files on the specified computer.

/se ***SessionName*** **:** Disconnects all open files that were created by the specified session on the computer specified by the **/s** parameter. The wildcard (*) can be used to disconnect all open files on the specified computer.

/op ***OpenFileName*** **:** Disconnects the open file that was created with the specified *OpenFileName* on the computer specified by the **/s** parameter. The wildcard (*) can be used to disconnect all open files on the specified computer.

/?: Displays help at the command prompt.

⇡Top of page

Examples

The following examples show how you can use the **openfiles /disconnect** command:

openfiles.exe /disconnect /id 1
openfiles.exe /disconnect /a hiropln
openfiles.exe /disconnect /o read/write
openfiles.exe /disconnect /op "c:\my documents\somedoc.doc"
openfiles /disconnect /session machine /id 5
openfiles.exe /disconnect /s srvmain /u maindom\hiropln /id 5
openfiles /disconnect /s srvmain /u maindom\hiropln /p p@ssW23 /id *

openfiles query

Queries and displays all open files.

⇡Top of page

Syntax

openfiles.exe /query [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] [**/fo** {**TABLE**|**LIST**|**CSV**}] [**/nh**] [**/v**]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer. (Do not use backslashes.) The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fo {TABLE|LIST|CSV} : Specifies the format to use for the query output. Valid values are **TABLE**, **LIST**, and **CSV**. The default value for output is **TABLE**.

/nh : Suppresses column header in the output. Valid only when the **/fo** parameter is set to **TABLE** or **CSV**.

/v : Specifies that verbose task information be displayed in the output.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **openfiles /query** command:

```
openfiles.exe /query
openfiles.exe /query /fo table /nh
openfiles.exe /query /fo list /v
openfiles.exe /query /s srvmain /u maindom\hiropln /p p@ssW23
```

Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Pagefileconfig.vbs

Enables an administrator to display and configure a system's paging file Virtual Memory settings.

For the command syntax, click a command:

pagefileconfig change

Changes a system's existing paging file Virtual Memory settings.

Syntax

pagefileconfig[**.vbs**] **/change** [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] {[**/i** *InitialPageFileSize*]|[**/m** *MaximumPageFileSize*]} **/vo** {*VolumeLetter*|*****} [**/vo** {*VolumeLetter2*|*****} [...]]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the script with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/i *InitialPageFileSize* : Specifies the new initial size (in MB) to use for the paging file specified.

/m *MaximumPageFileSize* : Specifies the new maximum size (in MB) to use for the paging file specified.

/vo {*VolumeLetter*|*} : Specifies the volume or volumes of the paging file settings to be changed. The volume is specified by a letter followed by a colon (for example, "C:").

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **pagefileconfig /change** command:

```
pagefileconfig.vbs /change /m 400 /vo c:
pagefileconfig.vbs /change /s srvmain /u maindom\hiropln /m 400 /vo c:
pagefileconfig.vbs /change /s srvmain /u maindom\hiropln /i 20 /vo *
pagefileconfig.vbs /change /s srvmain /u maindom\hiropln /p p@ssW23 /i 200 /m 500 /vo c: /vo d:
```

pagefileconfig create

Creates or adds an additional paging file to a system.

Top of page

Syntax

pagefileconfig.vbs /create [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] **/i** *InitialPageFileSize* **/m** *MaximumPageFileSize* **/vo** {*VolumeLetter*|*****} [**/vo** {*VolumeLetter2*|*****} [...]]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the script with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/i *InitialPageFileSize* : Specifies the new initial size (in MB) to use for the paging file specified.

/m *MaximumPageFileSize* : Specifies the new maximum size (in MB) to use for the paging file specified.

/vo {*VolumeLetter*|*} **:** Specifies the volume or volumes of the paging file settings to be created. The volume is specified by a letter followed by a colon (for example, "C:").

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **pagefileconfig /create** command:

pagefileconfig.vbs /create /i 140 /m 300 /vo d:
pagefileconfig.vbs /create /s srvmain /u maindom\hiropln /i 150 /m 300 /vo d:
pagefileconfig.vbs /create /s srvmain /u maindom\hiropln /i 50 /m 200 /vo *
pagefileconfig.vbs /create /s srvmain /u maindom\hiropln /p p@ssW23 /i 100 /m 600 /vo d: /vo e: /vo f:

pagefileconfig delete

Deletes a paging file from a system.

Top of page

Syntax

pagefileconfig.vbs /delete [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] **/vo** {*VolumeLetter*|*****} [**/vo** {*VolumeLetter2*|*****} [...]]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the script with the account permissions of the user specified by *User* or

Domain\User. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/vo {*VolumeLetter*|*} : Specifies the volume or volumes of the paging file settings to be deleted. The volume is specified by a letter followed by a colon (for example, "C:").

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **pagefileconfig /delete** command:

pagefileconfig.vbs /delete /vo d:
pagefileconfig.vbs /delete /s srvmain /u maindom\hiropln /vo d:
pagefileconfig.vbs /delete /s srvmain /u maindom\hiropln /p p@ssW23 /vo d: /vo e: /vo f:

pagefileconfig query

Queries and displays a system's paging file Virtual Memory settings.

pagefileconfig.vbs /query [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] [**/fo** {**TABLE**|**LIST**|**CSV**}]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the script with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fo {TABLE|LIST|CSV} : Specifies the format to use for the query output. Valid values are **TABLE**, **LIST**, and **CSV**. The default value is **LIST**.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **pagefileconfig /query** command:

pagefileconfig.vbs /query
pagefileconfig.vbs /query /fo table
pagefileconfig.vbs /query /s srvmain /u maindom\hiropln
pagefileconfig.vbs /query /s srvmain /u maindom\hiropln /p p@ssW23 /fo list

Top of page

Remarks

- Maximum paging file size is limited to available free disk space less the crashdump recovery settings for the boot drive.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Path

Sets the command path in the PATH environment variable, which is the set of directories used to search for executable files. Used without parameters, **path** displays the current command path.

Syntax

path [[**%path%**] [*Drive***:**]*Path* [**;**...]]

⇡Top of page

Parameters

[*Drive*:]*Path* : Specifies the drive and directory to set in the command path.

; : Separates directories in the command path.

%path% : Specifies Windows XP to append the command path to the existing set of directories listed in the PATH environment variable.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Using **;**

  When used as the only parameter, **;** deletes the existing command path value found in the PATH variable.

- Using **%path%**

  When you include **%path%** in the syntax, Cmd.exe replaces it with the command path value found in the PATH variable, eliminating the need to manually enter these values at the command line. For more information about substituting environment variable values, see Command shell overview in Related Topics.
- Current directory

  The operating system always searches in the current directory first, before it searches the directories in the command path.
- Files with the same name, different extensions

  You might have some files in the same directory that share the same file name 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.

  The operating system searches for a file by using default file name 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 at the command line.
- Two or more identical file names in the path

  If you have two or more files in the command path that have the same file name and extension, Windows XP searches for the specified file name first in the current directory, and then it searches the directories in the command path in the order in which they are listed in PATH.
- Searching the MS-DOS subsystem

  If you place the **path** command in your Autoexec.nt file, it automatically appends the specified MS-DOS subsystem search path to the Windows XP search path every time you log on to your computer. Cmd.exe does not use the Autoexec.nt file. When started from a shortcut, Cmd.exe inherits the environment variables set in My Computer/Properties/Advanced/Environment.

Top of page

Examples

The following command specifies that Windows XP 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
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command shell overview

Top of page

Manage Your Profile |Legal |Contact Us

Pathping

Provides information about network latency and network loss at intermediate hops between a source and destination. **Pathping** sends multiple Echo Request messages to each router between a source and destination over a period of time and then computes results based on the packets returned from each router. Because **pathping** displays the degree of packet loss at any given router or link, you can determine which routers or subnets might be having network problems. **Pathping** performs the equivalent of the **tracert** command by identifying which routers are on the path. It then sends pings periodically to all of the routers over a specified time period and computes statistics based on the number returned from each. Used without parameters, **pathping** displays help.

Syntax

pathping [**-n**] [**-h** *MaximumHops*] [**-g** *HostList*] [**-p** *Period*] [**-q** *NumQueries* [**-w** *Timeout*] [**-T**] [**-R**] [*TargetName*]

⇡Top of page

Parameters

-n : Prevents **pathping** from attempting to resolve the IP addresses of intermediate routers to their names. This might expedite the display of **pathping** results.

-h *MaximumHops* : Specifies the maximum number of hops in the path to search for the target (destination). The default is 30 hops.

-g *HostList* : Specifies that the Echo Request messages use the Loose Source Route option in the IP header with the set of intermediate destinations specified in *HostList*. With loose source routing, successive intermediate destinations can be separated by one or multiple routers. The maximum number of addresses or names in the host list is 9. The *HostList* is a series of IP addresses (in dotted decimal notation) separated by spaces.

-p *Period* : Specifies the number of milliseconds to wait between consecutive pings. The default is 250 milliseconds (1/4 second).

-q *NumQueries* : Specifies the number of Echo Request messages sent to each router in the path.

The default is 100 queries.

-w *Timeout* : Specifies the number of milliseconds to wait for each reply. The default is 3000 milliseconds (3 seconds).

-T : Attaches a layer-2 priority tag (for example, 802.1p) to the Echo Request messages that it sends to each of the network devices along the route. This helps to identify network devices that do not have layer-2 priority capability. This switch is used to test for Quality of Service (QoS) connectivity.

-R : Determines whether each network device along the route supports the Resource Reservation Protocol (RSVP), which allows the host computer to reserve a specified amount of bandwidth for a data stream. This switch is used to test for Quality of Service (QoS) connectivity.

***TargetName* :** Specifies the destination, which is identified either by IP address or host name.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Pathping parameters are case-sensitive.
- To avoid network congestion, pings should be sent at a sufficiently slow pace.
- To minimize the effects of burst losses, do not send pings too frequently.
- When using the **-p** parameter, pings are sent individually to each intermediate hop. Because of this, the interval between two pings sent to the same hop is *period* multiplied by the number of hops.
- When using the **-w** parameter, multiple pings can be sent in parallel. Because of this, the amount of time specified in the *Timeout* parameter is not bounded by the amount of time specified in the *Period* parameter for waiting between pings.
- Using the **-T** parameter

  Enabling layer-2 priority on the host computer allows packets to be sent with a layer-2 priority tag, which can be used by layer-2 devices to assign a priority to the packet. Legacy devices that do not recognize layer-2 priority will discard these packets, since they appear to be malformed. This parameter helps identify network computer that are discarding these packets.

- Using the **-R** parameter

  An RSVP reservation message for a nonexistent session is sent to each network device on the route. If the device does not support RSVP, it returns an Internet Control Message Protocol (ICMP) Destination Unreachable-Protocol Unreachable message. If the device does support RSVP, it returns an RSVP Reservation Error message. Some devices might not return either of these messages. If this occurs, a time-out message is displayed.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

The following example shows pathping command output:

```
D:\>pathping -n corp1

Tracing route to corp1 [10.54.1.196]
over a maximum of 30 hops:
  0  172.16.87.35
  1  172.16.87.218
  2  192.168.52.1
  3  192.168.80.1
  4  10.54.247.14
  5  10.54.1.196

Computing statistics for 125 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           172.16.87.35
                                0/ 100 =  0%   |
  1   41ms     0/ 100 =  0%     0/ 100 =  0%  172.16.87.218
                               13/ 100 = 13%   |
  2   22ms    16/ 100 = 16%     3/ 100 =  3%  192.168.52.1
                                0/ 100 =  0%   |
  3   24ms    13/ 100 = 13%     0/ 100 =  0%  192.168.80.1
                                0/ 100 =  0%   |
  4   21ms    14/ 100 = 14%     1/ 100 =  1%  10.54.247.14
                                0/ 100 =  0%   |
  5   24ms    13/ 100 = 13%     0/ 100 =  0%  10.54.1.196

Trace complete.
```

When **pathping** is run, the first results list the path. This is the same path that is shown using the **tracert** command. Next, a busy message is displayed for approximately 90 seconds (the time varies by hop count). During this time, information is gathered from all routers previously listed and from the links between them. At the end of this period, the test results are displayed.

In the sample report above, the **This Node/Link**, **Lost/Sent = Pct** and **Address** columns show that the link between 172.16.87.218 and 192.168.52.1 is dropping 13 percent of the packets. The routers at hops 2 and 4 also are dropping packets addressed to them, but this loss does not affect their ability to forward traffic that is not addressed to them.

The loss rates displayed for the links, identified as a vertical bar (|) in the **Address** column, indicate link congestion that is causing the loss of packets that are being forwarded on the path. The loss rates displayed for routers (identified by their IP addresses) indicate that these routers might be overloaded.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Tracert

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Pause

Suspends processing of a batch program and displays a message prompting the user to press any key to continue.

Syntax

pause

Top of page

Parameters

/? : Displays help at the command prompt.

Top of page

Remarks

- When you run **prompt** command, the following message appears:

  Press any key to continue . . .
- If you press CTRL+C to stop a batch program, the following message appears:

  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.

Top of page

Examples

To create a batch program that prompts the user to change disks in one of the drives, type:

```
@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.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Pbadmin

Administers phone books. Used without parameters, **pbadmin** starts Phone Book Administrator.

Syntax

pbadmin.exe /N *PhoneBookName* [**/R** *RegionFilePath\RegionFileName*] [**/P** *DataFilePath\DataFileName*]

pbadmin.exe /I *PhoneBookName* **/R** *RegionFilePath\RegionFileName*

pbadmin.exe /I *PhoneBookName* **/P** *DataFilePath\DataFileName*

pbadmin.exe /O *PhoneBookName ServerName UserName Password*

pbadmin.exe /B *PhoneBookName*

⇡Top of page

Parameters

/N *PhoneBookName* : Creates a phone book (.mdb). The *PhoneBookName* parameter specifies the name of the file to create.

/I *PhoneBookName* : Imports data into a phone book (.mdb) from a region (.txt) file or a phone book (.pbk or .txt) file. The *PhoneBookName* parameter specifies the phone book into which you want to import data. You must use the **/R** or the **/P** parameter in conjunction with the **/I** parameter.

/R *RegionFilePath\RegionFileName* : Specifies the location and the name of the region file you want to import into a phone book.

/P *DataFilePath\DataFileName* : Specifies the location and the name of the phone book file (. pbk or .txt) you want to import into a phone book.

/O *PhoneBookName ServerName UserName Password* : Configures options for a phone book. The *PhoneBookName* parameter specifies the name of the phone book for which you want to set options. The *ServerName* parameter specifies the name of the server to which the phone book is published. The *UserName* parameter specifies a user account with FTP permissions on the Phone Book Service (PBS) server. The *Password* parameter specifies the password for the user account on the PBS server.

/B ***PhoneBookName*** **:** Publishes a phone book. The *PhoneBookName* parameter specifies the name of the phone book to publish. Before you can use the **/B** parameter, you must configure the phone book with the **/O** parameter.

/?: Displays help at the command prompt.

Top of page

Remarks

- To create a phone book (.mdb), use this syntax:

  pbadmin.exe /N *PhoneBookName* [**/R** *RegionFilePath\RegionFileName*] [**/P** *DataFilePath\DataFileName*]
- To import data into a phone book from a region file, use this syntax:

  pbadmin.exe /I *PhoneBookName* **/R** *RegionFilePath\RegionFileName*
- To import data into a phone book from a phone book file (.pbk) or an appropriately formatted text file (.txt), use this syntax:

  pbadmin.exe /I *PhoneBookName* **/P** *DataFilePath\DataFileName*
- To configure publishing options for a phone book, use this syntax:

  pbadmin.exe /O *PhoneBookName ServerName UserName Password*
- To publish a phone book, use this syntax:

  pbadmin.exe /B *PhoneBookName*
- For **pbadmin**, the term "phone book" refers to a phone book (.mdb), and the term "phone book file" refers to a phone book file (.pbk).
- Command-line parameters are case-sensitive.
- A phone book name cannot consist of more than eight characters, cannot consist of all digits, and must not contain a space or any of the following symbols: **! , ; * = / \ : ? ' " < > | . & % {} [] @ ( ) ´ ` ~**
- If you specify a path, you must specify the full path. You cannot use spaces in the path to the phone book, to a phone book file, or to a region file.

- To use the **pbadmin** command, you must install Phone Book Administrator and then type the command from within the PBA directory. To install Phone Book Administrator, insert your Windows XP Professional or your Windows 2000 installation CD into your CD-ROM drive, and click **Browse this CD**. Open the **VALUEADD** folder, open the **MSFT** folder, open the **MGMT** folder, and open the **PBA** folder. Double-click **PBAINST**, and follow the instructions on your screen.

Top of page

Examples

In the examples below, fictional data is used.

To create a phone book named Awesome, type:

pbadmin.exe /N Awesome

To import data into a phone book named Awesome from a region file named Region1.txt located in the C:\Temp directory, type:

pbadmin.exe /I Awesome /R C:\Temp\Region1.txt

To configure a phone book file with the name of Awesome, the server name of Rainier, the user name of jobrown, and the password of mypassword, type:

pbadmin.exe /O Awesome Rainier jobrown mypassword

To publish a phone book named Awesome, type:

pbadmin.exe /B Awesome

Command shell overview

Top of page

Manage Your Profile |Legal |Contact Us

Pentnt

Detects floating point division error (if present) in the Pentium chip, disables floating point hardware, and turns on floating point emulation.

Syntax

pentnt [**-c**] [**-f**] [**-o**]

⇡Top of page

Parameters

-c : Enables conditional emulation.

-f : Enables forced emulation.

-o : Disables forced emulation and reenables floating-point hardware if it is present.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Parameters used with this command must be prefixed with **-** (hyphen) rather than **/** (slash).
- Using the **-c** command-line option

  Floating-point emulation is forced on 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.
- Using the **-f** command-line option Floating-point hardware is disabled and floating-point emulation is always 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.

- Using the **-o** command-line option If you select this parameter, you must restart the computer for the changes to take effect.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Perfmon

Allows you to open a Windows XP Performance console configured with settings files from Windows NT 4.0 version of Performance Monitor.

Syntax

perfmon.exe [*file_name*] [**/HTMLFILE:***converted_file settings_file*]

⇡Top of page

Parameters

.exe : Specifies the name of the file extension.

***file_name* :** Specifies the name of the settings file.

/HTMLFILE:*converted_file settings_file* : Specifies the name of the converted files, and the name of the original Windows NT 4.0 settings file.

⇡Top of page

Remarks

- This procedure works for the following types of Windows NT 4.0 version of Performance Monitor settings files: chart (.pmc), report (.pmr), alert (.pma), and log (.pml).
- To display the Windows NT 4.0 settings file in System Monitor, the system temporarily converts the file for use with Windows XP System Monitor, then discards the converted version after the console starts. If you want to save the settings file for permanent use with System Monitor, type:

  Perfmon [file_name] [/HTMLFILE:converted_file settings_file]

  Where **/HTMLFILE:converted_file** is the name given to the converted file and **settings_file** is the name of the original Windows NT 4.0 settings file.

- For more information about incorporating **Perfmon** into your Windows Management Instrumentation (WMI) scripts, see Scripting Logs and Monitoring at the Microsoft Windows Resource Kits Web site.

Top of page

Examples

To list your settings file, type:

Perfmon myfile_overview

To list converted and original file names, type:

Perfmon myfile_overview /Commands:newfile_review oldfile_sample.pml

Use Windows NT 4.0 version of Performance Monitor settings files

New ways to do familiar tasks

Performance monitoring in Windows NT 4.0

Understanding Performance

Command-line reference A-Z

Top of page

Ping

Verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The receipt of corresponding Echo Reply messages are displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution. Used without parameters, **ping** displays help.

Syntax

ping [**-t**] [**-a**] [**-n** *Count*] [**-l** *Size*] [**-f**] [**-i** *TTL*] [**-v** *TOS*] [**-r** *Count*] [**-s** *Count*] [{**-j** *HostList* | **-k** *HostList*}] [**-w** *Timeout*] [*TargetName*]

Top of page

Parameters

-t : Specifies that ping continue sending Echo Request messages to the destination until interrupted. To interrupt and display statistics, press CTRL-BREAK. To interrupt and quit ping, press CTRL-C.

-a : Specifies that reverse name resolution is performed on the destination IP address. If this is successful, ping displays the corresponding host name.

-n *Count* : Specifies the number of Echo Request messages sent. The default is 4.

-l *Size* : Specifies the length, in bytes, of the Data field in the Echo Request messages sent. The default is 32. The maximum *size* is 65,527.

-f : Specifies that Echo Request messages are sent with the Don't Fragment flag in the IP header set to 1. The Echo Request message cannot be fragmented by routers in the path to the destination. This parameter is useful for troubleshooting path Maximum Transmission Unit (PMTU) problems.

-i *TTL* : Specifies the value of the TTL field in the IP header for Echo Request messages sent. The default is the default TTL value for the host. For Windows XP hosts, this is typically 128. The maximum *TTL* is 255.

-v ***TOS*** **:** Specifies the value of the Type of Service (TOS) field in the IP header for Echo Request messages sent. The default is 0. *TOS* is specified as a decimal value from 0 to 255.

-r ***Count*** **:** Specifies that the Record Route option in the IP header is used to record the path taken by the Echo Request message and corresponding Echo Reply message. Each hop in the path uses an entry in the Record Route option. If possible, specify a *Count* that is equal to or greater than the number of hops between the source and destination. The *Count* must be a minimum of 1 and a maximum of 9.

-s ***Count*** **:** Specifies that the Internet Timestamp option in the IP header is used to record the time of arrival for the Echo Request message and corresponding Echo Reply message for each hop. The *Count* must be a minimum of 1 and a maximum of 4.

-j ***HostList*** **:** Specifies that the Echo Request messages use the Loose Source Route option in the IP header with the set of intermediate destinations specified in *HostList*. With loose source routing, successive intermediate destinations can be separated by one or multiple routers. The maximum number of addresses or names in the host list is 9. The host list is a series of IP addresses (in dotted decimal notation) separated by spaces.

-k ***HostList*** **:** Specifies that the Echo Request messages use the Strict Source Route option in the IP header with the set of intermediate destinations specified in *HostList*. With strict source routing, the next intermediate destination must be directly reachable (it must be a neighbor on an interface of the router). The maximum number of addresses or names in the host list is 9. The host list is a series of IP addresses (in dotted decimal notation) separated by spaces.

-w ***Timeout*** **:** Specifies the amount of time, in milliseconds, to wait for the Echo Reply message that corresponds to a given Echo Request message to be received. If the Echo Reply message is not received within the time-out, the "Request timed out" error message is displayed. The default time-out is 4000 (4 seconds).

TargetName **:** Specifies the destination, which is identified either by IP address or host name.

/? : Displays help at the command prompt.

Top of page

Remarks

- You can use **ping** to test both the computer name and the IP address of the computer. If pinging the IP address is successful, but pinging the computer name is not, you might have a name resolution problem. In this case, ensure that the computer name you are specifying can be resolved through the local Hosts file, by using Domain Name System (DNS) queries, or through NetBIOS name resolution techniques.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

The following example shows **ping** command output:

```
C:\>ping example.microsoft.com

Pinging example.microsoft.com [192.168.239.132] with 32 bytes of data:

Reply from 192.168.239.132: bytes=32 time=101ms TTL=124

Reply from 192.168.239.132: bytes=32 time=100ms TTL=124

Reply from 192.168.239.132: bytes=32 time=120ms TTL=124

Reply from 192.168.239.132: bytes=32 time=120ms TTL=124
```

To ping the destination 10.0.99.221 and resolve 10.0.99.221 to its host name, type:

```
ping -a 10.0.99.221
```

To ping the destination 10.0.99.221 with 10 Echo Request messages, each of which has a Data field of 1000 bytes, type:

```
ping -n 10 -l 1000 10.0.99.221
```

To ping the destination 10.0.99.221 and record the route for 4 hops, type:

```
ping -r 4 10.0.99.221
```

To ping the destination 10.0.99.221 and specify the loose source route of 10.12.0.1-10.29.3.1-10.1.44.1, type:

```
ping -j 10.12.0.1 10.29.3.1 10.1.44.1 10.0.99.221
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Popd

Changes the current directory to the directory stored by the **pushd** command.

Syntax

popd

⇡Top of page

Parameters

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Every time you use the **pushd** command, a single directory is stored for your use. However, you can store multiple directories by using the **pushd** command multiple times.

  The directories are stored sequentially in a virtual stack. If you use the **pushd** command once, the directory in which you use the command is placed at the bottom of the stack. If you use the command again, the second directory is placed on top of the first one. The process repeats every time you use the **pushd** command.

  You can use the **popd** command to change the current directory to the directory most recently stored by the **pushd** command. If you use the **popd** command, the directory on the top of the stack is removed from the stack as the current directory is changed to that directory. If you use the **popd** command again, the next directory on the stack is removed.
- When command extensions are enabled, the **popd** command removes any drive-letter assignations created by **pushd**.

⇡Top of page

Examples

You can use **pushd** and **popd** in a batch program to change the current directory from the one in which the batch program was run and then change it back. The following sample batch program shows how to do this:

```
@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
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Pushd

Cmd

Command-line reference A-Z

Top of page

Print

Sends a text file to a printer.

Syntax

print [**/d:***Printer*] [*Drive***:**][*Path*] *FileName* [**...**]

⇡Top of page

Parameters

/d:*Printer* : Specifies the printer on which you want to print the job. You can specify a local printer by specifying the port on your computer to which the printer is connected. 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 queue name (\\ServerName *ShareName*). If you do not specify a printer, the print job is sent to LPT1.

***Drive*: :** Specifies the logical or physical drive on which the file you want to print is located. This parameter is not required if the file you want to print is located on the current drive.

***Path* :** Specifies where, on the drive, the file you want to print is located. This parameter is not required if the file you want to print is located in the current directory.

***FileName* [...] :** Required. Specifies, by name, the file you want to print. You can include multiple files in one command.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- A file can print in the background if you send it to a printer connected to a serial or parallel port on the local computer.

- Many programs have their own print commands. You should use the **print** command for a program to print files that you create with that program.
- You can perform many configuration tasks from the command line by using the **mode** command. For more information about configuring a printer connected to a parallel port, configuring a printer connected to a serial port, displaying the status of a printer, or preparing a printer for code page switching, see **mode** in Related Topics.

⇡Top of page

Examples

To send the file Report.txt in the current directory to a printer connected to LPT2 on the local computer, type:

print /d:LPT2 report.txt

To send the file Report.txt in the c:\Accounting directory to the Printer1 print queue on the \\CopyRoom server, type:

print /d:\\copyroom\printer1 c:\accounting\report.txt

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Mode

Top of page

Manage Your Profile |Legal |Contact Us

Prncnfg.vbs

Configures or displays configuration information about a printer. Used without parameters, **prncnfg.vbs** displays command-line help for the **prncnfg.vbs** command.

To display configuration information about a printer

Syntax

cscript prncnfg.vbs -g [**-s** *RemoteComputer*] **-p** *PrinterName* [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-g : Required. Specifies that you want to display configuration information about a printer.

-s *RemoteComputer* : Specifies, by name, the remote computer that manages the printer about which you want to display information. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Required. Specifies, by name, the printer about which you want to display information.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer that hosts the printer about which you want to display information. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

Top of page

Example

To display configuration information for the printer named ColorPrinter_2 whose print queue is hosted by the remote computer named HRServer, type:

cscript prncnfg.vbs -g -s HRServer -p ColorPrinter_2

To configure a printer

Top of page

Syntax

cscript prncnfg.vbs -t [**-s** *RemoteComputer*] **-p** *PrinterName* [**-u** *UserName* **-w** *Password*] [**-r** *PortName*] [**-l** *Location*] [**-m** *Comment*] [**-h** *ShareName*] [**-f** *SeparatorText*] [**-y** *DataType*] [**-st** *StartTime*] [**-ut** *EndTime*] [**-o** *Priority*] [**-i** *DefaultPriority*] [{+ | -}**shared**] [{+ | -}**direct**] [{+ | -}**published**] [{+ | -}**hidden**] [{+ | -}**rawonly**] [{+ | -}**queued**] [{+ | -}**keepprintedjobs**] [{+ | -}**workoffline**] [{+ | -}**enabledevq**] [{+ | -}**docompletefirst**][{+ | -}**enablebidi**]

Top of page

Parameters

-t : Required. Specifies that you want to configure a printer.

-s *RemoteComputer* : Specifies, by name, the remote computer that manages the printer you want to configure. If you do not specify a computer, the printer is configured on the local computer.

-p *PrinterName* : Required. Specifies, by name, the printer you want to configure.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer on which you want to configure a printer. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

-r *PortName* : Specifies the port to which the printer is connected. If this is a parallel or a serial port, then use the ID of the port (for example, LPT1 or COM1). If this is a TCP/IP port, then use the port name that was specified when the port was added. For more information, see Related Topics.

-l *Location* : Specifies the printer location, such as "Copier Room."

-m *Comment* : Specifies the comment string.

-h *ShareName* : Specifies the share name.

-f *SeparatorText* : Specifies a file that contains the text that appears on the separator page.

-y *DataType* : Specifies the data types that the printer can accept. For more information on data types, see Related Topics.

-st *StartTime* : Configures the printer for limited availability. Specifies the time of day after which the printer is available. If you send a document to a printer when it is unavailable, the document is held (spooled) until the printer becomes available. You must specify time as a 24-hour clock. For example, to specify 11 P.M., type **2300**.

-ut *EndTime* : Configures the printer for limited availability. Specifies the time of day after which the printer is no longer available. If you send a document to a printer when it is unavailable, the document is held (spooled) until the printer becomes available. You must specify time as a 24-hour clock. For example, to specify 11 P.M., type **2300**.

-o *Priority* : Specifies a priority that the spooler uses to route print jobs. A print queue with a higher priority receives all its jobs before any queue with a lower priority.

-i *DefaultPriority* : Specifies the default priority assigned to each print job.

{+ | -}shared : Specifies whether this printer is shared on the network.

{+ | -}direct : Specifies whether the document should be sent directly to the printer without being spooled.

{+ | -}published : Specifies whether this printer should be published in Active Directory. If you publish the printer, other users can search for it based on its location and capabilities, such as color printing and stapling.

{+ | -}hidden : Reserved function.

{+ | -}rawonly : Specifies whether only raw data print jobs can be spooled on this queue.

{+ | -}queued : Specifies that the printer should not begin to print until after the last page of the document is spooled. The printing program is unavailable until the document has finished printing. However, using this option ensures that the whole document is available to the printer.

{+ | -}keepprintedjobs : Specifies whether the spooler should retain documents after they are printed. Enabling this option allows a user to resubmit a document to the printer from the print queue instead of from the printing program.

{+ | -}workoffline : Specifies whether you should be able to send print jobs to the print queue even if your computer is not connected to the network.

{+ | -}enabledevq : Specifies whether print jobs that do not match the printer setup (for example, PostScript files spooled to non-PostScript printers) should be held in the queue rather than being printed.

{+ | -}docompletefirst : Specifies whether the spooler should send to the appropriate queue print jobs with a lower priority that have completed spooling before sending to the same queue print jobs with a higher priority that have not completed spooling. If this option is enabled and no documents have completed spooling, the spooler will send larger documents before smaller ones. You should enable this option if you want to maximize printer efficiency at the cost of job priority. If this option is disabled, the spooler always sends higher priority jobs to their respective queues first.

{+ | -}enablebidi : Specifies whether the printer sends status information to the spooler.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Example

To configure a printer named ColorPrinter_2 so that the spooler in the remote computer named HRServer keeps print jobs after they have been printed, type:

cscript prncnfg.vbs -t -s HRServer -p ColorPrinter_2 +keepprintedjobs

To change the name of a printer

⇡Top of page

Syntax

cscript prncnfg.vbs -x [**-s** *RemoteComputer*] **-p** *PrinterName* **-z** *NewPrinterName* [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-x : Required. Specifies that you want to change the name of a printer.

-s *RemoteComputer* : Specifies, by name, the remote computer that manages the printer you want to rename. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Required. Specifies the current printer name.

-z *NewPrinterName* : Required. Specifies the new printer name.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer that hosts the printer you want to rename. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be

logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, **"***Computer Name***"**).

↑Top of page

Example

To change the name of a printer on the remote computer named HRServer from ColorPrinter_2 to ColorPrinter_3, type:

```
cscript prncnfg.vbs -x -s HRServer -p ColorPrinter_2 -z ColorPrinter_3
```

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one

Courier font	Code or program output

Prndrvr.vbs

Prnjobs.vbs

Prnmngr.vbs

Prnport.vbs

Prnqctl.vbs

Command-line reference A-Z

Print processors and data types

WMI overview

WMI overview

Top of page

Manage Your Profile |Legal |Contact Us

Prndrvr.vbs

Adds, deletes, and lists printer drivers. Used without parameters, **prndrvr.vbs** displays command-line help for the **prndrvr.vbs** command.

To install a printer driver

Syntax

cscript prndrvr.vbs -a [**-m** *DriverName*] [**-v** {**0** | **1** | **2** | **3**}] [**-e** *Environment*] [**-s** *RemoteComputer*] [**-h** *Path*] [**-i** *FileName***.inf**] [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-a : Required. Specifies that you want to install a driver.

-m *DriverName* : Specifies, by name, the driver you want to install. Drivers are often named for the model of printer they support. See the printer documentation for more information.

-v {0 | 1 | 2 | 3} : Specifies the version of the driver you want to install. See the description of the **-e** *Environment* parameter for information on which versions are available for which environment. If you do not specify a version, the version of the driver appropriate for the version of Windows running on the computer on which you are installing the driver is installed.

- Version **0** supports Windows 95, Windows 98, and Windows Millennium Edition.
- Version **1** supports Windows NT 3.51.
- Version **2** supports Windows NT 4.0.
- Version **3** supports Windows XP and Windows 2000.

-e *Environment* : Specifies the environment for the driver you want to install. If you do not specify an environment, the environment of the computer on which you are installing the driver is used. The following table lists the driver environments that are available and the versions that are available for each.

Environment	Available versions
"Windows NT x86"	1, 2, and 3
"Windows NT Alpha_AXP"	1 and 2
"Windows IA64"	3
"Windows NT R4000"	1
"Windows NT PowerPC"	1
"Windows 4.0"	0

-s *RemoteComputer* : Specifies the remote computer on which you want to install the driver. If you do not specify a computer, the driver is installed on the local computer.

-h *Path* : Specifies the path to the driver file. If you do not specify a path, the path to the location from which Windows was installed is used.

-i *FileName*.inf : Specifies the file name for the driver you want to install. If you do not specify a file name, ntprint.inf is used.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer on which you want to install the driver. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, **"***Computer Name***"**).

Top of page

Example

To add version 3 of the printer driver named "Color Printer Driver 1" to the local computer, which offers a Windows NT x86-based environment, type:

cscript prndrvr.vbs -a -m "Color Printer Driver 1" -v 3 -e "Windows NT x86"

To delete a printer driver

⇡Top of page

Syntax

cscript prndrvr.vbs -d [**-s** *RemoteComputer*] **-m** *DriverName* **-v** {**0** | **1** | **2** | **3**} **-e** *Environment*

[**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-d : Required. Specifies that you want to delete a driver.

-s *RemoteComputer* : Specifies the remote computer from which you want to delete the driver. If you do not specify a computer, the driver is deleted from the local computer.

-m *DriverName* : Required. Specifies, by name, the driver you want to delete. Drivers are often named for the model of printer they support. See the printer documentation for more information.

-v {0 | 1 | 2 | 3} : Required. Indicates the version of the driver to be deleted. See the description of the **-e** *Environment* parameter for information on which versions are available in which environment.

- Version **0** supports Windows 95, Windows 98, and Windows Millennium Edition.
- Version **1** supports Windows NT 3.51.
- Version **2** supports Windows NT 4.0.
- Version **3** supports Windows XP and Windows 2000.

-e *Environment* : Required. Specifies the environment for the driver you want to delete. The following table lists the driver environments that are available and the versions that are available for each.

Environment	Available versions
"Windows NT x86"	1, 2, and 3
"Windows NT Alpha_AXP"	1 and 2
"Windows IA64"	3
"Windows NT R4000"	1
"Windows NT PowerPC"	1
"Windows 4.0"	0

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer from which you want to delete the driver. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- When you delete a printer driver, you must include both the environment and the version parameters.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To list the printer drivers on a computer

Top of page

Syntax

cscript prndrvr.vbs -l [**-s** *RemoteComputer*] [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-l : Required. Specifies that you want to list all the drivers on a computer.

-s *RemoteComputer* : Indicates the remote computer whose drivers you want to list. If you do not specify a computer, drivers on the local computer are listed.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer whose drivers you want to list. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, **"***Computer Name***"**).

To delete all unused printer drivers from a computer

⇡Top of page

Syntax

cscript prndrvr.vbs -x [**-s** *RemoteComputer*] [**-u** *UserName* **-w** *Password*]

↑Top of page

Parameters

-x : Required. Specifies that you want to delete all unused printer drivers from a computer.

-s *RemoteComputer* : Specifies the remote computer from which you want to delete drivers. If you do not specify a computer, drivers are deleted from the local computer.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer from which you want to delete drivers. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Prncnfg.vbs

Prnjobs.vbs

Prnmngr.vbs

Prnport.vbs

Prnqctl.vbs

Command-line reference A-Z

WMI overview

WMI overview

Top of page

Manage Your Profile |Legal |Contact Us

Prnjobs.vbs

Pauses, resumes, cancels, and lists print jobs. Used without parameters, **prnjobs.vbs** displays command-line help for the **prnjobs.vbs** command.

To pause a print job

Syntax

cscript prnjobs -z [**-s** *RemoteComputer*] **-p** *PrinterName* **-j** *JobNumber* [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-z : Required. Specifies that you want to pause a print job.

-s *RemoteComputer* : Specifies, by name, the remote computer to which the print job you want to pause was sent. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Required. Specifies, by name, the printer that would print the job you want to pause.

-j *JobNumber* : Required. Specifies, by ID number, the print job you want to pause.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer to which the print job you want to pause was sent. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- To display a list of print jobs and their ID numbers, use this command with the **-l** parameter.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Example

To pause a print job with a job ID of 27 sent to the remote computer named HRServer for printing on the printer named ColorPrinter, type:

cscript prnjobs.vbs -z -s HRServer -p ColorPrinter -j 27

To resume a print job

⇡Top of page

Syntax

cscript prnjobs -m [**-s** *RemoteComputer*] **-p** *PrinterName* **-j** *JobNumber* [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-m : Required. Specifies that you want to resume a print job.

-s *RemoteComputer* : Specifies, by name, the remote computer to which the print job you want to resume was sent. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Required. Specifies, by name, the printer that will print the job you want to resume.

-j *JobNumber* : Required. Specifies, by ID number, the print job you want to resume.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer to which the print job you want to resume was sent. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- To display a list of print jobs and their ID numbers, use this command with the **-l** parameter.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To cancel a print job

⇡Top of page

Syntax

cscript prnjobs -x [**-s** *RemoteComputer*] **-p** *PrinterName* **-j** *JobNumber* [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-x : Required. Specifies that you want to cancel a print job.

-s *RemoteComputer* : Specifies, by name, the remote computer to which the job you want to cancel was sent. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Required. Specifies, by name, the printer that would print the job that you want to cancel.

-j *JobNumber* : Required. Specifies, by ID number, the print job you want to cancel.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer to which the print job you want to cancel was sent. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- To display a list of print jobs and their ID numbers, use this command with the **-l** parameter.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To list the print jobs in a print queue

Top of page

Syntax

cscript prnjobs -l [**-s** *RemoteComputer*] [**-p** *PrinterName*] [**-u** *UserName* **-w** *Password*]

Top of page

Parameters

-l : Required. Specifies that you want to list all the print jobs in a print queue.

-s *RemoteComputer* : Specifies, by name, the remote computer that hosts the print queue whose jobs you want to list. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Specifies, by name, the printer whose print queue contains the jobs you want to list. If you do not specify a printer, then all jobs in all print queues are listed.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer that hosts the print queue whose jobs you want to list. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- To display a list of print jobs and their ID numbers, use this command with the **-l** parameter.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

↑Top of page

Example

To list all current print jobs in the queue for the local printer named ColorPrinter_2, type:

cscript prnjobs.vbs -l -p ColorPrinter_2

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply

Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Prncnfg.vbs

Prnjobs.vbs

Prnmngr.vbs

Prnport.vbs

Prnqctl.vbs

Command-line reference A-Z

WMI overview

WMI overview

Top of page

Manage Your Profile |Legal |Contact Us

Prnmngr.vbs

Adds, deletes, and lists printers or printer connections, in addition to setting and displaying the default printer. Used without parameters, **prnmngr.vbs** displays command-line help for the **prnmngr.vbs** command.

To add a local printer

Syntax

cscript prnmngr.vbs -a -p *PrinterName* [**-s** *RemoteComputer*] **-m** *DriverName* **-r** *PortName* [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-a : Required. Specifies that you want to add a local printer.

-s *RemoteComputer* : Specifies, by name, the remote computer to which you want to add a local printer. If you do not specify a computer, the printer is added to the local computer.

-p *PrinterName* : Required. Specifies, by name, the local printer that you want to add.

-m *DriverName* : Required. Specifies, by name, the driver for the local printer you want to add. Drivers are often named for the model of printer they support. See the printer documentation for more information.

-r *PortName* : Required. Specifies the port to which the printer is connected. If this is a parallel or a serial port, use the ID of the port (for example, LPT1 or COM1). If this is a TCP/IP port, use the port name that was specified when the port was added. For more information, see Related Topics.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer to which you want to add a local printer. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be

logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

⇧Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

⇧Top of page

Example

To add a printer named ColorPrinter_2 that is connected to LPT1 on the local computer and requires a printer driver called Color Printer Driver1, type:

cscript prnmngr.vbs -a -p ColorPrinter_2 -m "Color Printer Driver1" -r lpt1:

To add a printer connection

⇧Top of page

Syntax

cscript prnmngr.vbs -ac -p *PrinterName*

⇧Top of page

Parameters

-ac : Required. Specifies that you want to add a printer connection.

-p *PrinterName* : Required. Specifies, by name, the printer for which you want to add a connection.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To delete a printer

Top of page

Syntax

cscript prnmngr.vbs -d -p *PrinterName* [**-u** *UserName* **-w** *Password*] [**-s** *RemoteComputer*]

Top of page

Parameters

-d : Required. Specifies that you want to delete a printer.

-p *PrinterName* : Required. Specifies, by name, the printer that you want to delete.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer from which you want to delete a printer. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

-s *RemoteComputer* : Specifies, by name, the remote computer to which you want to add a local printer. If you do not specify a computer, the printer is added to the local computer.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Example

To delete the printer named ColorPrinter_2 from the remote computer named HRServer, type:

cscript prnmngr.vbs -d -s HRServer -p ColorPrinter_2

To delete all of the printers from a computer

⇡Top of page

Syntax

cscript prnmngr.vbs -x [**-s** *RemoteComputer*] [**-u** *UserName*] [**-w** *Password*]

⇡Top of page

Parameters

-x : Required. Specifies that you want to delete all printers from a computer.

-s *RemoteComputer* : Specifies, by name, the remote computer from which you want to delete all printers. If you do not specify a computer, all printers are deleted from the local computer.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer from which you want to delete all printers. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be

logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To display the default printer

↑Top of page

Syntax

cscript prnmngr.vbs -g

↑Top of page

Parameters

-g : Required. Specifies that you want to display the default printer.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.

- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To set the default printer

⇡Top of page

Syntax

cscript prnmngr.vbs -t -p *PrinterName*

⇡Top of page

Parameters

-t : Required. Specifies that you want to set the default printer.

-p *PrinterName* : Required. Specifies, by name, the printer that you want to set as the default printer.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To list all of the printers for a computer

⇡Top of page

Syntax

cscript prnmngr.vbs -l [**-s** *RemoteComputer*] [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-l : Required. Specifies that you want to list all the printers for a computer.

-s *RemoteComputer* : Specifies, by name, the remote computer for which you want to list printers. If you do not specify a computer, the printers added to the local computer are listed.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer for which you want to list printers. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information on WMI, see Related Topics.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line

Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Prncnfg.vbs

Prnjobs.vbs

Prndrvr.vbs

Prnport.vbs

Prnqctl.vbs

Command-line reference A-Z

Using the command-based script host (CScript.exe)

WMI overview

WMI overview

Top of page

Manage Your Profile |Legal |Contact Us

Prnport.vbs

Creates, deletes, and lists standard TCP/IP printer ports, in addition to displaying and changing port configuration. Used without parameters, **prnport.vbs** displays help for the **prnport.vbs** command.

To create a standard TCP/IP printer port

Syntax

cscript prnport.vbs -a -r *PortName* [**-s** *RemoteComputer*] **-h** *IPAddress* [**-u** *UserName* **-w** *Password*] [**-o** {**raw -n** *PortNumber* | **lpr**}] [**-q** *QueueName*] [**-m**{**e** | **d**}] [**-i** *IndexName*] [**-y** *CommunityName*] [**-2**{**e** | **d**}]

Top of page

Parameters

-a : Required. Specifies that you want to create a standard TCP/IP printer port.

-r *PortName* : Required. Specifies the port to which the printer is connected.

-s *RemoteComputer* : Specifies, by name, the remote computer to which you want to add the port. If you do not specify a computer, the port is added to the local computer.

-h *IPAddress* : Required. Specifies the IP address you want to assign to the port.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer on which you want to create a standard TCP/IP printer port. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

-o {raw -n *PortNumber* | lpr} : Specifies which protocol the port uses: TCP raw or TCP LPR. If you use TCP raw, specifies the port number for a TCP raw printer port. By default, this is port number 9100. For more information, see Related Topics. Most printers use TCP raw. On UNIX networks, printers often use TCP LPR. For more information about TCP raw, see Related Topics. For more information about TCP LPR, see RFC 1179 on the RFC Editor Web site.

-q *QueueName* : Specifies the queue name for a TCP raw port.

-m{e | d} : Specifies whether SNMP is enabled. The parameter **e** enables SNMP. The parameter **d** disables SNMP.

-i *IndexName* : Specifies the SNMP index, if SNMP is enabled. For more information, see RFC 1759 at the RFC Editor Web site.

-y *CommunityName* : Specifies the SNMP community name, if SNMP is enabled. For more information, see Related Topics.

-2{e | d} : Specifies whether double spools (also known as respooling) are enabled for TCP LPR ports. Double spools are necessary because TCP LPR must include an accurate byte count in the control file that is sent to the printer, but the protocol cannot get the count from the local print provider. Therefore, when a file is spooled to a TCP LPR print queue, it is also spooled as a temporary file in the system32 directory. TCP LPR determines the size of the temporary file and sends the size to the server running LPD. The parameter **e** enables double spools. The parameter **d** disables double spools.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If you want to change the configuration for a standard TCP/IP printer port after you create it, you can use the **cscript prnport.vbs** command with the **-t** parameter.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").
- Web addresses can change, so you might be unable to connect to the Web site or sites mentioned here.

To delete a standard TCP/IP printer port

⇡Top of page

Syntax

cscript prnport.vbs -d -r *PortName* [**-s** *RemoteComputer*] [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-d : Required. Specifies that you want to delete a standard TCP/IP printer port.

-r *PortName* : Required. Specifies the standard TCP/IP printer port that you want to delete.

-s *RemoteComputer* : Specifies, by name, the remote computer from which to delete the port. If you do not specify a computer, the port is deleted from the local computer.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer on which you want to delete a standard TCP/IP printer port. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

⇡Top of page

Example

To delete the standard TCP/IP printer port named IP_192.168.12.128 from the remote computer named HRServer, type:

cscript prnport.vbs -d -r IP_192.168.12.128 -s HRServer

To list all of the standard TCP/IP printer ports on a computer

⇡Top of page

Syntax

cscript prnport.vbs -l [**-s** *RemoteComputer*] [**-u** *UserName* **-w** *Password*]

⇡Top of page

Parameters

-l : Required. Specifies that you want to list all standard TCP/IP printer ports on a computer.

-s *RemoteComputer* : Specifies, by name, the remote computer for which you want to list ports. If you do not specify a computer, the ports on the local computer are listed.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer for which you want to list all standard TCP/IP printer ports. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

Top of page

Example

To list all standard TCP/IP printer ports on the remote computer named HRServer, type:

cscript prnport.vbs -l -s HRServer

To display the configuration of a standard TCP/IP printer port

Top of page

Syntax

cscript prnport.vbs -g -r *PortName* [**-s** *RemoteComputer*] [**-u** *UserName* **-w** *Password*]

Top of page

Parameters

-g : Required. Specifies that you want to display the configuration of a standard TCP/IP printer port.

-r *PortName* : Required. Specifies the port whose configuration you want to display.

-s *RemoteComputer* : Specifies, by name, the remote computer that hosts the port whose configuration you want to display. If you do not specify a computer, information is displayed for the port as it is configured on the local computer.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer that hosts the port whose configuration you want to display. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To configure a standard TCP/IP printer port

Top of page

Syntax

cscript prnport.vbs -t -r *PortName* [**-s** *RemoteComputer*] [**-o** {**raw -n** *PortNumber* | **lpr**}] [**-h** *IPAddress*] [**-q** *QueueName*] [**-m**{**e** | **d**}] [**-i** *IndexName*] [**-y** *CommunityName*] [**-2**{**e** | **d**}] [**-u** *UserName* **-w** *Password*]

Top of page

Parameters

-t : Required. Specifies that you want to configure a standard TCP/IP printer port.

-r *PortName* : Required. Specifies the port to which the printer is connected.

-s *RemoteComputer* : Specifies, by name, the remote computer on which you want to configure the port. If you do not specify a computer, the port is configured on the local computer.

-o {raw -n *PortNumber* | lpr} : Specifies which protocol the port uses: TCP raw or TCP LPR. If you use TCP raw, specifies the port number for a TCP raw printer port. By default, this is port number 9100. For more information, see Related Topics. Most printers use TCP raw. On UNIX networks, printer ports often use TCP LPR. For more information about TCP raw, see Related Topics. For more information about TCP LPR, see RFC 1179 at the RFC Editor Web site.

-h *IPAddress* : Specifies, by IP address, the printer for which you want to configure the port.

-q *QueueName* : Specifies the queue name for a TCP raw port.

-m{e | d} : Specifies whether SNMP is enabled. The parameter **e** enables SNMP. The parameter **d** disables SNMP.

-i *IndexName* : Specifies the SNMP index, if SNMP is enabled. For more information, see RFC 1759 at the RFC Editor Web site.

-y *CommunityName* : Specifies the SNMP community name, if SNMP is enabled. For more information, see Related Topics.

-2{e | d} : Specifies whether double spools (also known as respooling) are enabled for TCP LPR ports. Double spools are necessary because TCP LPR must include an accurate byte count in the control file that is sent to the printer, but the protocol cannot get the count from the local print provider. Therefore, when a file is spooled to a TCP LPR print queue, it is also spooled as a temporary file in the system32 directory. TCP LPR determines the size of the temporary file and sends the size to the server running LPD. The parameter **e** enables double spools. The parameter **d** disables double spools.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer on which you want to configure a port. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").
- Web addresses can change, so you might be unable to connect to the Web site or sites mentioned here.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Prncnfg.vbs

Prnjobs.vbs

Prnmngr.vbs

Prndrvr.vbs

Prnqctl.vbs

Command-line reference A-Z

Using the command-based script host (CScript.exe)

Defining communities

WMI overview

WMI overview

Top of page

Prnqctl.vbs

Prints a test page, pauses or resumes a printer, and clears a printer queue. Used without parameters, **prnqctl.vbs** displays command-line help for the **prnqctl.vbs** command.

To pause printing

Syntax

```
cscript prnqctl.vbs -z [-s RemoteComputer] -p PrinterName [-u UserName -w Password]
```

⇧Top of page

Parameters

-z : Required. Specifies that you want to pause printing.

-s *RemoteComputer* : Specifies, by name, the remote computer to which the printer you want to pause is attached. If you do not specify a computer, the printer attached to the local computer is paused.

-p *PrinterName* : Required. Specifies, by name, the printer you want to pause.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer that hosts the printer on which you want to pause printing. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

⇧Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, **"***Computer Name***"**).

Top of page

Example

To pause all printing on the printer named ColorPrinter_2 hosted on the remote computer named HRServer, type:

cscript prnqctl.vbs -z -s HRServer -p ColorPrinter_2

To resume printing

Top of page

Syntax

cscript prnqctl.vbs -m [**-s** *RemoteComputer*] **-p** *PrinterName* [**-u** *UserName* **-w** *Password*]

Top of page

Parameters

-m : Required. Specifies that you want to resume printing.

-s *RemoteComputer* : Specifies, by name, the remote computer to which the printer you want to resume is attached. If you do not specify a computer, the printer attached to the local computer is resumed.

-p *PrinterName* : Required. Specifies, by name, the printer on which you want to resume printing.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer that manages the printer on which you want to resume printing. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

Top of page

Example

To resume all printing on the printer named ColorPrinter_2 hosted on the remote computer named HRServer, type:

cscript prnqctl.vbs -m -s HRServer -p ColorPrinter_2

To print a test page

Top of page

Syntax

cscript prnqctl.vbs -e [**-s** *RemoteComputer*] **-p** *PrinterName* [**-u** *UserName* **-w** *Password*]

Top of page

Parameters

-e : Required. Specifies that you want to print a test page.

-s *RemoteComputer* : Specifies, by name, the remote computer to which the printer on which you want to print a test page is attached. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Required. Specifies, by name, the printer on which you want to print a test page.

-u *UserName* -w *Password* : Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer that manages the printer on which you want to print a test page. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

To cancel all jobs spooled to a printer

Top of page

Syntax

cscript prnqctl.vbs -x [**-s** *RemoteComputer*] **-p** *PrinterName* [**-u** *UserName* **-w** *Password*]

Top of page

Parameters

-x : Required. Specifies that you want to cancel all jobs spooled to a printer.

-s *RemoteComputer* : Specifies, by name, the remote computer to which the printer for which you want to cancel all jobs is attached. If you do not specify a computer, the local computer is used.

-p *PrinterName* : Required. Specifies, by name, the printer for which you want to cancel all print jobs.

-u ***UserName*** **-w** ***Password*** **:** Specifies an account with permissions to connect by using Windows Management Instrumentation (WMI) services to the computer on which you want to cancel all print jobs. All members of the Administrators group for that computer have these permissions, but the permissions can also be granted to other users. If you do not specify an account, you must be logged on under an account with these permissions for the command to work. For more information about WMI, see Related Topics.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- This command starts a script that is located in the *systemroot*\system32 directory. You must type this command at a command prompt with that directory as the current directory, or you must type the full path to that directory at the beginning of the **cscript** command.
- If the information that you supply contains spaces, use quotation marks around the text (for example, "*Computer Name*").

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Prncnfg.vbs

Prnjobs.vbs

Prnmngr.vbs

Prnport.vbs

Prndrvr.vbs

Command-line reference A-Z

Using the command-based script host (CScript.exe)

WMI overview

WMI overview

Top of page

Manage Your Profile |Legal |Contact Us

Prompt

Changes the Cmd.exe prompt. Used without parameters, **prompt** resets the command prompt to the default setting, the current drive letter followed by the current directory and a greater-than symbol (>).

Syntax

prompt [*text*]

⇡Top of page

Parameters

***text* :** Specifies any text and information you want included in your system prompt.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- 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 XP version number.

- The following table lists the character combinations you can include instead of, or in addition to, one or more character strings in the *text* command-line option. The list includes a brief description of the text or information that each character combination adds to your command prompt.

Character	Description
$q	= (equal sign)
$$	$ (dollar sign)
$t	Current time
$d	Current date
$p	Current drive and path
$v	Windows XP 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

•When command extensions are enabled (that is, the default) the **prompt** command supports the formatting characters listed in the following table.

Character	Description
$+	Zero or more plus sign (+) characters depending upon the depth of the **pushd** directory stack, one character for each level pushed.
$m	The remote name associated with the current drive letter or the empty string if current drive is not a network drive.

•If you include the **$p** character in the text parameter, your disk is read, after you enter each command, to determine the current drive and path. This can take extra time, especially for floppy disk drives.

⇡Top of page

Examples

The following example sets the command prompt to display the current drive and path followed by the greater-than symbol (>):

prompt pg

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

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Command-line reference A-Z

↑Top of page

Manage Your Profile |Legal |Contact Us

Pushd

Stores the name of the current directory for use by the **popd** command before changing the current directory to the specified directory.

Syntax

pushd [*Path*]

⇡Top of page

Parameters

Path **:** Specifies the directory to which the current directory should be changed. This command supports relative paths.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Every time you use the **pushd** command, a single directory is stored for your use. However, you can store multiple directories by using the **pushd** command multiple times.

  The directories are stored sequentially in a virtual stack. If you use the **pushd** command once, the directory in which you use the command is placed at the bottom of the stack. If you use the command again, the second directory is placed on top of the first one. The process repeats every time you use the **pushd** command.

  You can use the **popd** command to change the current directory to the directory most recently stored by the **pushd** command. If you use the **popd** command, the directory on the top of the stack is removed from the stack as the current directory is changed to that directory. If you use the **popd** command again, the next directory on the stack is removed.
- If command extensions are enabled, the **pushd** command accepts either a network path or a local drive letter and path.

- If you specify a network path, the **pushd** command temporarily assigns the first unused drive letter (starting with Z:) to the specified network resource. The command then changes the current drive and directory to the specified directory on the newly assigned drive. If you use the **popd** command with command extensions enabled, the **popd** command removes the drive-letter assignation created by **pushd**.

Top of page

Examples

You can use the **pushd** command and the **popd** command in a batch program to change the current directory from the one in which the batch program was run and then change it back. The following sample batch program shows how to do this:

```
@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
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Popd

Cmd

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Terminal Services query commands

For more information about the Terminal Services query commands, see the following topics:

•Query process
•Query session
•Query termserver
•Query user

Top of page

Manage Your Profile |Legal |Contact Us

Query process

Displays information about processes running on a terminal server. You can use this command to find out which programs a specific user is running, and also which users are running a specific program.

Syntax

query process [{ *|*ProcessID*|*UserName*|*SessionName*|**/id:***nn*|*ProgramName*}] [**/server:** *ServerName*] [**/system**]

⇡Top of page

Parameters

*** :** Lists the processes for all sessions.

***ProcessID* :** Specifies the numeric ID identifying the process you want to query.

***UserName* :** Specifies the name of the user whose processes you want to list.

***SessionName* :** Specifies the name of the session whose processes you want to list.

/id:*nn* : Specifies the ID of the session whose processes you want to list.

***ProgramName* :** Specifies the name of the program whose processes you want to query. The .exe extension is required.

/server:*ServerName* : Specifies the terminal server whose processes you want to list. If unspecified, the server where you are currently logged on is used.

/system : Specifies that current information about system processes is displayed. If you do not want system processes to be displayed, do not use this parameter.

/? : Displays help at the command prompt.

Top of page

Remarks

- Administrators have full access to all **query process** functions.
- If you do not specify the *UserName*, *SessionName*, **/id:***SessionID*, *ProgramName*, or * parameters, **query process** displays only the processes belonging to the current user.
- If a session is specified, it must identify an active session. You can use wildcards to identify the process.
- **Query process** returns the following information:

  - The user who owns the process
  - The session that owns the process
  - The ID of the session
  - The name of the process
  - The state of the process
  - The ID of the process
- When **query process** returns information, a less than (>) symbol is displayed before each process belonging to the current session.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Query session

Displays information about sessions on a terminal server. The list includes information not only about active sessions but about other sessions that the server runs.

Syntax

query session [{*SessionName*|*UserName*|*SessionID*}] [**/server:***ServerName*] [**/mode**] [**/flow**] [**/connect**] [**/counter**]

⇡Top of page

Parameters

***SessionName* :** The name of the session you want to query.

***UserName* :** The name of the user whose sessions you want to query.

***SessionID* :** The ID of the session you want to query.

/server:*ServerName* : Identifies the terminal server to query. The default is the current server.

/mode : Displays current line settings.

/flow : Displays current flow-control settings.

/connect : Displays current connect settings.

/counter : Displays current counters information, including the total number of sessions created, disconnected, and reconnected.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- A user can always query the session to which the user is currently logged on. To query other sessions, the user must have Query Information access permission.
- If you do not specify a session using *SessionName*, *UserName*, or *SessionID*, **query session** displays information about all active sessions in the system.
- When **query session** returns information, a less than (>) symbol is displayed before the current session.

⇡Top of page

Examples

To display information about all active sessions on server SERVER2, type:

query session /server:SERVER2

To display information about active session MODEM02, type:

query session MODEM02

Sample output for query session is as follows:

```
C:\>query session
 SESSIONNAME    USERNAME       ID STATE  TYPE   DEVICE
>console        administrator   0 active wdcon
 rdp-tcp#1      client1         1 active wdtshare
 rdp-tcp                        2 listen wdtshare
                                4 idle
                                5 idle
```

The less than (>) symbol indicates the current session. **SESSIONNAME** specifies the name assigned to the session. **USERNAME** indicates the user name of the user connected to the session. **STATE** provides information about the current state of the session. **TYPE** indicates the session type. **DEVICE**, which is not present for the console or network-connected sessions, is the device name assigned to the session. The comment following session information is from the session profile.

Any sessions in which the initial state is configured as DISABLED do not show up in the query session list until they are enabled.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Terminal Services commands

Top of page

Manage Your Profile |Legal |Contact Us

Query termserver

Displays a list of all terminal servers on the network.

Syntax

query termserver [*ServerName*] [**/domain:***Domain*] [**/address**] [**/continue**]

⇡Top of page

Parameters

***ServerName* :** Specifies the name identifying the terminal server.

/domain:*Domain* : Specifies the domain to query for terminal servers. You do not need to specify a domain if you are querying the domain in which you are currently working.

/address : Displays the network and node addresses for each server.

/continue : Prevents pausing after each screen of information is displayed.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- **Query termserver** searches the network for all attached terminal servers and returns the following information:

  - The name of the server.
  - The network (and node address if the **/address** option is used).

⇡Top of page

Examples

To display information about all terminal servers on the network, type:

query termserver

To display information about the terminal server named *server3*, type:

query termserver server3

To display information about all terminal servers in domain SYSTEM, type:

query termserver /domain:system

To display the network and node address for the terminal server named *server3*, type:

query termserver server3 /address

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Terminal Services commands

Query user

Displays information about user sessions on a terminal server.

Syntax

query user [{*UserName*|*SessionName*|*SessionID*}] [**/server:***ServerName*]

↑Top of page

Parameters

***UserName* :** Specifies the logon name of the user you want to query.

***SessionName* :** Specifies the name of the session you want to query.

***SessionID* :** Specifies the ID of the session you want to query.

/server:*ServerName* : Specifies the terminal server you want to query. Otherwise, the current terminal server is used.

/? : Displays help at the command prompt.

↑Top of page

Remarks

- You can use this command to find out if a specific user is logged on to a specific terminal server. **Query user** returns the following information:

  - The name of the user
  - The name of the session on the terminal server
  - The session ID
  - The state of the session (active or disconnected)
  - The idle time (the number of minutes since the last keystroke or mouse movement at the session)
  - The date and time the user logged on
- To use **query user**, you must have Full Control permission or Query Information special access permission.
- If you use **query user** without specifying a user name, session name, or session ID, a list of all users who are logged on to the server is returned. Alternatively, you can also use **query session** to display a list of all sessions on a server.
- When **query user** returns information, a less than (>) symbol is displayed before the current session.
- The **/server** parameter is required only if you use **query user** from a remote server.

↑Top of page

Examples

To display information about all users logged on the system, type:

query user

To display information about the user USER1 on server SERVER1, type:

query user USER1 /server:SERVER1

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Query session

Terminal Services commands

Top of page

Rasdial

You can automate the connection process for any Microsoft client by using a simple batch file and the **rasdial** command. The **rasdial** command starts a network connection by using a specified entry.

Syntax

rasdial *connectionname* [*username* [*password* | *****]] [**/domain:***domain*] [**/phone:***phonenumber*] [**/callback:***callbacknumber*] [**/phonebook:***phonebookpath*] [**/prefixsuffix**]

The **rasdial** command disconnects a network connection by using the following syntax:

rasdial [*connectionname*] **/disconnect**

⇡Top of page

Parameters

None : Used without options, **rasdial** displays the status of current connections.

***connectionname* :** Specifies an entry in the current .pbk file, located in the *systemroot*\System32\Ras folder. If the connection name contains spaces or special characters, enclose the connection name in quotation marks (").

The Rasphone.pbk file is used unless the **Personal Phonebook** option is selected. If the **Personal Phonebook** option is selected, the file *username*.pbk is used. The name is shown on the Rasphone title bar when **Personal Phonebook/p** is selected. Numbers are appended if name conflicts occur.

***username* [*password* | *] :** Specifies a user name and password with which to connect. If an asterisk is used, the user is prompted for the password, but does not display the characters typed.

/domain:*domain* : Specifies the domain the user account is located in.

If not present, the last value of the **Domain** field in the **Connect To** dialog box is used.

/phone:*phonenumber* : Substitutes the specified phone number for the entry's phone number in Rasphone.pbk.

/callback:*callbacknumber* : Substitutes the specified callback number for the entry's callback number in Rasphone.pbk.

/disconnect : Disconnects the specified entry.

/phonebook:*phonebookpath* : Specifies the path to the phonebook file. The default is *systemroot* \System32\Ras\*username*.pbk. You can specify a full path to the file.

/prefixsuffix : Applies the current TAPI location dialing settings to the phone number. These settings are configured in Telephony, which is located in Control Panel. This option is off by default.

⇡Top of page

Rasdial examples

To connect to the OFFICE entry in Rasphone.pbk, type:

rasdial office

To connect to the OFFICE 2 entry in Rasphone.pbk and specify a callback number, type:

rasdial "office 2" /callback:555-0100

To disconnect from the EAST OFFICE entry, type:

rasdial "EAST OFFICE" /d

Note

The following Rasphone.exe features are not supported with **rasdial**:

- Entries that require Terminal mode user entry during the dial sequence.
- Operator-assisted or manual dialing.

Rcp

Copies files between a Windows XP computer and a system running **rshd**, the remote shell service (daemon). Windows XP and Windows 2000 do not provide rshd service. Used without parameters, **rcp** displays help.

Syntax

rcp [{**-a** | **-b**}] [**-h**] [**-r**] [*Host*][**.***User***:**] [*Source*] [*Host*][**.***User***:**] [*Path\Destination*]

⇡Top of page

Parameters

-a : Specifies ASCII transfer mode. This mode converts the end-of-line (EOL) characters to a carriage return for UNIX and a carriage return/line feed for computers. This is the default transfer mode.

-b : Specifies binary image transfer mode. No carriage return/line feed conversion occurs.

-h : Transfers source files that are marked with the hidden attribute to the Windows XP computer. Otherwise, hidden files are not copied.

-r : Recursively copies to the destination the contents of all subdirectories of the source.

***Host* :** Specifies the local or remote host. If *Host* is specified as an IP address or if the host name contains dots (.), you must specify the user.

***User* :** Specifies the user name. If the user name is not specified, the name of the user who is currently logged on is used.

***Source* :** Specifies the files to copy.

***Path\Destination* :** Specifies the path relative to the logon directory on the remote host. Use the backslash (\), quotation mark ("), or apostrophe () escape characters in remote paths to use wildcard characters on the remote host. If multiple source files are specified, the destination is a directory.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Third-party transfers

  The **rcp** command, a connectivity command, can also be used for third-party transfers. You can run the **rcp** command from a Windows XP computer to copy files between two other computers that are running **rshd**. The **rshd** daemon is available on UNIX computers, so the Windows XP computer can participate in a third-party transfer only as the system from which the commands are run.
- Using the **-r** parameter

  Both the *Source* and *Path\Destination* must be directories. However, you can use **-r** without recursion if the source is not a directory.
- Using the *Source* and *Path\Destination* parameters

  If the file name does not begin with a forward slash (/) for UNIX or a backslash (\) for Windows XP, it is assumed to be relative to the current working directory. On Windows XP, this is the directory from which the command is run. On the remote system, it is the logon directory for the remote user. A period (.) indicates the current directory. You can use the backslash (\), quotation mark ("), or apostrophe () escape characters in remote paths as wildcard characters on the remote computer.
- 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 with **rcp**.

- The .rhosts file

  The .rhosts file specifies which remote systems 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. Both the **rsh** and **rcp** commands transmit the local user name to the remote system. The remote system uses this name and the IP address (usually resolved to a computer name) of 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 domain, the primary domain controller must be available to resolve the logon name because it is not cached on the local computer. Because the user name is required as part of the **rsh** protocol, the command fails if it cannot be obtained.

  The .rhosts file is a text file in which 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 pound sign (#), for example:

  ```
  computer5marie #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 the specific implementation of the .rhosts file on a remote computer, see the remote system documentation.

  Additionally, you can add your computer name to the /Etc/Hosts file on the remote computer. This allows the remote system to authenticate remote requests for your computer when you use the Windows XP TCP/IP utilities.
- Specifying computers (hosts)

  Use the *Computer***.***User* parameters to use a user name other than the current one. If *Computer***.** *User* is specified with *Source*, the .rhosts file on the remote computer must contain an entry for *User*, as follows:

  ```
  rcp host99.user7:file1 corp7.admin:file2
  ```

  The .rhosts file on corp7 should have an entry for user7 on host99.

  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 part of the domain name from being interpreted as a user name, as follows:

  ```
  rcp domain-name1.user:user92 domain-name2.user:user7
  ```

- Remote processing

  Remote processing is performed by a command that is run from the user logon shell on most UNIX systems. The .profile or .cshrc of the user is executed before parsing file names, and exported shell variables can be used (using the escape character or quotation marks) in remote file names.
- Copying Files

  If you attempt to copy a number of files to a file rather than a directory, only the last file is copied. The **rcp** command cannot copy a file onto itself (the Source and Path/Destination cannot be the same.)

  If you have logged onto the Windows XP Professional computer using a domain other than the local one, and the primary domain controller is unavailable, the command fails because **rcp** cannot determine the local user name. The same restriction applies to **rsh**.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To copy a local file to the logon directory of a remote computer, type:

```
rcp filename remotecomputer:
```

To copy a local file to an existing directory with a new file name on a remote computer, type:

```
rcp filename remotecomputer:/directory/newfilename
```

To copy multiple local files to the subdirectory of a remote logon directory, type:

```
rcp file1 file2 file3 remotecomputer:subdirectory/filesdirectory
```

To copy from a remote source to the current directory of the local computer, type:

```
rcp remotecomputer:filename
```

To copy multiple files from multiple remote sources to a remote destination with different user names, type:

```
rcp remote1.user1:file1 remote2.user2:file2 remotedest.destuser:directory
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Recover

Recovers readable information from a bad or defective disk.

Syntax

recover [*Drive*:][*Path*] *FileName*

Top of page

Parameters

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the file you want to recover.

/? : Displays help at the command prompt.

Top of page

Remarks

- The **recover** command reads a file sector by sector and recovers data from the good sectors. Data in bad sectors is lost.
- Limitation on [drive:][path]filename

  You cannot use wildcards (* and ?) with the **recover** command. You must specify a file.
- Reentering lost data

  Because all data in bad sectors is lost when you recover a file, you should recover files one at a time. You can use this method 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** does not affect them.

Top of page

Examples

To recover the file Story.txt in the \Fiction directory in drive D:, type:

recover d:\fiction\story.txt

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Chkdsk

Chkdsk

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Using command redirection operators

You can use redirection operators to redirect command input and output streams from the default locations to different locations. The input or output stream location is referred to as a handle

The following table lists operators that you can use to redirect command input and output streams.

Redirection operator	Description
>	Writes the command output to a file or a device, such as a printer, instead of the Command Prompt window.
<	Reads the command input from a file, instead of reading input from the keyboard.
>>	Appends the command output to the end of a file without deleting the information that is already in the file.
>&	Writes the output from one handle to the input of another handle.
<&	Reads the input from one handle and writes it to the output of another handle.
|	Reads the output from one command and writes it to the input of another command. Also known as a pipe.

By default, you send the command input (that is, the STDIN handle) from your keyboard to Cmd.exe, and then Cmd.exe sends the command output (that is, the STDOUT handle) to the Command Prompt window.

The following table lists the available handles.

Handle	Numeric equivalent of handle	Description
STDIN	0	Keyboard input
STDOUT	1	Output to the Command Prompt window
STDERR	2	Error output to the Command Prompt window

UNDEFINED	3-9	These handles are defined individually by the application and are specific to each tool.

The numbers zero through nine (that is, 0-9) represent the first 10 handles. You can use Cmd.exe to run a program and redirect any of the first 10 handles for the program. To specify which handle you want to use, type the number of the handle before the redirection operator. If you do not define a handle, the default < redirection input operator is zero (0) and the default > redirection output operator is one (1). After you type the < or > operator, you must specify where you want to read or write the data. You can specify a file name or another existing handle.

To specify redirection to existing handles, use the ampersand (&) character followed by the handle number that you want to redirect (that is, **&***handle#*). For example, the following command redirects handle 2 (that is, STDERR) into handle 1 (that is, STDOUT):

```
1<&2
```

Duplicating handles

The & redirection operator duplicates output or input from one specified handle to another specified handle. For example, to send **dir** output to File.txt and send the error output to File.txt, type:

```
dir>c:\file.txt 2>&1
```

When you duplicate a handle, you duplicate all characteristics of the original occurrence of the handle. For example, if a handle has write-only access, all duplicates of that handle have write-only access. You cannot duplicate a handle with read-only access into a handle with write-only access.

⇡Top of page

Redirecting command input (<)

To redirect command input from the keyboard to a file or device, use the < operator. For example, to get the command input for the **sort** command from File.txt:

```
sort<file.txt
```

The contents of File.txt appear in the Command Prompt window as an alphabetized list.

The < operator opens the specified file name with read-only access. As a result, you cannot write to the file when you use this operator. For example, if you start a program with <&2, all attempts to read handle 0 fail because handle 2 is initially opened with write-only access.

Note

- Zero is the default handle for the < redirection input operator.

⇡Top of page

Redirecting command output (>)

Almost all commands send output to your Command Prompt window. Even commands that send output to a drive or printer display messages and prompts in the Command Prompt window.

To redirect command output from the Command Prompt window to a file or device, use the > operator. You can use this operator with most commands. For example, to redirect **dir** output to Dirlist.txt:

dir>dirlist.txt

If Dirlist.txt does not exist, Cmd.exe creates it. If Dirlist.txt exists, Cmd.exe replaces the information in the file with the output from the **dir** command.

To run the **netsh routing dump** command and then send the command output to Route.cfg, type:

netsh routing dump>c:\route.cfg

The > operator opens the specified file with write-only access. As a result, you cannot read the file when you use this operator. For example, if you start a program with redirection >&0, all attempts to write handle 1 fail because handle 0 is initially opened with read-only access.

Note

- One is the default handle for the > redirection output operator.

⇡Top of page

Using the <& operator to redirect input and duplicate

To use the redirection input operator <&, the file you specify must already exist. If the input file exists, Cmd.exe opens it as read-only and sends the characters contained in the file as input to the command as if they were input from the keyboard. If you specify a handle, Cmd.exe duplicates the handle you specify onto the existing handle in the system.

For example, to open File.txt as input read to handle 0 (that is, STDIN), type:

<file.txt

To open File.txt, sort the contents and then send the output to the Command Prompt window (that is, STDOUT), type:

sort<file.txt

To find File.txt, and then redirect handle 1 (that is, STDOUT) and handle 2 (that is, STDERR) to the Search.txt, type:

findfile file.txt>search.txt 2<&1

To duplicate a user-defined handle 3 as input read to handle 0 (that is, STDIN), type:

<&3

⇡Top of page

Using the >& operator to redirect output and duplicate

If you redirect output to a file and you specify an existing file name, Cmd.exe opens the file as write-only and overwrites the file's contents. If you specify a handle, Cmd.exe duplicates the file onto the existing handle.

To duplicate a user-defined handle 3 into handle 1, type:

>&3

To redirect all of the output, including handle 2 (that is, STDERR), from the **ipconfig** command to handle 1 (that is, STDOUT), and then redirect the ouput to Output.log, type:

ipconfig.exe>>output.log 2>&1

Top of page

Using the >> redirection operator to append output

To add the output from a command to the end of a file without losing any of the information already in the file, use two consecutive greater than signs (that is, >>). For example, the following command appends the directory list produced by the **dir** command to the Dirlist.txt file:

dir>>dirlist.txt

To append the output of the **netstat** command to the end of Tcpinfo.txt, type:

netstat>>tcpinfo.txt

Top of page

Using the pipe operator (|)

The pipe operator (|) takes the output (by default, STDOUT) of one command and directs it into the input (by default, STDIN) of another command. For example, the following command sorts a directory:

dir | sort

In this example, both commands start simultaneously, but then the **sort** command pauses until it receives the **dir** command's output. The **sort** command uses the **dir** command's output as its input, and then sends its output to handle 1 (that is, STDOUT).

Top of page

Combining commands with redirection operators

You can create custom commands by combining filter commands with other commands and file names. For example, you can use the following command to store the names of files that contain the string "LOG":

dir /b | find "LOG" > loglist.txt

The **dir** command's output is sent through the **find** filter command. File names that contain the string "LOG" are stored as a list of file names (for example, NetshConfig.log, Logdat.svd, and Mylog.bat) in the Loglist.txt file.

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 file names that include the string "Log", and then displays them in one Command Prompt window at a time:

dir c:\ /s /b | find "LOG" | more

By using a pipe (|), you direct Cmd.exe to send the **dir** command output through the **find** filter command. The **find** command selects only file names that contain the string "LOG." The **more** command displays the file names that are selected by the **find** command, one Command Prompt window at a time. For more information about filter commands, see Using filters

Top of page

Manage Your Profile |Legal |Contact Us

Reg

Adds, changes, and displays registry subkey information and values in registry entries.

To view the command syntax, click a command:

reg add

Adds a new subkey or entry to the registry.

Syntax

reg add *KeyName* [**/v** *EntryName*|**/ve**] [**/t** *DataType*] [**/s** *separator*] [**/d** *value*] [**/f**]

Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName**PathToSubkey* format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**.

/v *EntryName* : Specifies the name of the entry to be added under the specified subkey.

/ve : Specifies that the entry that is added to the registry has a null value.

/t *DataType* : Specifies the data type for the value of the entry. *DataType* can be one of the following:

REG_SZ

REG_MULTI_SZ

REG_DWORD_BIG_ENDIAN

REG_DWORD

REG_BINARY

REG_DWORD_LITTLE_ENDIAN

REG_LINK

REG_FULL_RESOURCE_DESCRIPTOR

REG_EXPAND_SZ

/s *separator* : Specifies the character used to separate multiple instances of data. Used when REG_MULTI_SZ is specified as the data type and more than one entry needs to be listed. If not specified, the default separator is \0.

/d *value* : Specifies the value for the new registry entry.

/f : Adds the subkey or entry without asking for confirmation.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Subtrees cannot be added with this operation. This version of Reg does not ask for confirmation when adding a subkey.
- The following table lists the return values for the **reg add** operation.

Value	Description
0	Success
1	Failure

⇡Top of page

Examples

The following examples show how you can use the **reg add** command:

reg add \hklm\software\myco /v data /t reg_binary /d fe340ead
reg add "hkcu\software\microsoft\winmine" /v Name3 /t reg_sz /d Anonymous
reg add "hkcu\software\microsoft\winmine" /v Time3 /t reg_dword /d 5

reg compare

Compares specified registry subkeys or entries.

⇡Top of page

Syntax

reg compare *KeyName1 KeyName2* [**/v** *EntryName* | **/ve**] {[**/oa**]|[**/od**]|[**/os**]|[**on**]} [**/s**]

⇡Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName*\PathtoSubkey format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**. If a remote computer is specified, you can use the **HKLM** and **HKU** subtrees only.

/v *EntryName* : Compares a specific entry under the subkey.

/ve : Specifies that only entries that have no value will be compared.

{[/oa]|[/od]|[/os]|[on]} : Specifies how differences and matches are displayed. The default is **/od**.

Value	Description
/oa	Specifies that all differences and matches are displayed. By default, only the differences are listed.
/od	Specifies that only differences are displayed. This is the default behavior.
/os	Specifies that only matches are displayed. By default, only the differences will be listed.
/on	Specifies that nothing is displayed. By default, only the differences will be listed.

/s ***Separator*** **:** Compares all subkeys and entries.

/?: Displays help at the command prompt.

Top of page

Remarks

- The following table lists the return values for the **reg compare** operation.

Value	Description
0	The comparison is successful and the result is identical.
1	The comparison failed.
2	The comparison is successful and differences were found.

Top of page

Examples

The following example shows how you can use the **reg compare** command:

reg compare "hkcu\software\microsoft\winmine" "hkcu\software\microsoft\winmine" /od /s

reg copy

Copies a registry entry to a specified location in the local or remote computer.

Top of page

Syntax

reg copy *KeyName1 KeyName2* [**/s**] [**/f**]

Top of page

Parameters

***KeyName1* :** Specifies the full path of the subkey to copy. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName*\PathToSubkey format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**. If a remote computer is specified, you can use the **HKLM** and **HKU** subtrees only.

***KeyName2* :** Specifies the full path of the subkey destination. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName*\PathToSubkey format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**. If a remote computer is specified, you can use the **HKLM** and **HKU** subtrees only.

/s : Copies all subkeys and entries under the specified subkey.

/f : Copies the subkey without asking for confirmation.

/?: Displays help at the command prompt.

Top of page

Remarks

- This version of Reg does not ask for confirmation when copying a subkey.
- The following table lists the return values for the **reg copy** operation.

Value	Description
0	Success
1	Failure

Top of page

Examples

The following examples show how you can use the **reg copy** command:

```
reg copy "hkcu\software\microsoft\winmine" "hkcu\software\microsoft\winminebk" /s /f
```

reg copy "hkcu\software\microsoft\winminebk" "hkcu\software\microsoft\winmine" /s

reg delete

Deletes a subkey or entries from the registry.

Top of page

Syntax

reg delete *KeyName* [{**/v** *EntryName*|**/ve**|**/va**}] [**/f**]

Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName**PathToSubkey* format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**.

/v *EntryName* : Deletes a specific entry under the subkey. If no entry is specified, then all entries and subkeys under the subkey will be deleted.

/ve : Specifies that only entries that have no value will be deleted.

/va : Deletes all entries under the specified subkey. Subkeys under the specified subkey are not deleted with this parameter.

/f : Deletes the existing registry subkey or entry without asking for confirmation.

/?: Displays help at the command prompt.

Top of page

Remarks

- The following table lists the return values for the **reg delete** operation.

Value	Description
0	Success
1	Failure

⇡Top of page

Examples

The following examples show how you can use the **reg delete** command:

reg delete "hkcu\software\microsoft\winmine" /v Name1
reg delete "hkcu\software\microsoft\winmine" /v Time1
reg delete "hkcu\software\microsoft\winmine" /va

reg export

Creates a copy of specified subkeys, entries, and values into a file so that it can be transferred to other servers.

⇡Top of page

Syntax

reg export *KeyName FileName*

⇡Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. The **export** operation works only with the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**.

***FileName* :** Specifies the name and path of the file to be exported. The file must have a .reg extension.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- The following table lists the return values for the **reg export** operation.

Value	Description
0	Success
1	Failure

⇡Top of page

Examples

The following example shows how you can use the **reg export** command:

reg export "hkcu\software\microsoft\winmine" c:\data\regbackups\wmbkup.reg

reg import

Copies a file containing exported registry subkeys, entries, and values into the local computer's registry.

⇡Top of page

Syntax

reg import *FileName*

⇡Top of page

Parameters

***FileName* :** Specifies he name and path of the file that will be copied into the registry of the local computer. This file needs to be created beforehand with the **reg export** operation.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

•The following table lists the return values for the **reg import** operation.

Value	Description
0	Success
1	Failure

⇡Top of page

Examples

The following example shows how you can use the **reg import** command:

reg import "hkcu\software\microsoft\winmine" c:\data\regbackups\wmbkup.reg

reg load

Writes saved subkeys and entries back to a different subkey in the registry. This is intended to be a temporary file that can be used for troubleshooting or editing registry entries.

⇡Top of page

Syntax

reg load *KeyName FileName*

⇡Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName**PathToSubkey* format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**.

***FileName* :** Specifies the name and path of the file that will be loaded. This file must have been created with the **reg save** operation using a .hiv extension.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

•The following table lists the return values for the **reg load** operation.

Value	Description
0	Success
1	Failure

⇡Top of page

Examples

The following example shows how you can use the **reg load** command:

reg load "hkcu\software\microsoft\winminebk2" wmbkup.hiv

reg query

Returns a list of the next tier of subkeys and entries located under a subkey in the registry.

⇡Top of page

Syntax

reg query *KeyName* [{**/v** *EntryName*|**/ve**}] [**/s**]

Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName**PathToSubkey* format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**. If a remote computer is specified, you can use the **HKLM** and **HKU** subtrees only.

/v *EntryName* : Returns a specific entry and its value. This parameter only returns entries that are in the tier directly below the specified subkey. Entries that are located in subkeys under the current subkey will not be found. When *EntryName* is omitted, all entries under the subkey are returned.

/ve : Specifies that only entries that have no value will be returned.

/s : Returns all subkeys and entries in all tiers. Without this parameter, only the next tier of subkeys and entries will be returned.

/?: Displays help at the command prompt.

Top of page

Remarks

- The following table lists the return values for the **reg query** operation.

Value	Description
0	Success
1	Failure

Top of page

Examples

The following examples show how you can use the **reg query** command:

reg query "hklm\system\currentcontrolset\control\session manager" /v maxstacktracedepth

reg query "hkcu\software\microsoft\winmine" /s

reg restore

Writes saved subkeys and entries back to the registry.

Top of page

Syntax

reg restore *KeyName FileName*

Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. The **restore** operation works only with the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**.

***FileName* :** Specifies the name and path of the file that will be written back to the registry. This file needs to be created beforehand with the **reg save** operation using a .hiv extension.

/?: Displays help at the command prompt.

Top of page

Remarks

- This operation is used to overwrite registry entries that have been edited. Before editing entries, save the parent subkey with the **reg save** operation. If the edit fails, restore the subkey with this operation.

•The following table lists the return values for the **reg restore** operation.

Value	Description
0	Success
1	Failure

⇡Top of page

Examples

The following example shows how you can use the **reg restore** command:

reg restore "hkcu\software\microsoft\winmine" wmbkup.hiv

reg save

Saves a copy of specified subkeys, entries, and values of the registry in a specified file.

⇡Top of page

Syntax

reg save *KeyName FileName*

⇡Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName**PathToSubkey* format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**.

***FileName* :** Specifies the name and path of the file that is created. If no path is specified, then the current path is used.

/?: Displays help at the command prompt.

Top of page

Remarks

- The following table lists the return values for the **reg save** operation.

Value	Description
0	Success
1	Failure

Top of page

Examples

The following example shows how you can use the **reg save** command:

reg save "hkcu\software\microsoft\winmine" wmbkup.hiv

reg unload

Removes a section of the registry that was loaded using the **reg load** operation.

Top of page

Syntax

reg unload *KeyName*

Top of page

Parameters

***KeyName* :** Specifies the full path of the subkey. For remote computers, include the computer name before the path of the subkey in the \\*ComputerName\PathToSubkey* format. Omitting *ComputerName* causes the operation to default to the local computer. Start the path with the appropriate subtree. The valid subtrees are **HKLM**, **HKCU**, **HKCR**, **HKU**, and **HKCC**.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- The following table lists the return values for the **reg unload** operation.

Value	Description
0	Success
1	Failure

⇡Top of page

Examples

The following example shows how you can use the **reg unload** command:

reg unload "hkcu\software\microsoft\winminebk2"

Caution

- Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

Caution

- Do not edit the registry directly unless you have no alternative. The registry editor bypasses standard safeguards, allowing settings that can degrade performance, damage your system, or even require you to reinstall Windows. You can safely alter most registry settings by using the programs in Control Panel or Microsoft Management Console (MMC). If you must edit the registry directly, back it up first. Read the Registry Editor Help for more information.

⇡Top of page

Remarks

- Using Reg directly edits the registry of local or remote computers. These changes can render the computers inoperable and cause the need for a new installation of the operating system. Instead of directly editing the registry, use Control Panel or Microsoft Management Console, whenever possible, to make changes to the registry.
- Some operations allow the viewing or configuration of registry entries on local or remote computers, while others allow only the configuration of local computers. Also, accessing the registry remotely might limit the parameters that you can use in an operation. Check the syntax for each operation to verify that it can be used on remote computers and to verify the parameters that can be used in that situation.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Regsvr32

This command-line tool registers .dll files as command components in the registry.

Syntax

regsvr32 [**/u**] [**/s**] [**/n**] [**/i**[:cmdline]] *dllname*

Top of page

Parameters

/u : Unregisters server.

/s : Specifies **regsvr32** to run silently and to not display any message boxes.

/n : Specifies not to call DllRegisterServer. You must use this option with **/i**.

/i:cmdline : Calls DllInstall passing it an optional [cmdline]. When used with **/u**, it calls dll uninstall.

***dllname* :** Specifies the name of the dll file that will be registered.

/? : Displays help at the command prompt.

Top of page

Examples

To register the .dll for the Active Directory Schema, type:

regsvr32 schmmgmt.dll

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Relog

Extracts performance counters from performance counter logs into other formats, such as **text-TSV** (for tab-delimited text), **text-CSV** (for comma-delimited text), **binary-BIN**, or **SQL**.

Syntax

relog [*FileName* [*filename* ...]] [**-a**] [**-c** *Path* [*path* ...]] [**-cf** *FileName*] [**-f** {**bin**|**csv**|**tsv**|**SQL**}] [**-t** *value*] [**-o** {*output file* | *DSN!counter_log*}] [**-b** *M/d/yyyy [[hh:]mm:]ss*] [**-e** *M/d/yyyy [[hh:]mm:]ss*] [**-config** *FileName*] [**-q**]

⇡Top of page

Parameters

***FileName* [*filename* ...] :** Specifies the pathname of an existing performance counter log. You can specify multiple input files.

-a : Appends output file instead of overwriting. This option does not apply to SQL format where the default is always to append.

-c *Path* [*path* ...] : Specifies the performance counter path to log. To specify multiple counter paths, separate them with a space and enclose the counter paths in quotation marks (for example, **"***CounterPath1 CounterPath2***"**).

-cf *FileName* : Specifies the pathname of the text file that lists the performance counters to be included in a relog file. Use this option to list counter paths in an input file, one per line. Default setting is all counters in the original log file are relogged.

-f {bin|csv|tsv|SQL} : Specifies the pathname of the output file format. The default format is *bin*. For a SQL database, the output file specifies the DSN!counter_log. You can specify the database location by using the ODBC manager to configure the DSN (Database System Name).

-t *value* : Specifies sample intervals in "*n*" records. Includes every nth data point in the relog file. Default is every data point.

-o {*output file* | *DSN!counter_log*} : Specifies the pathname of the output file or SQL database where the counters will be written.

-b *M/d/yyyy hh[:mm[:ss]]* : Specifies begin time for copying first record from the input file. Date and time must be in this exact format *M/d/yyyy hh:mm:ss*.

-e *M/d/yyyy hh[:mm[:ss]]* : Specifies end time for copying last record from the input file. Date and time must be in this exact format *M/d/yyyy hh:mm:ss*.

-config *FileName* : Specifies the pathname of the settings file that contains command-line parameters.

-q : Displays the performance counters and time ranges of log files specified in the input file.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Counter path format

The general format for counter paths is as follows: [\\Computer]*object*[parent/*instance#index*]*counter*] where:

The parent, instance, index, and counter components of the format may contain either a valid name or a wildcard character. The computer, parent, instance, and index components are not necessary for all counters.

You determine the counter paths to use based on the counter itself. For example, the *LogicalDisk* object has an instance *index*, so you must provide the *#index* or a wildcard. Therefore, you could use the following format:

\LogicalDisk(*/*#*)*

In comparison, the *Process* object does not require an instance *index*. Therefore, you could use the following format:

\Process(*)\ID Process

The following is a list of the possible formats:

- \\machine\object(parent/instance#index)\counter
- \\machine\object(parent/instance)\counter
- \\machine\object(instance#index)\counter
- \\machine\object(instance)\counter
- \\machine\object\counter
- \object(parent/instance#index)\counter
- \object(parent/instance)\counter
- \object(instance#index)\counter
- \object(instance)\counter
- \object\counter

If a wildcard character is specified in the parent name, all instances of the specified object that match the specified instance and counter fields will be returned.

If a wildcard character is specified in the instance name, all instances of the specified object and parent object will be returned if all instance names corresponding to the specified index match the wildcard character.

If a wildcard character is specified in the counter name, all counters of the specified object are returned.

Partial counter path string matches (for example, pro*) are not supported.

- Counter files

Counter files are text files that list one or more of the performance counters in the existing log. Copy the full counter name from the log or the **-q** output in [\\Computer \ *object* [*instance*] \ *counter*] format. List one counter path on each line.

- Copying counters

When executed, Relog copies specified counters from every record in the input file, converting the format if necessary. Wildcard paths are allowed in the counter file.

- Saving input file subsets

Use the **-t** parameter to specify that input files are inserted into output files at intervals of every nth record. By default, data is relogged from every record.

- Using **-b** and **-e** parameters with log files

You can specify that your output logs include records from before begin-time (**-b**) to provide data for counters that require computation values of the formatted value. The output file will have the last records from input files with timestamps less than the end-time (**-e**) parameter.

- Using the **-config** option

  The contents of the setting file used with the **-config** option should have the following format:

  [*command_option*]

  value

  where *command_option* is a command line option and *value* specifies its value. For example:

  [o]

  output.txt

  [f]

  csv

  [t]

  5
- For more information about incorporating **Relog** into your Windows Management Instrumentation (WMI) scripts, see Scripting Logs and Monitoring at the Microsoft Windows Resource Kits Web site.

⇡Top of page

Examples

To resample existing trace logs at fixed intervals of 30, list counter paths, output files and formats, type the following command:

Relog c:\perflogs\daily_trace_log.blg -cf counter_file.txt -o c:\perflogs\reduced_log.csv -t 30 -f csv

To resample existing trace logs at fixed intervals of 30, list counter paths and output files, type the following command:

Relog c:\perflogs\daily_trace_log.blg -cf counter_file.txt -o c:\perflogs\reduced_log.blg -t 30

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Logman

Tracerpt

Typeperf

Command-line reference A-Z

Top of page

Rem

Enables you to include comments (remarks) in a batch file or in your configuration files.

Syntax

rem [*comment*]

⇡Top of page

Parameters

***comment* :** Specifies any string of characters you want to include as a comment.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•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 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. The blank lines are ignored when processing the batch program.

⇡Top of page

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
```

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Echo

Command-line reference A-Z

Rename (ren)

Changes the name of a file or a set of files.

Syntax

rename [*Drive*:][*Path*] *filename1 filename2*

ren [*Drive*:][*Path*] *filename1 filename2*

⇡Top of page

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. If you use wildcards (* and ?), *filename2* specifies the new names for the files. You cannot specify a new drive or path when renaming files.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Renaming files

  You can rename all files matching the specified file name. You cannot use the **rename** command to rename files across drives or to move files to a different directory location.
- Using wildcards with rename

  You can use wildcards (* and ?) in either file name 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 file name that already exists, **rename** displays the following message:

Duplicate file name or file not found

⇡Top of page

Examples

Suppose you want to change the extensions of all the file names in the current directory that have the extension .txt; for example, you want to change the .txt extensions to .doc extensions. To make this change, type:

ren *.txt *.doc

To rename a file or directory named Chap10 to Part10, type:

ren chap10 part10

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Move

Copy

Xcopy

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

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 file names to the destination directory.

Syntax

replace [*drive1***:**][*path1*] *FileName* [*drive2***:**][*path2*] [**/a**] [**/p**] [**/r**] [**/w**]

replace [*drive1***:**][*path1*] *FileName* [*drive2***:**][*path2*] [**/p**] [**/r**] [**/s**] [**/w**] [**/u**]

⇡Top of page

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 file name 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 command-line option with the **/s** or **/u** command-line option.

/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 command-line option but attempt to replace a read-only file, an error results and stops the replacement operation.

/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.

/s : Searches all subdirectories of the destination directory and replaces matching files. You cannot use the **/s** command-line option with the **/a** command-line option. The **replace** command does not search subdirectories specified in *path1*.

/u : Replaces (updates) only those files on the destination directory that are older than those in the source directory. You cannot use the **/u** command-line option with the **/a** command-line option.

/? : Displays help at the command prompt.

⇧Top of page

Remarks

•Replace messages

As **replace** adds or replaces files, the file names are displayed on the screen. After the **replace** command is finished, a summary line is displayed 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** command-line option 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, click **attrib** in the Related Topics list.

•Replace exit codes

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

Exit code	Description
0	The **replace** command successfully replaced or added the files.
1	The **replace** command encountered an incorrect version of MS-DOS.
2	The **replace** command could not find the source files.

3	The **replace** command 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, click **if** in the Related Topics list.

↑Top of page

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:

```
replace a:\phones.cli c:\ /s
```

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:

```
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** command-line option is included, **replace** adds only those files from drive A that do not exist on drive C.

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Attrib

Command-line reference A-Z

Top of page

Reset session

Enables you to reset (delete) a session from the terminal server.

Syntax

reset session {*SessionName*|*SessionID*} [**/server:***ServerName*] [**/v**]

⇡Top of page

Parameters

***SessionName* :** The name of the session you want to reset. To determine the name of the session, use the **query session** command.

***SessionID* :** The ID of the session to reset.

/server:*ServerName* : Specifies the terminal server containing the session you want to reset. Otherwise, the current terminal server is used.

/v : Displays information about the actions being performed.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- You can always reset your own sessions, but you must have Full Control access permission to reset another user's session.
- Be aware that resetting a user's session without warning can result in loss of data at the session.
- You should reset a session only when it malfunctions or appears to have stopped responding.
- The **/server** parameter is required only if you use **reset session** from a remote server.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Query session

Terminal Services commands

Top of page

Manage Your Profile |Legal |Contact Us

Rexec

Runs commands on remote computers running the Rexec service (daemon). The **rexec** command authenticates the user name on the remote computer before executing the specified command. Windows XP and Windows 2000 do not provide the Rexec service. Used without parameters, **rexec** displays help.

Syntax

rexec [*Host*] [**-l** *UserName*] [**-n**] [*Command*]

⇡Top of page

Parameters

Host **:** Specifies the remote host (computer) on which to run *Command* by IP address or name.

-l *UserName* : Specifies the user name on the remote computer. If omitted, the user name of the user who is currently logged on is used.

-n : Redirects the input of **rexec** to the NUL device. This prevents the display of the command results on the local computer.

Command **:** Specifies the command to run on the remote computer.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Standard operation

  The **rexec** command prompts the user for a password and authenticates the given password on the remote computer. If the authentication succeeds, the command is run.

  The **rexec** command 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. The **rexec** normally quits when the remote command quits.
- Using Redirection symbols

  for redirection to occur on the remote computer, enclose redirection symbols in quotation marks (for example, **">>"**). If you do not use quotation marks, 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 by using **rexec**. You can, however, use **telnet** instead.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To execute the **telcon** command on the remote computer vax1 using the name admin1, type the following command:

rexec vax1 -l admin1 telcon

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Rexec

Using command redirection operators

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Rmdir (rd)

Removes (that is, deletes) a directory.

Syntax

rmdir [*Drive***:**]*Path* [**/s**] [**/q**]

rd [*Drive***:**]*Path* [**/s**] [**/q**]

Top of page

Parameters

[*Drive*:]*Path* : Specifies the location and name of the directory that you want to delete.

/s : Removes the specified directory and all subdirectories including any files. Use **/s** to remove a tree.

/q : Runs **rmdir** in quiet mode. Deletes directories without confirmation.

/? : Displays help at the command prompt.

Top of page

Remarks

- Using **rmdir** at the Recovery Console

 The **rmdir** command, with different parameters, is available from the Recovery Console.

- 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, the following message appears:

  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 Related Topics.
- Using the backslash character with the path parameter

  If you insert a backslash (\) before the first directory name in path, the directory is treated 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, the directory is treated 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, the following message appears:

  The process cannot access the file because it is being used by another process.

Top of page

Examples

To delete a directory named \User\Smith, first ensure that the directory is empty. To do this, type:

dir \user\smith /a

Only the "." and ".." symbols should display.

Then, from any directory except \User\Smith, type:

rmdir \user\smith

To delete the directory \User and all of the subdirectories and files, type:

rmdir /s \user

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Attrib

Mkdir

Dir

Command-line reference A-Z

Top of page

Route

Displays and modifies the entries in the local IP routing table. Used without parameters, **route** displays help.

Syntax

route [**-f**] [**-p**] [*Command* [*Destination*] [**mask** *Netmask*] [*Gateway*] [**metric** *Metric*]]
[**if** *Interface*]]

Top of page

Parameters

-f : Clears the routing table of all entries that are not host routes (routes with a netmask of 255.255.255.255), the loopback network route (routes with a destination of 127.0.0.0 and a netmask of 255.0.0.0), or a multicast route (routes with a destination of 224.0.0.0 and a netmask of 240.0.0.0). If this is used in conjunction with one of the commands (such as **add**, **change**, or **delete**), the table is cleared prior to running the command.

-p : When used with the **add** command, the specified route is added to the registry and is used to initialize the IP routing table whenever the TCP/IP protocol is started. By default, added routes are not preserved when the TCP/IP protocol is started. When used with the **print** command, the list of persistent routes is displayed. This parameter is ignored for all other commands. Persistent routes are stored in the registry location **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes**.

***Command* :** Specifies the command you want to run. The following table lists valid commands.

Command	Purpose
add	Adds a route.
change	Modifies an existing route.
delete	Deletes a route or routes.

print	Prints a route or routes.

Destination **:** Specifies the network destination of the route. The destination can be an IP network address (where the host bits of the network address are set to 0), an IP address for a host route, or 0.0.0.0 for the default route.

mask *Netmask* : Specifies the netmask (also known as a subnet mask) associated with the network destination. The subnet mask can be the appropriate subnet mask for an IP network address, 255.255.255.255 for a host route, or 0.0.0.0 for the default route. If omitted, the subnet mask 255.255.255.255 is used. Because of the relationship between the destination and the subnet mask in defining routes, the destination cannot be more specific than its corresponding subnet mask. In other words, there cannot be a bit set to 1 in the destination if the corresponding bit in the subnet mask is a 0.

Gateway **:** Specifies the forwarding or next hop IP address over which the set of addresses defined by the network destination and subnet mask are reachable. For locally attached subnet routes, the gateway address is the IP address assigned to the interface that is attached to the subnet. For remote routes, available across one or more routers, the gateway address is a directly reachable IP address that is assigned to a neighboring router.

metric *Metric* : Specifies an integer cost metric (ranging from 1 to 9999) for the route, which is used when choosing among multiple routes in the routing table that most closely match the destination address of a packet being forwarded. The route with the lowest metric is chosen. The metric can reflect the number of hops, the speed of the path, path reliability, path throughput, or administrative properties.

if *Interface* : Specifies the interface index for the interface over which the destination is reachable. For a list of interfaces and their corresponding interface indexes, use the display of the **route print** command. You can use either decimal or hexadecimal values for the interface index. For hexadecimal values, precede the hexadecimal number with **0x**. When the **if** parameter is omitted, the interface is determined from the gateway address.

/? : Displays help at the command prompt.

Top of page

Remarks

- Large values in the **metric** column of the routing table are the result of allowing TCP/IP to automatically determine the metric for routes in the routing table based on the configuration of IP address, subnet mask, and default gateway for each LAN interface. Automatic determination of the interface metric, enabled by default, determines the speed of each interface and adjusts the metrics of routes for each interface so that the fastest interface creates the routes with the lowest metric. To remove the large metrics, disable the automatic determination of the interface metric from the advanced properties of the TCP/IP protocol for each LAN connection.
- Names can be used for *Destination* if an appropriate entry exists in the local Networks file stored in the *systemroot*\System32\Drivers\Etc folder. Names can be used for the *gateway* as long as they can be resolved to an IP address through standard host name resolution techniques such as Domain Name System (DNS) queries, use of the local Hosts file stored in the *systemroot* \system32\drivers\etc folder, and NetBIOS name resolution.
- If the command is **print** or **delete**, the *Gateway* parameter can be omitted and wildcards can be used for the destination and gateway. The *Destination* value can be a wildcard value specified by an asterisk (*). If the destination specified contains an asterisk (*) or a question mark (?), it is treated as a wildcard and only matching destination routes are printed or deleted. The asterisk matches any string, and the question mark matches any single character. For example, 10.*.1, 192.168.*, 127.*, and *224* are all valid uses of the asterisk wildcard.
- Using an invalid combination of a destination and subnet mask (netmask) value displays a "Route: bad gateway address netmask" error message. This error message appears when the destination contains one or more bits set to 1 in bit locations where the corresponding subnet mask bit is set to 0. To test this condition, express the destination and subnet mask using binary notation. The subnet mask in binary notation consists of a series of 1 bits, representing the network address portion of the destination, and a series of 0 bits, representing the host address portion of the destination. Check to determine whether there are bits in the destination that are set to 1 for the portion of the destination that is the host address (as defined by the subnet mask).
- The **-p** parameter is only supported on the route command for Windows NT 4.0, Windows 2000, Windows Millennium Edition, and Windows XP. This parameter is not supported by the **route** command for Windows 95 or Windows 98.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To display the entire contents of the IP routing table, type:

route print

To display the routes in the IP routing table that begin with *10.*, type:

route print 10.*

To add a default route with the default gateway address of 192.168.12.1, type:

route add 0.0.0.0 mask 0.0.0.0 192.168.12.1

To add a route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0 and the next hop address of 10.27.0.1, type:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1

To add a persistent route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0 and the next hop address of 10.27.0.1, type:

route -p add 10.41.0.0 mask 255.255.0.0 10.27.0.1

To add a route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0, the next hop address of 10.27.0.1, and the cost metric of 7, type:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 metric 7

To add a route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0, the next hop address of 10.27.0.1, and using the interface index 0x3, type:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 if 0x3

To delete the route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0, type:

route delete 10.41.0.0 mask 255.255.0.0

To delete all routes in the IP routing table that begin with *10.*, type:

route delete 10.*

To change the next hop address of the route with the destination of 10.41.0.0 and the subnet mask of 255.255.0.0 from 10.27.0.1 to 10.27.0.25, type:

route change 10.41.0.0 mask 255.255.0.0 10.27.0.25

⇧Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

TCP/IP utilities and services

Top of page

Manage Your Profile |Legal |Contact Us

Rsh

Runs commands on remote computers running the RSH service or daemon. Windows XP and Windows 2000 do not provide an RSH service. An RSH service called Rshsvc.exe is provided with the Windows 2000 Server Resource Kit. Used without parameters, **rsh** displays help.

Syntax

rsh [*Host*] [**-l** *UserName*] [**-n**] [*Command*]

⇧Top of page

Parameters

***Host* :** Specifies the remote host (computer) on which to run *Command*.

-l *UserName* : Specifies the user name to use on the remote computer. If omitted, the user name of the user who is currently logged on is used.

-n : Redirects the input of **rsh** to the NUL device. This prevents the display of the command results on the local computer.

***Command* :** Specifies the command to run.

/? : Displays help at the command prompt.

⇧Top of page

Remarks

•Standard operation

The **rsh** command 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 quits when the remote command quits.

- Using Redirection symbols

  Enclose redirection symbols in quotation marks for redirection to occur on the remote computer (for example, ">>"). If you do not use quotation marks, 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 XP Professional that is logged on to a domain, the primary domain controller for the domain must be available to verify the user name or the **rsh** command fails.
- The .rhosts file

  The .rhosts file typically permits network access on UNIX systems. The .rhosts file lists computer names and associated logon names that have access to remote computers. When you run **rcp**, **rexec**, or **rsh** commands remotely with a correctly configured .rhosts file, you do not need to provide logon and password information for the remote computer.

  The .rhosts file is a text file in which 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 pound sign (#). For example:

  ```
  host7 #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 the specific implementation of the .rhosts file on a remote computer, see the remote system documentation.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

⇡Top of page

Examples

To execute the telcon command on the remote computer vax1 using the name admin1, type:

```
rsh vax1 -l admin1 telcon
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Using command redirection operators

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Rsm

Manages media resources using Removable Storage. Using the **rsm** command, you can run batch scripts for applications that do not currently support the Removable Storage API.

To allocate media from a media pool

Syntax

rsm allocate __/m__*media_pool_name* **/o**{**errunavail**|**new**|**next**} {**/l**{**g**|**f**}*logical_media_id* | **/p**{**g**|**f**} *partition_id*} [__/ln__*logical_media_name*] [__/ld__*logical_media_description*] [__/pn__*partition_name*] [__/pd__*partition_description*] [__/t__*timeout*] [**/b**]

⇡Top of page

Parameters

__/m__***media_pool_name*** **:** Media are allocated from the specified media pool. This means that you will have allocated media in that pool.

/o : Permits the use of one of the parameters listed in the following table.

Value	Description
errunavail	Prevents the submission of an operator request for new media if none can be allocated with the specified constraints.
new	Allocates a partition that cannot be shared with another application. This can be used to reserve the second side of two-sided media.
next	Allocates the next side of media that was previously allocated using the **new** parameter.

__/l{g|f}__***logical_media_id*** **:** Specifies the media to be allocated, using the logical media ID. You can use the GUID (with the **lg** command-line option), or the friendly name (with the **lf** command-line option). *Logical_media_id* specifies the next side of multi-sided media to allocate. This parameter is optional and must be used with the **/o** command-line option and the **next** parameter. After deallocating this media, the logical media ID is invalid.

/p{g|f}*partition_id* : Specifies the partition to be allocated, using the partition ID. You can use the GUID (with the **pg** command-line option), or the friendly name (with the **pf** command-line option). This parameter is optional and remains persistent even after the media is deallocated.

/ln*logical_media_name* : Specifies the media to be allocated, using the logical media name.

/ld*logical_media_description* : Specifies the media to be allocated, using the logical media description.

/pn*partition_name* : Specifies the partition to be allocated, using the partition name.

/pd*partition_description* : Specifies the media partition to be allocated, using the partition description.

/t*timeout* : Specifies the command timeout, in milliseconds. The default timeout value is infinite.

/b : Only the GUID for the allocate operation is displayed. This aids in scripting where you want to pass the output of one command to the next with minimal parsing.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can use logical media names and side names can be used in other commands to specify media as the parameter to the **/lf** or **/pf** switch, respectively. If logical media names (which are friendly names) are not used with the allocate command, you can use only GUIDs in subsequent commands to specify logical media.

To create a media pool

Top of page

Syntax

rsm createpool /m*media_pool_name* **/a**{**existing**|**always**|**new**} [**/t**{**g**|**f**}*media_pool_type_id*] [**/d**] [**/r**]

Top of page

Parameters

/m*media_pool_name* : Specifies the name of the media pool to be created.

/a : Permits the use of one of the parameters listed in the following table.

Value	Description
existing	Opens the existing media pool or returns an error if the media pool specified does not exist.
always	Opens the existing media pool or creates a new media pool if not found.
new	Creates a new media pool or returns an error if the media pool specified already exists.

/tg*media_pool_type_id* : Specifies the type of media the media pool will contain, using the GUID. The default type is a media pool that contains other media pools.

/tf*media_pool_type_id* : Specifies the type of media the media pool will contain, using the friendly name. The default type is a media pool that contains other media pools.

/d : Permits the media pool to automatically draw media from the free media pool. If the **/d** switch is not included, the media pool will not be permitted to draw media from the free media pool.

/r : Permits the media pool to automatically return media to the free media pool. If the **/r** switch is not included, the media pool will not be permitted to return media to the free media pool.

/?: Displays help at the command prompt.

To deallocate media

Top of page

Syntax

rsm deallocate /l{**g**|**f**}*logical_media_id* | **/p**{**g**|**f**}*partition_id*

Top of page

Parameters

/lg*logical_media_id* : Specifies the logical media to deallocate, using the GUID.

/lf*logical_media_id* : Specifies the logical media to deallocate, using the friendly name.

/pg*partition_id* : Specifies the media side to deallocate, using the GUID.

/pf*partition_id* : Specifies the media side to deallocate, using the friendly name.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can use the logical media name or the partition name to specify the logical media to deallocate only if one of these names were specified with the **allocate** command using the **/ln** or **/pn** switch respectively. Otherwise, you must specify either the logical media ID (LMID) or the partition ID (PARTID) instead.

To delete a media pool

Top of page

Syntax

rsm deletepool /m*media_pool_name*

Top of page

Parameters

/m*media_pool_name* : Specifies the name of the media pool to be deleted.

/?: Displays help at the command prompt.

To dismount media from a drive

Top of page

Syntax

rsm dismount {**/l**{**g**|**f**}*logical_media_id* | **/p**{**g**|**f**}*partition_id*
rsm dismount [**/o**[**deferred**]]}

Top of page

Parameters

/lg*logical_media_id* : Specifies the logical media to dismount, using the GUID.

/lf*logical_media_id* : Specifies the logical media to dismount, using the friendly name.

/pg*partition_id* : Specifies the media side to dismount, using the GUID.

/pf*partition_id* : Specifies the media side to dismount, using the friendly name.

/o : When used with the optional **deferred** parameter, this optional switch marks the media as dismountable, but the media is kept in the drive. Subsequent mount commands can be completed normally. If not used, the media is dismounted from the drive immediately.

/?: Displays help at the command prompt.

Top of page

Remarks

- The logical media name or the partition name can be used to specify the logical media to dismount only if one of these names were specified with the **allocate** command using the **/ln** or **/pn** switch, respectively. Otherwise, you must specify either the logical media ID (LMID) or the partition ID (PARTID) instead.

To eject media from a library

The media to be ejected can be specified in one of three ways:

- •You can specify the physical media to eject using either the physical-media ID (PMID) or the physical media name.
- •You can eject the media in a specified slot within a specified library.
- •You can eject the media in a specified drive within a specified library.

Top of page

Syntax

rsm eject {**/p**{**g**|**f**}*physical_media_id* | **/s**{**g**|**f**}*slot_id* **/l**{**g**|**f**}*library_id* | **/d**{**g**|**f**}*drive_id* **/l**{**g**|**f**} *library_id*
rsm eject [**/o***eject_operation_id*]
rsm eject [**/a**{**start**|**stop**|**queue**}]
rsm eject [**/b**]

Top of page

Parameters

/pg*physical_media_id* : Specifies the physical media to eject, using the GUID.

/pf*physical_media_id* : Specifies the physical media to eject, using the friendly name.

/sg*slot_id* : Specifies the slot holding the media to eject, using the GUID.

/sf*slot_id* : Specifies the slot holding the media to eject, using the friendly name.

/lg*library_id* : Specifies the library containing the slot or drive from which to eject the media, using the GUID.

/lf*library_id* : Specifies the library containing the slot or drive from which to eject the media, using the friendly name.

/dg*drive_id* : Specifies the drive holding the media to eject, using the GUID.

/df*drive_id* : Specifies the drive holding the media to eject, using the friendly name.

/o*eject_operation_id* : Used to specify the GUID for the particular eject operation. The optional **/o** switch can be used in conjunction with the **/a** switch and the **stop** parameter to terminate a particular eject operation. This can also be used in conjunction with the **/a** switch and the **start** parameter to display the GUID of the particular eject operation.

/a : Permits the use of one of the parameters listed in the following table.

Value	Description
Start	The default, starts the eject operation immediately. The media is ejected until a timeout occurs, or unless another eject command is issued with the **/a** command-line option and the **stop** parameter. Such eject commands must also specify the eject operation GUID using the **/o** command-line option. The timeout parameter is specified in the library object (for all eject operations) for the library. To set this timeout parameter, you must use the Removable Storage API. Can also be used in conjunction with the **/o** switch to display the GUID of a particular eject operation.
Stop	Terminates the eject operation prior to a timeout expiring. The particular eject operation can be determined using the GUID displayed when the **start** parameter is used with the **/a** switch and the **/o** switch.
Queue	Queues the media for later ejection. This can be used for libraries with multi-slot inject/eject (IE) ports.

/b : Displays only the eject operation GUID for scripting purposes.

/?: Displays help at the command prompt.

To eject media from an ATAPI changer

⇡Top of page

Syntax

rsm ejectatapi /n*atapi_changer_number*

⇡Top of page

Parameters

/n*atapi_changer_number* : Specifies the changer number. *Atapi_changer_number* is the number found at the end of the string for the device name of the changer. For example, \\.\CdChanger0 has 0 as the ATAPI changer number.

/?: Displays help at the command prompt.

Top of page

Remarks

- Before you run this command, manually stop the **ntmssvc** service.

To inventory the media in a specified automated library

Top of page

Syntax

rsm inventory /l{**g**|**f**}*library_id* **/a**{**full**|**fast**|**default**|**none**|**stop**}

Top of page

Parameters

/lf*library_id* : Specifies the library to inventory, using the friendly name.

/lg*library_id* : Specifies the library to inventory, using the GUID.

/a : Required. Specifies the type of inventory operation to perform. The following table lists valid inventory operations.

Value	Description
Full	Performs a full on-media inventory of the library. Removable Storage mounts each tape or disk in the library and reads the on-media identifier.
Fast	Performs a bar code inventory, if the specified library has a bar code reader installed. If the library has no bar code reader, Removable Storage checks the storage slots and reads the on-media identifier on media in slots that were previously empty.
Default	Performs an inventory using the default method specified in the library's **Properties** dialog box.
None	Performs no inventory.

Stop	Stops the current inventory for the specified library, if one is being performed.

/?: Displays help at the command prompt.

To mount media in the designated library

The logical media to be mounted can be specified using either the logical-media ID (LMID) or the logical media name.

Top of page

Syntax

rsm mount {**/l**{**g**|**f**}*logical_media_id* | **/p**{**g**|**f**}*partition_id* | [**/s**{**g**|**f**}*slot_id* **/c**{**g**|**f**}*changer_id*}
rsm mount [**/d**{**g**|**f**}*drive_id*]
rsm mount /o{**errunavail**|**drive**|**read**|**write**|**offline**}
rsm mount [**/r**{**normal**|**high**|**low**|**highest**|**lowest**}]
rsm mount [**/t***timeout*]

Top of page

Parameters

/lf*logical_media_id* : Specifies the logical media to mount, using the friendly name.

/pf*partition_id* : Specifies the media side to mount, using the friendly name.

/pg*partition_id*

/lg*logical_media_id* : Specifies the logical media to mount, using the GUID.

/cg*changer_id* : Specifies the changer that contains the media to be mounted, using the GUID. This can only be used in conjunction with the **/sg** switch and the slot GUID.

/cf*changer_id* : Specifies the changer that contains the media to be mounted, using the friendly name. This can be used only in conjunction with the **/sg** switch and the slot GUID.

/sg*slot_id* : Specifies the media slot that contains the media to be mounted, using the GUID. This can only be used in conjunction with the **/cg** switch and the changer GUID.

/sf*slot_id* **:** Specifies the media slot that contains the media to be mounted, using the friendly name. This can only be used in conjunction with the **/cg** switch and the changer GUID.

/dg*drive_id* **:** Specifies the particular drive on which to mount the applicable media, using the GUID. This parameter is optional, and must be used in conjunction with the **/o** switch and the **drive** parameter.

/df*drive_id* **:** Specifies the particular drive on which to mount the applicable media, using the friendly name. This parameter is optional, and must be used in conjunction with the **/o** switch and the **drive** parameter.

/o : Permits the use of one of the parameters listed in the following table.

Value	Description
Errunavail	Generates an error if either the media or the drive is unavailable.
Drive	Specifies that a particular drive is to be mounted. This parameter is used in conjunction with the **/d** switch.
Read	Mounts the media for read access.
Write	Mounts the media for write access. If this parameter is used, completed media will not be mounted.
Offline	Generates an error if the media is offline.

/r : Optionally specifies the mount order, or priority. Mount priority may also be specified using one of the listed parameters, **normal** (the default), **high**, **low**, **highest**, or **lowest**.

/t : Optionally specifies the command timeout, in milliseconds. The default timeout is infinite.

/?: Displays help at the command prompt.

Top of page

Remarks

- When using the mount command, you can specify the media to be mounted using either the **/l** switch, the **/p** switch, or a combination of the **/s** switch and the **/c** switch.

To refresh a library, physical media, or all devices of a particular media type

This command causes a single poll of the target devices so that the Removable Storage database contains the current state of the device. This command can be useful after media insert or eject operations.

⇡Top of page

Syntax

rsm refresh {**/l**{**g**|**f**]*library_id* | **/p**{**g**|**f**}*physical_media_id* | **/tg***media_type_id*}

⇡Top of page

Parameters

/lg*library_id* : Specifies the library to refresh, using the GUID.

/lf*library_id* : Specifies the library to refresh, using the friendly name.

/pg*physical_media_id* : Specifies the physical media to refresh, using the GUID.

/pf*physical_media_id* : Specifies the physical media to refresh, using the friendly name.

/tg*media_type_id* : Specifies the media type to be refreshed. Only the GUID can be specified. This parameter can be used to refresh all removable media devices by specifying the GUID for the removable media. This GUID can be determined using the view command as follows: **rsm view /tmedia_type /guiddisplay**. **/?Displays help at the command prompt.**

To display a list of media objects

⇡Top of page

Syntax

rsm view /t{**drive**|**library**|**changer**|**storageslot**|**iedoor**|**ieport**|**physical_media**| **media_pool**|**partition**|**logical_media**|**media_type**|**drive_type**|**librequest**}
rsm view [**/cg***container_id*]
rsm view [**/guiddisplay**]
rsm view [**/b**]

⇡Top of page

Parameters

/t {drive|library|changer|storageslot|iedoor|ieport|physical_media| media_pool|partition| logical_media|media_type|drive_type|librequest} : Displays a list of media objects of the specified type. When used without any parameter, the command displays a list of all media pools in the Removable Storage system (collection of libraries).

/cg*container_id* : Specifies the GUID for the object container. The type of container depends on the object type (parameter) specified with the **/t** switch. If the container ID is not specified, all instances of the applicable object type are displayed.

/guiddisplay : Displays both the GUID and the friendly name for objects.

/b : Displays only the object GUID for scripting purposes. **/?Displays help at the command prompt.**

⇡Top of page

Remarks

- If the **/guiddisplay** switch and the **/b** switch are not used, only the friendly names for objects are displayed.

⇡Top of page

Remarks

- If a command succeeds, then the code ERROR_SUCCESS is returned. All commands that fail return an error code, which can be used for scripting purposes. The error code is either a system-defined error code or one of the error codes listed in the following table.

Error code	Description
536870913	Invalid arguments were specified. Frequently, this is caused by a space after an argument switch, for example, **/t 50** instead of **/t50**.
536870914	Duplicate argument switches were specified. For example, the **allocate** command used with two **/m** switches.
536870915	No GUID matches the friendly name that was specified. Check capitalization because friendly names are case-sensitive.
536870916	An insufficient number of argument switches were specified. Check to see if a required switch is missing.
536870917	An invalid GUID was specified. Use the **view** command to determine the correct GUID for an object.
536870918	This is returned only by the **ejectatapi** command. Verify that the ATAPI changer is functioning correctly.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Removable Storage overview

Understanding Removable Storage

Using Removable Storage

⇡Top of page

Manage Your Profile |Legal |Contact Us

Runas

Allows a user to run specific tools and programs with different permissions than the user's current logon provides.

Syntax

runas [{**/profile**|**/noprofile**}] [**/env**] [**/netonly**] [**/smartcard**] [**/showtrustlevels**] [**/trustlevel**] **/ user:***UserAccountName* **program**

⇡Top of page

Parameters

/profile : Loads the user's profile. **/profile** is the default.

/no profile : Specifies that the user's profile is not to be loaded. This allows the application to load more quickly, but it can also cause a malfunction in some applications.

/env : Specifies that the current network environment be used instead of the user's local environment.

/netonly : Indicates that the user information specified is for remote access only.

/smartcard : Indicates whether the credentials are to be supplied from a smartcard.

/showtrustlevels : Lists the **/trustlevel** options.

/trustlevel : Specifies the level of authorization at which the application is to run. Use **/ showtrustlevels** to see the trust levels available.

/user:*UserAccountName* : Specifies the name of the user account under which to run the program. The user account format should be *user@domain* or *Domain\User*.

program : Specifies the program or command to run using the account specified in **/user**.

/? : Displays help at the command prompt.

Top of page

Remarks

- It is good practice for administrators to use an account with restrictive permissions to perform routine, nonadministrative tasks, and to use an account with broader permissions only when performing specific administrative tasks. To accomplish this without logging off and back on, log on with a regular user account, and then use the **runas** command to run the tools that require the broader permissions.
- For examples of the use of the **runas** command, see Related Topics.
- The use of **runas** is not restricted to administrator accounts, although that is the most common use. Any user with multiple accounts can use **runas** to run a program, MMC console, or Control Panel item with alternate credentials.
- If you want to use the Administrator account on your computer, for the **/user:** parameter, type one of the following:

  /user:*AdministratorAccountName***@***ComputerName*

  /user:*ComputerName\AdministratorAccountName*
- If you want to use this command as a domain administrator, type one of the following:

  /user:*AdministratorAccountName***@***DomainName*

  /user:*DomainName\AdministratorAccountName*
- With the **runas** command, you can run programs (*.exe), saved MMC consoles (*.msc), shortcuts to programs and saved MMC consoles, and Control Panel items. You can run them as an administrator while you are logged on to your computer as a member of another group, such as the Users or Power Users group.
- You can use the **runas** command start any program, MMC console, or Control Panel item. As long as you provide the appropriate user account and password information, the user account has the ability to log on to the computer, and the program, MMC console, or Control Panel item is available on the system and to the user account.
- With the **runas** command, you can administer a server in another forest (the computer from which you run a tool and the server you administer are in different domains).
- If you try to start a program, MMC console, or Control Panel item from a network location using **runas**, it might fail because the credentials used to connect to the network share are different from the credentials used to start the program. The latter credentials may not be able to gain access to the same network share.
- Some items, such as the Printers folder and desktop items, are opened indirectly and cannot be started with the **runas** command.

- If the **runas** command fails, the **Secondary Logon** service might not be running or the user account you are using might not be valid. To check the status of the **Secondary Logon** service, in Computer Management, click **Services and Applications**, and then click **Services**. To test the user account, try logging on to the appropriate domain using the account.

⇡Top of page

Examples

To start an instance of the command prompt as an administrator on the local computer, type:

runas /user:*localmachinename*\administrator cmd

When prompted, type the administrator password.

To start an instance of the Computer Management snap-in using a domain administrator account called **companydomain\domainadmin**, type:

runas /user:companydomain\domainadmin "mmc %windir%\system32\compmgmt.msc"

When prompted, type the account password.

To start an instance of Notepad using a domain administrator account called **user** in a domain called **domain.microsoft.com**, type:

runas /user:user@domain.microsoft.com "notepad my_file.txt"

When prompted, type the account password.

To start an instance of a command prompt window, saved MMC console, Control Panel item, or program that will administer a server in another forest, type:

runas /netonly /user:*domain\username "command"*

domain\username must be a user with sufficient permissions to administer the server. When prompted, type the account password.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Command-line reference A-Z

Create a shortcut using runas command parameters

Top of page

Manage Your Profile |Legal |Contact Us

SC

Communicates with the Service Controller and installed services. SC.exe retrieves and sets control information about services. You can use SC.exe for testing and debugging service programs. Service properties stored in the registry can be set to control how service applications are started at boot time and run as background processes. SC.exe parameters can configure a specific service, retrieve the current status of a service, as well as stop and start a service. You can create batch files that call various SC.exe commands to automate the startup or shutdown sequence of services. SC.exe provides capabilities similar to Services in the Administrative Tools item in Control Panel.

For the command syntax, click any of the following **sc** commands:

sc boot

Indicates whether the last boot should be saved as the last-known-good configuration.

Syntax

sc [*ServerName*] **boot** [{**bad**|**OK**}]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the Universal Naming Convention (UNC) format ("\\myserver"). To run SC.exe locally, ignore this parameter.

[{bad|OK}] : Specifies whether the last boot was bad or whether it should be saved as the last-known-good boot configuration.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **sc boot** command:

sc boot ok

sc boot bad

sc config

Modifies the value of a service's entries in the registry and in the Service Control Manager's database.

Top of page

Syntax

sc [*ServerName*] **config** [*ServiceName*] [**type=** {**own**|**share**|**kernel**|**filesys**|**rec**|**adapt**|**interact type=** {**own**|**share**}}] [**start=** {**boot**|**system**|**auto**|**demand**|**disabled**}] [**error=** {**normal**|**severe**|**critical**|**ignore**}] [**binpath=** *BinaryPathName*] [**group=** *LoadOrderGroup*] [**tag=** {**yes**|**no**}] [**depend=** *dependencies*] [**obj=** {*AccountName*|*ObjectName*}] [**displayname=** *DisplayName*] [**password=** *Password*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the Universal Naming Convention (UNC) format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

type= {own|share|kernel|filesys|rec|adapt|interact type= {own|share}} : Specifies the service type.

Value	Description
own	The service runs in its own process. It does not share an executable file with other services. This is the default.
share	The service runs as a shared process. It shares an executable file with other services.
kernel	Driver.
filesys	File system driver.
rec	File system-recognized driver (identifies file systems used on the computer).
adapt	Adapter driver (identifies hardware items such as keyboard, mouse, and disk drive).
interact	The service can interact with the desktop, receiving input from users. Interactive services must be run under the LocalSystem account. This type must be used in conjunction with **type= own** or **type= shared** (for example, **type= interact type= own**). Using **type= interact** by itself will generate an invalid parameter error.

start= {boot|system|auto|demand|disabled} : Specifies the start type for the service.

Value	Description
boot	A device driver that is loaded by the boot loader.
system	A device driver that is started during kernel initialization.
auto	A service that automatically starts each time the computer is restarted and runs even if no one logs on to the computer.
demand	A service that must be manually started. This is the default value if **start=** is not specified.
disabled	A service that cannot be started. To start a disabled service, change the start type to some other value.

error= {normal|severe|critical|ignore} : Specifies the severity of the error if the service fails to start during boot.

Value	Description
normal	The error is logged and a message box is displayed informing the user that a service has failed to start. Startup will continue. This is the default setting.
severe	The error is logged (if possible). The computer attempts to restart with the last-known-good configuration. This could result in the computer being able to restart, but the service may still be unable to run.
critical	The error is logged (if possible). The computer attempts to restart with the last-known-good configuration. If the last-known-good configuration fails, startup also fails, and the boot process halts with a Stop error.
ignore	The error is logged and startup continues. No notification is given to the user beyond recording the error in the Event Log.

binpath= ***BinaryPathName*** **:** Specifies a path to the service binary file.

group= ***LoadOrderGroup*** **:** Specifies the name of the group of which this service is a member. The list of groups is stored in the registry in the HKLM\System \CurrentControlSet\Control\ServiceGroupOrder subkey. The default is null.

tag= {yes|no} : Specifies whether or not to obtain a TagID from the CreateService call. Tags are only used for boot-start and system-start drivers.

depend= ***dependencies*** **:** Specifies the names of services or groups which must start before this service. The names are separated by forward slashes (/).

obj= {*AccountName*|*ObjectName*} : Specifies a name of an account in which a service will run, or specifies a name of the Windows driver object in which the driver will run. The default is **LocalSystem**.

displayname= ***DisplayName*** **:** Specifies a friendly, meaningful name that can be used in user-interface programs to identify the service to users. For example, the subkey name of one service is wuauserv, which is not be helpful to the user, and the display name is Automatic Updates.

password= ***Password*** **:** Specifies a password. This is required if an account other than the LocalSystem account is used.

/?: Displays help at the command prompt.

Top of page

Remarks

- Without a space between a parameter and its value (for example, **type= own**, not **type=own**), the operation will fail.

Top of page

Examples

The following example shows how you can use the **sc config** command:

sc config NewService binpath= "ntsd -d c:\windows\system32\NewServ.exe"

sc continue

Sends a CONTINUE control request to a service in order to resume a paused service.

Top of page

Syntax

sc [*ServerName*] **continue** [*ServiceName*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

/?: Displays help at the command prompt.

Top of page

Remarks

- Use the **continue** operation to resume a paused service.

Top of page

Examples

The following example shows how you can use the **sc continue** command:

sc continue tapisrv

sc control

Sends a CONTROL B to a service.

Top of page

Syntax

sc [*ServerName*] **control** [*ServiceName*] [{**paramchange**|**netbindadd**|**netbindremove**|**netbindenable**|**netbinddisable**|*UserDefinedControlB*}]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

{paramchange|netbindadd|netbindremove|netbindenable|netbinddisable|*UserDefinedControlB*} : Specifies a control to send to a service.

/?: Displays help at the command prompt.

sc create

Creates a subkey and entries for the service in the registry and in the Service Control Manager's database.

Top of page

Syntax

sc [*ServerName*] **create** [*ServiceName*] [**type=** {**own**|**share**|**kernel**|**filesys**|**rec**|**adapt**|**interact type=** {**own**|**share**}}] [**start=** {**boot**|**system**|**auto**|**demand**|**disabled**}] [**error=** {**normal**|**severe**|**critical**|**ignore**}] [**binpath=** *BinaryPathName*] [**group=** *LoadOrderGroup*] [**tag=** {**yes**|**no**}] [**depend=** *dependencies*] [**obj=** {*AccountName*|*ObjectName*}] [**displayname=** *DisplayName*] [**password=** *Password*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

type= {own|share|kernel|filesys|rec|adapt|interact type= {own|share}} : Specifies the service type. The default is **type= own**.

Value	Description
own	The service runs in its own process. It does not share an executable file with other services. This is the default.
share	The service runs as a shared process. It shares an executable file with other services.
kernel	Driver.
filesys	File system driver.
rec	File system recognized driver (identifies file systems used on the computer).
interact	The service can interact with the desktop, receiving input from users. Interactive services must be run under the LocalSystem account. This type must be used in conjunction with **type= own** or **type= shared** (that is, **type= interact type= own**). Using **type= interact** by itself will generate an invalid parameter error.

start= {boot|system|auto|demand|disabled} : Specifies the start type for the service. The default start is **start= demand**.

boot	A device driver that is loaded by the boot loader.
system	A device driver that is started during kernel initialization.
auto	A service that automatically starts each time the computer is restarted and runs even if no one logs on to the computer.

demand	A service that must be manually started. This is the default value if **start=** is not specified.
disabled	A service that cannot be started. To start a disabled service, change the start type to some other value.

error= {normal|severe|critical|ignore} : Specifies the severity of the error if the service fails to start during boot. The default is **error= normal**.

normal	The error is logged and a message box is displayed informing the user that a service has failed to start. Startup will continue. This is the default setting.
severe	The error is logged (if possible). The computer attempts to restart with the last-known-good configuration. This could result in the computer being able to restart, but the service may still be unable to run.
critical	The error is logged (if possible). The computer attempts to restart with the last-known-good configuration. If the last-known-good configuration fails, startup also fails, and the boot process halts with a Stop error.
ignore	The error is logged and startup continues. No notification is given to the user beyond recording the error in the Event Log.

binpath= ***BinaryPathName*** **:** Specifies a path to the service binary file. There is no default for **binpath=** and this string must be supplied.

group= ***LoadOrderGroup*** **:** Specifies the name of the group of which this service is a member. The list of groups is stored in the registry in the HKLM\System \CurrentControlSet\Control\ServiceGroupOrder subkey. The default is null.

tag= {yes|no} : Specifies whether or not to obtain a TagID from the CreateService call. Tags are only used for boot-start and system-start drivers.

depend= ***dependencies*** **:** Specifies the names of services or groups that must start before this service. The names are separated by forward slashes (/).

obj= {*AccountName*|*ObjectName*} : Specifies a name of an account in which a service will run, or specifies a name of the Windows driver object in which the driver will run.

displayname= ***DisplayName*** **:** Specifies a friendly name that can be used by user-interface programs to identify the service.

password= ***Password*** **:** Specifies a password. This is required if an account other than LocalSystem is used.

/?: Displays help at the command prompt.

Top of page

Remarks

- Without a space between a parameter and its value (that is, **type= own**, not **type=own**), the operation will fail.

Top of page

Examples

The following examples show how you can use the **sc create** command:

```
sc \\myserver create NewService binpath= c:\windows\system32\NewServ.exe
sc create NewService binpath= c:\windows\system32\NewServ.exe type= share start= auto depend= "+TDI Netbios"
```

sc delete

Deletes a service subkey from the registry. If the service is running or if another process has an open handle to the service, then the service is marked for deletion.

↑Top of page

Syntax

sc [*ServerName*] **delete** [*ServiceName*]

↑Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- Use Add or Remove Programs to delete DHCP, DNS, or any other built-in operating system services. Add or Remove Programs will not only remove the registry subkey for the service, but it will also uninstall the service and delete any shortcuts to the service.

↑Top of page

Examples

The following example shows how you can use the **sc delete** command:

sc delete newserv

sc description

Sets the description string for a service.

Top of page

Syntax

sc [*ServerName*] **description** [*ServiceName*] [*Description*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***Description* :** Specifies a description for the specified service. If no string is specified, the description of the service is not modified. There is no limit to the number of characters that can be contained in the service description.

/?: Displays help at the command prompt.

Top of page

Examples

The following example shows how you can use the **sc description** command:

sc description newserv "Runs quality of service control."

sc enumdepend

Lists the services that cannot run unless the specified service is running.

Top of page

Syntax

sc [*ServerName*] **enumdepend** [*ServiceName*] [*BufferSize*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***BufferSize* :** Specifies the size (in bytes) of the enumeration buffer. The default is 1024 bytes.

/?: Displays help at the command prompt.

Top of page

Remarks

- If the buffer is not big enough, the **enumdepend** operation will output dependencies only partially, and will specify the additional buffer size required to output all dependencies. If the output is truncated, rerun the operation and specify the larger buffer size.

Top of page

Examples

The following examples show how you can use the **sc enumdepend** command:

sc enumdepend rpcss 5690
sc enumdepend tapisrv

sc failure

Specifies what action to take upon failure of the service.

Top of page

Syntax

sc [*ServerName*] **failure** [*ServiceName*] [**reset=** *ErrorFreePeriod*] [**reboot=** *BroadcastMessage*] [**command=** *CommandLine*] [**actions=** *FailureActionsAndDelayTime*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

reset= ***ErrorFreePeriod* :** Specifies the length of the period (in seconds) with no failures after which the failure count should be reset to 0. This parameter must be used in conjunction with the **actions=** parameter.

reboot= ***BroadcastMessage* :** Specifies the message to be broadcast upon failure of the service.

command= ***CommandLine* :** Specifies the command line to be run upon failure of the service. For more information about how to run a batch or VBS file upon failure, see Remarks.

actions= ***FailureActionsAndDelayTime* :** Specifies the failure actions and their delay time (in milliseconds) separated by the forward slash (/). The following actions are valid: **run**, **restart**, and **reboot**. This parameter must be used in conjunction with the **reset=** parameter. Use **actions= ""** to take no action upon failure.

/?: Displays help at the command prompt.

Top of page

Remarks

- Not all services allow changes to their failure options. Some run as part of a service set.
- To run a batch file upon failure, specify *cmd***.exe** *Drive***:\***FileName***.bat** to the **command=** parameter, where *Drive***:\***FileName***.bat** is the fully qualified name of the batch file.
- To run a VBS file upon failure, specify *cscript drive***:\***myscript***.vbs** to the **command=** parameter, where *drive***:\***myscript***.vbs** is the fully qualified name of the script file.
- It is possible to specify three different actions to the **actions=** parameter, which will be used the first, second, and third time a service fails.
- Without a space between a parameter and its value (that is, **type= own**, not **type=own**), the operation will fail.

Top of page

Examples

The following examples show how you can use the **sc failure** command:

```
sc failure msftpsvc reset= 30 actions= restart/5000
sc failure dfs reset= 60 command= c:\windows\services\restart_dfs.exe actions= run/5000
sc failure dfs reset= 60 actions= reboot/30000
sc failure dfs reset= 60 reboot= "The Distributed File System service has failed. Because of this, the computer will reboot in 30 seconds." actions= reboot/30000
sc failure myservice reset= 3600 reboot= "MyService crashed -- rebooting machine" command= "%windir%\MyServiceRecovery.exe" actions= restart/5000/run/10000/reboot/60000
```

sc getdisplayname

Gets the display name associated with a particular service.

Top of page

Syntax

sc [*ServerName*] **getdisplayname** [*ServiceName*] [*BufferSize*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***BufferSize* :** Specifies the size (in bytes) of the buffer. The default is 1024 bytes.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **sc getdisplayname** command:

sc getdisplayname clipsrv
sc getdisplayname tapisrv
sc getdisplayname sharedaccess

sc getkeyname

Gets the key name associated with a particular service, using the display name as input.

Top of page

Syntax

sc [*ServerName*] **getkeyname** [*ServiceDisplayName*] [*BufferSize*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceDisplayName* :** Specifies the display name of the service.

***BufferSize* :** Specifies the size (in bytes) of the buffer. The default is 1024 bytes.

/?: Displays help at the command prompt.

Top of page

Remarks

- If the *ServiceDisplayName* contains spaces, use quotation marks around the text (that is, **"***Service Display Name***"**).

Top of page

Examples

The following examples show how you can use the **sc getkeyname** command:

sc getkeyname "remote procedure call (rpc)"

sc getkeyname "internet connection sharing"
sc getkeyname clipbook

sc interrogate

Sends an INTERROGATE control request to a service.

⇡Top of page

Syntax

sc [*ServerName*] **interrogate** [*ServiceName*]

⇡Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Sending INTERROGATE to a service causes the service to update its status with the Service Control Manager.

⇡Top of page

Examples

The following examples show how you can use the **sc interrogate** command:

sc interrogate sharedaccess
sc interrogate rpcss

sc lock

Locks the Service Control Manager's database.

Top of page

Syntax

sc [*ServerName*] **lock**

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

/?: Displays help at the command prompt.

Top of page

Remarks

- Locking the Service Control Manager's database prevents any services from starting. Use this if you want to make sure that a service will not be started after it has been stopped. This will allow you to take some action (for example, deleting the service) without interference.
- Using the **lock** operation locks the Service Control Manager's database and then allows the database to be unlocked by typing **u**. You can also kill the process from which you locked the database.

Top of page

Examples

The following example shows how you can use the **sc lock** command:

sc lock

sc pause

Sends a PAUSE control request to a service.

Top of page

Syntax

sc [*ServerName*] **pause** [*ServiceName*]

⇧Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

/?: Displays help at the command prompt.

⇧Top of page

Remarks

- Use the **pause** operation to pause a service before shutting it down.
- Not all services can be paused.
- Not all services perform the same when paused. Some continue to service existing clients, but refuse to accept new clients. Others cease to service existing clients and also refuse to accept new ones.

⇧Top of page

Examples

The following example shows how you can use the **sc pause** command:

sc pause tapisrv

sc qc

Queries the configuration information for a service.

⇧Top of page

Syntax

sc [*ServerName*] **qc** [*ServiceName*] [*BufferSize*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***BufferSize* :** Specifies the size (in bytes) of the buffer. The default is 1024 bytes.

/?: Displays help at the command prompt.

Top of page

Remarks

- The **qc** operation displays the following information about a service: SERVICE_NAME (service's registry subkey name), TYPE, ERROR_CONTROL, BINARY_PATH_NAME, LOAD_ORDER_GROUP, TAG, DISPLAY_NAME, DEPENDENCIES, and SERVICE_START_NAME.
- Administrators can use SC to determine the binary name of any service and find out if it shares a process with other services by typing the following at the command line:

  sc qc *ServiceName*

  SC can help match up services in the Services node of Microsoft Management Console (MMC) with processes in System Monitor. If the binary name is Services.exe, then the service shares the Service Controller process.

  Services.exe starts all services. To conserve system resources, several Win32 services developed for Windows are written to share the Services.exe process. These services are not listed as separate processes in System Monitor or Task Manager. The same is true of Svchost.exe which is a service host process that many operating services share.

  There might not be a process for every Win32 service because third-party Win32 services can also be configured to share processes. SC can be used to get configuration information on these services. If a service does not share its process with other services, however, there will be a process for it in System Monitor when the service is running.

- SC can be useful for developers of services because it provides more detailed and accurate information about services than Services.exe, which is included with Windows. Services.exe can determine whether a service is running, stopped, or paused. Although these tools are adequate for a debugged application that is running smoothly, the information they provide about a service being developed can be misleading. For example, a service that is starting is shown as started whether it is actually running or not.

  SC implements calls to all Windows service control application programming interface (API) functions. Set the parameters to these functions by specifying them at the command line.

  Using SC, you can query the service status and retrieve the values stored in the status structure fields. Services.exe cannot provide you with the complete status of a service, but SC shows the exact state of the service, as well as the last checkpoint number and wait hint. You can use the checkpoint as a debugging tool because it indicates how far the initialization progressed before the program stopped responding. SC also lets you specify the name of a remote computer so that you can call the service API functions or view the service status structures on a remote computer.

Top of page

Examples

The following examples show how you can use the **sc qc** command:

sc qc \\myserver newsrvice
sc qc rpcss 248

sc qdescription

Displays the description string of a service.

Top of page

Syntax

sc [*ServerName*] **qdescription** [*ServiceName*] [*BufferSize*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***BufferSize* :** Specifies the size (in bytes) of the buffer. The default is 1024 bytes.

/?: Displays help at the command prompt.

Top of page

Examples

The following examples show how you can use the **sc qdescription** command:

sc qdescription rpcss
sc qdescription rpcss 138

sc qfailure

Displays the actions that will be performed if the specified service fails.

Top of page

Syntax

sc [*ServerName*] **qfailure** [*ServiceName*] [*BufferSize*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***BufferSize* :** Specifies the size (in bytes) of the buffer. The default is 1024 bytes.

/?: Displays help at the command prompt.

Top of page

Remarks

- The **qfailure** operation displays the following information about a service: SERVICE_NAME (service's registry subkey name), RESET_PERIOD, REBOOT_MESSAGE, COMMAND_LINE, and FAILURE_ACTIONS.

Top of page

Examples

The following examples show how you can use the **sc qfailure** command:

sc qfailure rpcss
sc qfailure rpcss 20

sc query

Obtains and displays information about the specified service, driver, type of service, or type of driver.

Top of page

Syntax

sc [*ServerName*] **query** [*ServiceName*] [**type=** {**driver**|**service**|**all**}] [**type=** {**own**|**share**|**interact**|**kernel**|**filesys**|**rec**|**adapt**}] [**state=** {**active**|**inactive**|**all**}] [**bufsize=** *BufferSize*] [**ri=** *ResumeIndex*] [**group=** *GroupName*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation. This **query** parameter is not used in conjunction with other **query** parameters (other than *ServerName*).

type= {driver|service|all} : Specifies what to enumerate. The default type is **service**.

Value	Description
driver	Specifies that only drivers are enumerated.
service	Specifies that only services are enumerated.

all	Specifies that both drivers and services are enumerated.

type= {own|share|interact|kernel|filesys|rec|adapt} : Specifies the type of services or type of drivers to enumerate.

Value	Description
own	The service runs in its own process. It does not share an executable file with other services. This is the default.
share	The service runs as a shared process. It shares an executable file with other services.
interact	The service can interact with the desktop, receiving input from users. Interactive services must be run under the LocalSystem account.
kernel	Driver.
filesys	File system driver.

state= {active|inactive|all} : Specifies the started state of the service for which to enumerate. The default state is **active**.

Value	Description
active	Specifies all active services.
inactive	Specifies all paused or stopped services.
all	Specifies all services.

bufsize= ***BufferSize*** **:** Specifies the size (in bytes) of the enumeration buffer. The default size is 1024 bytes. Increase the size of the enumeration buffer when the display resulting from a query exceeds 1024 bytes.

ri= ***ResumeIndex*** **:** Specifies the index number at which to begin or resume the enumeration. The default is 0. Use this parameter in conjunction with the **bufsize=** parameter when more information is returned by a query than the default buffer can display.

group= ***GroupName*** **:** Specifies the service group to enumerate. The default is all groups.

/?: Displays help at the command prompt.

Top of page

Remarks

•Without a space between a parameter and its value (that is, **type= own**, not **type=own**), the operation will fail.

- The **query** operation displays the following information about a service: SERVICE_NAME (service's registry subkey name), TYPE, STATE (as well as states which are not available), WIN32_EXIT_B, SERVICE_EXIT_B, CHECKPOINT, and WAIT_HINT.
- The **type=** parameter can be used twice in some cases. The first appearance of the **type=** parameter specifies whether to query services, drivers, or all. The second appearance of the **type=** parameter specifies a type from the **create** operation to further narrow a query's scope.
- When the display resulting from a **query** command exceeds the size of the enumeration buffer, a message similar to the following is displayed:

  Enum: more data, need 1822 bytes start resume at index 79

  To display the remaining **query** information, rerun **query**, setting **bufsize=** to be the number of bytes and **ri=** to the specified index. For example, the remaining output would be displayed by typing the following at the command line:

  sc query bufsize= 1822 ri= 79

Top of page

Examples

The following examples show how you can use the **sc query** command:

sc query
sc query messenger
sc query type= driver
sc query type= service
sc query state= all
sc query bufsize= 50
sc query ri= 14
sc query type= service type= interact
sc query type= driver group= ndis

sc queryex

Obtains and displays extended information about the specified service, driver, type of service, or type of driver.

Top of page

Syntax

sc [*ServerName*] **queryex** [**type=** {**driver**|**service**|**all**}] [**type=** {**own**|**share**|**interact**|**kernel**|**filesys**|**rec**|**adapt**}] [**state=** {**active**|**inactive**|**all**}] [**bufsize=** *BufferSize*] [**ri=** *ResumeIndex*] [**group=** *GroupName*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation. This **queryex** parameter is not used in conjunction with any other **queryex** parameters except *ServerName*.

type= {driver|service|all} : Specifies what to enumerate. The default type is **service**.

Value	Description
driver	Specifies that only drivers are enumerated.
service	Specifies that only services are enumerated.
all	Specifies that both drivers and services are enumerated.

type= {own|share|interact|kernel|filesys|rec|adapt} : Specifies the type of services or type of drivers to enumerate.

Value	Description
own	The service runs in its own process. It does not share an executable file with other services. This is the default.
share	The service runs as a shared process. It shares an executable file with other services.
interact	The service can interact with the desktop, receiving input from users. Interactive services must be run under the LocalSystem account.
kernel	Driver.
filesys	File system driver.

state= {active|inactive|all} : Specifies the started state of the service for which to enumerate. The default state is **active**.

Value	Description
active	Specifies all active services.
inactive	Specifies all paused or stopped services.
all	Specifies all services.

bufsize= ***BufferSize*** **:** Specifies the size (in bytes) of the enumeration buffer. The default size is 1024 bytes.

ri= ***ResumeIndex*** **:** Specifies the index number at which to begin or resume the enumeration. The default is 0.

group= ***GroupName*** **:** Specifies the service group to enumerate. The default is all groups.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Without a space between a parameter and its value (that is, **type= own**, not **type=own**), the operation will fail.
- The **queryex** operation displays the following information about a service: SERVICE_NAME (service's registry subkey name), TYPE, STATE (as well as states that are not available), WIN32_EXIT_B, SERVICE_EXIT_B, CHECKPOINT, WAIT_HINT, PID, and FLAGS.
- The **type=** parameter can be used twice in some cases. The first appearance of the **type=** parameter specifies whether to query services, drivers, or all. The second appearance of the **type=** parameter specifies a type from the **create** operation to further narrow a query's scope.
- When the display resulting from a **queryex** command exceeds the size of the enumeration buffer, a message similar to the following is displayed:

  Enum: more data, need 2130 bytes start resume at index 75

  To display the remaining **queryex** information, rerun **queryex**, setting **bufsize=** to be the number of bytes and **ri=** to the specified index. For example, the remaining output would be displayed by typing the following at the command line:

  sc queryex bufsize= 2130 ri= 75

⇡Top of page

Examples

The following examples show how you can use the **sc queryex** command:

sc queryex messenger
sc queryex group= ""

sc querylock

Queries and displays the lock status for the Service Control Manager's database.

⇡Top of page

Syntax

sc [*ServerName*] **querylock**

⇡Top of page

Parameter

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

/?: Displays help at the command prompt.

sc sdset

Sets a service's security descriptor using Service Descriptor Definition Language (SDDL).

⇡Top of page

Syntax

sc [*ServerName*] **sdset** *ServiceName ServiceSecurityDescriptor*

⇡Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***ServiceSecurityDescriptor* :** Specifies the service descriptor in SDDL.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- For more information about SDDL, see "Security Descriptor Definition Language" at the MSDN Online Library.

sc sdshow

Displays a service's security descriptor using SDDL.

Top of page

Syntax

sc [*ServerName*] **sdshow** *ServiceName*

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

/?: Displays help at the command prompt.

Top of page

Remarks

- For more information about SDDL, see "Security Descriptor Definition Language" at the MSDN Online Library.

Top of page

Examples

sc sdshow rpcss

sc start

Starts a service running.

Top of page

Syntax

sc [*ServerName*] **start** *ServiceName* [*ServiceArguments*]

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

***ServiceArguments* :** Specifies service arguments to pass to the service to be started.

/?: Displays help at the command prompt.

Top of page

Examples

The following example shows how you can use the **sc start** command:

sc start tapisrv

sc stop

Sends a STOP control request to a service.

Top of page

Syntax

sc [*ServerName*] **stop** *ServiceName*

Top of page

Parameters

***ServerName* :** Specifies the name of the remote server on which the service is located. The name must use the UNC format ("\\myserver"). To run SC.exe locally, ignore this parameter.

***ServiceName* :** Specifies the service name returned by the **getkeyname** operation.

/?: Displays help at the command prompt.

Top of page

Remarks

- Not all services can be stopped.

Top of page

Examples

The following example shows how you can use the **sc stop** command:

sc stop tapisrv

Top of page

Remarks

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Schtasks

Schedules commands and programs to run periodically or at a specific time. Adds and removes tasks from the schedule, starts and stops tasks on demand, and displays and changes scheduled tasks.

To view the command syntax, click the following command:

schtasks create

Creates a new scheduled task.

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc** *schedule* [**/mo** *modifier*] [**/d** *day*] [**/m** *month*[**,** *month*...] [**/i** *IdleTime*] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]] **/?**

Top of page

Parameters

/tn *TaskName* : Specifies a name for the task.

/tr *TaskRun* : Specifies the program or command that the task runs. Type the fully qualified path and file name of an executable file, script file, or batch file. If you omit the path, SchTasks.exe assumes that the file is in the *Systemroot*\System32 directory.

/sc *schedule* : Specifies the schedule type. Valid values are MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY, ONCE, ONSTART, ONLOGON, ONIDLE.

Value	Description
MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY	Specifies the time unit for the schedule.

ONCE	The task runs once at a specified date and time.
ONSTART	The task runs every time the system starts. You can specify a start date, or run the task the next time the system starts.
ONLOGON	The task runs whenever a user (any user) logs on. You can specify a date, or run the task the next time the user logs on.
ONIDLE	The task runs whenever the system is idle for a specified period of time. You can specify a date, or run the task the next time the system is idle.

/mo *modifier* : Specifies how often the task runs within its schedule type. This parameter is required for a MONTHLY schedule. This parameter is valid, but optional, for a MINUTE, HOURLY, DAILY, or WEEKLY schedule. The default value is 1.

Schedule type	Modifier	Description
MINUTE	**1 - 1439**	The task runs every *n* minutes.
HOURLY	**1 - 23**	The task runs every *n* hours.
DAILY	**1 - 365**	The task runs every *n* days.
WEEKLY	**1 - 52**	The task runs every *n* weeks.
MONTHLY	**1 - 12**	The task runs every *n* months.
LASTDAY	The task runs on the last day of the month.	
FIRST, **SECOND**, **THIRD**, **FOURTH**, **LAST**	Use with the **/d** *day* parameter to run a task on a particular week and day. For example, on the third Wednesday of the month.	

/d *day* : Specifies a day of the week or a day of a month. Valid only with a WEEKLY or

MONTHLY schedule.

Schedule type	Day values
WEEKLY	Optional. Valid values are MON - SUN and * (every day). MON is the default.
MONTHLY	A value of MON - SUN is required when the FIRST, SECOND, THIRD, FOURTH, or LAST modifier (**/mo**) is used. A value of 1 - 31 is optional and is valid only with no modifier or a modifier of the 1 - 12 type. The default is day 1 (the first day of the month).

/m *month*[,*month*...] : Specifies a month of the year. Valid values are JAN - DEC and * (every month). The **/m** parameter is valid only with a MONTHLY schedule. It is required when the LASTDAY modifier is used. Otherwise, it is optional and the default value is * (every month).

/i *IdleTime* : Specifies how many minutes the computer is idle before the task starts. Type a whole number from 1 to 999. This parameter is valid only with an ONIDLE schedule, and then it is required.

/st *StartTime* : Specifies the time of day that the task starts in HH:MM:SS 24-hour format. The default value is the current local time when the command completes. The **/st** parameter is valid with MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY, and ONCE schedules. It is required with a ONCE schedule.

/sd *StartDate* : Specifies the date that the task starts in *MM*/*DD*/*YYYY* format. The default value is the current date. The **/sd** parameter is valid with all schedules, and is required for a ONCE schedule.

/ed *EndDate* : Specifies the last date that the task is scheduled to run. This parameter is optional. It is not valid in a ONCE, ONSTART, ONLOGON, or ONIDLE schedule. By default, schedules have no ending date.

/s *Computer* : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [*domain*\]*user* : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p *password* : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/ru {[*Domain*\]*User* | "System"} : Runs the tasks with the permission of the specified user account. By default, the task runs with the permissions of the user logged on to the computer running SchTasks.

Value	Description
[*Domain*\]*User*	Specifies a user account.
"System" or ""	Specifies the NT Authority\System account, which is used by the operating system.

/rp *Password* : Specifies the password of the user account that is specified in the **/ru** parameter. If you omit this parameter when specifying a user account, SchTasks.exe prompts you for the password and obscures the text you type. Tasks run with with permissions of the NT Authority \System account do not require a password and SchTasks.exe does not prompt for one.

/?: Displays help at the command prompt.

Top of page

Remarks

- Using the **/s**, **/u**, and **/p** command-line options

  The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.
- SchTasks.exe does not verify program file locations or user account passwords. If you do not enter the correct file location or the correct password for the user account, the task is created, but it does not run. Also, if the password for an account changes or expires, and you do not change the password saved in the task, then the task does not run.
- The NT Authority\System account does not have interactive logon rights. Users do not see and cannot interact with programs run with system permissions.
- Each task runs only one program. However, you can create a batch file that starts multiple tasks, and then schedule a task that runs the batch file.
- You can test a task as soon as you create it. Use the **run** operation to test the task and then check the SchedLgU.txt file (*Systemroot*\SchedLgU.txt) for errors.

Top of page

Syntax and examples for each schedule type

To view the command syntax, click a command:

schtasks create minute

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc minute** [**/mo** {1 - 1439}] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a task to run every 20 minutes

The following command schedules a security script, Sec.vbs, to run every 20 minutes. Because the command does not include a starting date or time, the task starts 20 minutes after the command completes, and runs every 20 minutes thereafter whenever the system is running. Notice that the security script source file is located on a remote computer, but that the task is scheduled and executes on the local computer.

schtasks /create /sc minute /mo 20 /tn "Security Script" /tr \\central\data\scripts\sec.vbs

In response, SchTasks.exe displays a message explaining that the task will run with the permissions of the current user and requests the current user's password. When you enter the password, SchTasks.exe obscures the text you type.

```
The task will be created under current logged-in user name.
Please enter the password
************
```

Then, SchTasks.exe displays a message indicating that the task is scheduled:

SUCCESS: The Scheduled Task "Security Script" has successfully been created.

A query shows the task that the command scheduled:

```
TaskName                  Next Run Time            Status
========================= ======================== ==============
Security Script           10:50:00 AM , 4/4/2001
```

schtasks create hourly

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc hourly** [**/mo** {1 - 365}] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a command that runs every hour at five minutes past the hour

The following command schedules the MyApp program to run hourly beginning at five minutes past midnight. Because the **/mo** parameter is omitted, the command uses the default value for the hourly schedule, which is every (1) hour. If this command is issued after 12:05 A.M., the program will not run until the next day.

schtasks /create /sc hourly /st 00:05:00 /tn "My App" /tr c:\apps\myapp.exe

To schedule a command that runs every five hours

The following command schedules the MyApp program to run every five hours beginning on the first day of March 2001. It uses the **/mo** parameter to specify the interval and the **/sd** parameter to specify the start date. Because the command does not specify a start time, the current time is used as the start time.

schtasks /create /sc hourly /mo 5 /sd 03/01/2001 /tn "My App" /tr c:\apps\myapp.exe

schtasks create daily

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc daily** [**/mo** {1 - 365}] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a task that runs every day

The following example schedules the MyApp program to run once a day, every day, at 8:00 A.M.

until December 31, 2001. Because it omits the **/mo** parameter, the default interval of 1 is used to run the command every day.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc daily /st 08:00:00 /ed 12/31/2001

To schedule a task that runs every other day

The following example schedules the MyApp program to run every other day at 1:00 P.M. (13:00) beginning on December 31, 2001. The command uses the **/mo** parameter to specify an interval of two (2) days.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc daily /mo 2 /st 13:00:00 /sd 12/31/2001

schtasks create weekly

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc weekly** [**/d** {*MON* - *SUN* | *}] [**/mo** {1 - 52}] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a task that runs every six weeks

The following command schedules the MyApp program to run on a remote computer every six weeks. The command uses the **/mo** parameter to specify the interval. It also uses the **/s** parameter to specify the remote computer and the **/ru** parameter to schedule the task to run with the permissions of the user's Administrator account. Because the **/rp** parameter is omitted, SchTasks.exe prompts the user for the Administrator account password.

Also, because the command is run remotely, all paths in the command, including the path to MyApp.exe, refer to paths on the remote computer.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc weekly /mo 6 /s Server16 /ru Admin01

To schedule a task that runs every other week on Friday

The following command schedules a task to run every other Friday. It uses the **/mo** parameter to specify the two-week interval and the **/d** parameter to specify the day of the week. To schedule a task that runs every Friday, omit the **/mo** parameter or set it to 1.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc weekly /mo 2 /d FRI

schtasks create monthly

Syntax

General Monthly Schedule Syntax : schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc monthly** [**/mo** {**FIRST** | **SECOND** | **THIRD** | **FOURTH** | **LAST** | **LASTDAY**] [**/d** {**MON - SUN** | **1 - 31**}] [**/m** {**JAN - DEC**[**,JAN - DEC**...] | *****}] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Specific Week Syntax : schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc monthly /mo** {**FIRST** | **SECOND** | **THIRD** | **FOURTH** | **LAST**} **/d** {**MON - SUN**} [**/m** {**JAN - DEC**[**,JAN - DEC**...] | *****}] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Lastday Syntax : schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc monthly /mo LASTDAY /m** {**JAN - DEC**[**,JAN - DEC**...] | *****} [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Specific Date Syntax : schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc monthly /d** {**1 - 31**} [**/m** {**JAN - DEC**[**,JAN - DEC**...] | *****}] [**/st** *StartTime*] [**/sd** *StartDate*] [**/ed** *EndDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Examples

To schedule a task for the first day of every month

The following command schedules the MyApp program to run on the first day of every month. Because the default modifier is none (no modifier), the default day is day 1, and the default month is every month, the command does not need any additional parameters.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly

To schedule a task for the last day of every month

The following command schedules the MyApp program to run on the last day of every month. It uses the **/mo** parameter to specify the last day of the month and the **/m** parameter with the wildcard character (*) to indicate that the program runs on the last day of every month.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /mo lastday /m *

To schedule a task that runs every three months

The following command schedules the MyApp program to run every three months. It uses the **/mo** parameter to specify the interval.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /mo 3

To schedule a task for the second Sunday of every month

The following command schedules the MyApp program to run on the second Sunday of every month. It uses the **/mo** parameter to specify the second week of the month and the **/d** parameter to specify the day.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /mo SECOND /d SUN

To schedule a task for the 15th days of May and June

The following command schedules the MyApp program to run on May 15 and June 15 at 3:00 PM (15:00). It uses the **/d** parameter to specify the date and the **/m** parameter to specify the months. It also uses the **/st** parameter to specify the start time.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /d 15 /m MAY,JUN /st 15:00:00

schtasks create once

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc once /st** *StartTime* **/sd** *StartDate* [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a task that runs one time

The following command schedules the MyApp program to run at midnight on January 1, 2002. It uses the **/ru** parameter to run the task with the permissions of the user's Administrator account and the **/rp** parameter to provide the password for the Administrator account.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc once /st 00:00:00 /sd 01/01/2002 /ru Admin23 /rp p@ssworD1

schtasks create onstart

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc onstart** [**/sd** *StartDate*] [**/s** *computer* [**/u** [*domain*\] *user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a task that runs every time the system starts

The following command schedules the MyApp program to run every time the system starts, beginning on March 15, 2001:

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc onstart /sd 03/15/2001

schtasks create onlogon

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc onlogon** [**/sd** *StartDate*] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/ru** {[*Domain*\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a task that runs when a user logs on to a remote computer

The following command schedules a batch file to run every time a user (any user) logs on to the remote computer. It uses the **/s** parameter to specify the remote computer. Because the command is remote, all paths in the command, including the path to the batch file, refer to a path on the remote computer.

schtasks /create /tn "Start Web Site" /tr c:\myiis\webstart.bat /sc onlogon /s Server23

schtasks create onidle

Syntax

schtasks /create /tn *TaskName* **/tr** *TaskRun* **/sc onidle /i**_IdleTime_ [**/sd** *StartDate*] [**/s** *computer* [**/u** [*domain*\\]*user* **/p** *password*]] [**/ru** {[*Domain*\\]*User* | **"System"**} [**/rp** *Password*]]

Example

To schedule a task that runs whenever the computer is idle

The following command schedules the MyApp program to run whenever the computer is idle. It uses the required **/i** parameter to specify that the computer must remain idle for ten minutes before the task starts.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc onidle /i 10

⇡Top of page

More examples

To view the example, click the example name:

To create a task that runs with System permissions

The following command schedules the MyApp program to run with permissions of the NT Authority\System account. In this example, the task is scheduled to run on the first day of every month, but you can use any schedule type for a task run with system permissions.

The command uses the **/ru "System"** parameter to specify the system security context. Because system tasks do not use a password, the **/rp** parameter is omitted.

schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /d 1 /ru "System"

In response, SchTasks.exe displays an informational message and a success message. It does not prompt for a password.

```
INFO: The task will be created under user name ("NT AUTHORITY
\SYSTEM").
SUCCESS: The Scheduled task "My App" has successfully been
created.
```

To create a task that runs more than one program

Each task runs only one program. However, you can create a batch file that runs multiple programs and then schedule a task to run the batch file. The following procedure demonstrates this method:

1. Create a batch file that starts the programs you want to run.

   In this example, you create a batch file that starts Event Viewer (Eventvwr.exe) and System Monitor (Perfmon.exe).

   - Open a text editor, such as Notepad.
   - Type the name and fully qualified path to the executable file for each program. In this case, the file includes the following statements.

     ```
     C:\Windows\System32\Eventvwr.exe
     C:\Windows\System32\Perfmon.exe
     ```
   - Save the file as MyApps.bat.
2. Use SchTasks.exe to create a task that runs MyApps.bat.

   The following command creates the Monitor task, which runs whenever anyone logs on. It uses the **/tn** parameter to name the task, and the **/tr** parameter to run MyApps.bat. It uses the **/sc** parameter to indicate the OnLogon schedule type and the **/ru** parameter to specify the Administrator account.

   ```
   schtasks /create /tn Monitor /tr C:\MyApps.bat /sc onlogon /ru Reskit\Administrator
   ```

   As a result of this command, whenever a user logs on to the computer, the task starts both Event Viewer and System Monitor.

schtasks change

Changes one or more of the following properties of a task.

- The program that the task runs (**/tr**).
- The user account under which the task runs (**/ru**).
- The password for the user account (**/rp**).

⇡Top of page

Syntax

schtasks /change /tn *TaskName* [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] [**/tr** *TaskRun*] [**/ru** [*Domain*\]*User* | **"System"**] [**/rp** *Password*]

⇡Top of page

Parameters

/tn *TaskName* : Identifies the task to be changed. Enter the task name.

/s *Computer* : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [*domain*\]*user* : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p *password* : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/tr *TaskRun* : Changes the program that the task runs. Enter the fully qualified path and file name of an executable file, script file, or batch file. If you omit the path, SchTasks.exe assumes that the file is in the *Systemroot*\System32 directory. The specified program replaces the original program run by the task.

/ru [*Domain*\]*User* | "System" : Changes the user account for the task.

Value	Description
[*Domain*\]*User*	Specifies a user account.
"System" or ""	Specifies the NT Authority\System account, which is used by the operating system.

When you change the user account, you must also change the user password. If a command has an **/ru** parameter but not an **/rp** parameter, SchTasks.exe prompts for a new password and obscures the text you type.

Tasks run with with permissions of the NT Authority\System account do not require a password and SchTasks.exe does not prompt for one.

/rp *Password* : Changes the account password for the task. Enter the new password.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Using the **/s**, **/u**, and **/p** command-line options

  The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.
- The **/tn** and **/s** parameters identify the task. The **/tr**, **/ru**, and **/rp** parameters specify properties of the task that you can change.
- A command that uses the **change** operation must change at least one task property.
- The NT Authority\System account does not have interactive logon rights. Users do not see and cannot interact with programs run with system permissions.

⇡Top of page

Examples

To change the program that a task runs

The following command changes the program that the Virus Check task runs from VirusCheck.exe to VirusCheck2.exe. This command uses the **/tn** parameter to identify the task and the **/tr** parameter to specify the new program for the task. (You cannot change the task name.)

schtasks /change /tn "Virus Check" /tr C:\VirusCheck2.exe

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "Virus Check" has
been changed.
```

As a result of this command, the Virus Check task now runs VirusCheck2.exe.

To change the password for a remote task

The following command changes the password of the user account for the RemindMe task on the remote computer, Svr01. The command uses the **/tn** parameter to identify the task and the **/s** parameter to specify the remote computer. It uses the **/rp** parameter to specify the new password, p@ssWord3.

This procedure is required whenever the password for a user account expires or changes. If the password saved in a task is no longer valid, then the task does not run.

```
schtasks /change /tn RemindMe /s Svr01 /rp p@ssWord3
```

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "RemindMe" has been
changed.
```

As a result of this command, the RemindMe task now runs under its original user account, but with a new password.

To change the program and user account for a task

The following command changes the program that a task runs and changes the user account under which the task runs. Essentially, it uses an old schedule for a new task. This command changes the Notepad task, which starts Notepad.exe every morning at 9:00 a.m., to start Internet Explorer instead.

The command uses the **/tn** parameter to identify the task. It uses the **/tr** parameter to change the program that the task runs and the **/ru** parameter to change the user account under which the task runs.

The **/rp** parameter, which provides the password for the user account, is omitted. You must provide a password for the account, but you can use the **/rp** parameter and type the password in clear text, or wait for SchTasks.exe to prompt you for a password, and then enter the password in obscured text.

```
schtasks /change /tn Notepad /tr "c:\program files\Internet Explorer\iexplore.exe" /ru
DomainX\Admin01
```

In response, SchTasks.exe requests the password for the user account. It obscures the text you type, so the password is not visible.

```
Please enter the password for DomainX\Admin01: *********
```

Note that the **/tn** parameter identifies the task and that the **/tr** and **/ru** parameters change the properties of the task. You cannot use another parameter to identify the task and you cannot change the task name.

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "Notepad" has been
changed.
```

As a result of this command, the RemindMe task now runs under its original user account, but with a new password.

To change a program to the System account

The following command changes the SecurityScript task so that it runs with permissions of the NT Authority\System account. It uses the **/ru ""** parameter to indicate the System account.

schtasks /change /tn SecurityScript /ru ""

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "SecurityScript" has
been changed.
```

Because tasks run with System account permissions do not require a password, SchTasks.exe does not prompt for one.

schtasks run

Starts a scheduled task immediately. The **run** operation ignores the schedule, but uses the program file location, user account, and password saved in the task to run the task immediately.

Top of page

Syntax

schtasks /run /tn *TaskName* [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] **/?**

Top of page

Parameters

/tn *TaskName* : Identifies the task. This parameter is required.

/s *Computer* : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [*domain*\\]*user* : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p *password* : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/?: Displays help at the command prompt.

⇧Top of page

Remarks

- Using the **/s**, **/u**, and **/p** command-line options

  The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.
- Use this operation to test your tasks. If a task does not run, check the Task Scheduler Service transaction log, *Systemroot*\SchedLgU.txt, for errors.
- Running a task does not affect the task schedule and does not change the next run time scheduled for the task.
- To run a task remotely, the task must be scheduled on the remote computer. When you run it, the task runs only on the remote computer. To verify that a task is running on a remote computer, use Task Manager or the Task Scheduler transaction log, *Systemroot*\SchedLgU.txt.

⇧Top of page

Examples

To run a task on the local computer

The following command starts the "Security Script" task.

```
schtasks /run /tn "Security Script"
```

In response, SchTasks.exe starts the script associated with the task and displays the following message:

```
SUCCESS: The Scheduled Task "Security Script" is running ......
```

To run a task on a remote computer

The following command starts the Update task on a remote computer, Svr01:

schtasks /run /tn Update /s Svr01

In this case, SchTasks.exe displays the following error message:

```
ERROR: Unable to run the Scheduled Task "Update".
```

To find the cause of the error, look in the Scheduled Tasks transaction log, C:\Windows\SchedLgU.txt on Svr01. In this case, the following entry appears in the log:

```
"Update.job" (update.exe) 3/26/2001 1:15:46 PM ** ERROR **
The attempt to log on to the account associated with the task
failed, therefore, the task did not run.
        The specific error is:
        0x8007052e: Logon failure: unknown user name or bad
password.
        Verify that the task's Run-as name and password are valid
and try again.
```

Apparently, the user name or password in the task is not valid on the system. The following **schtasks /change** command updates the user name and password for the Update task on Svr01:

schtasks /change /tn Update /s Svr01 /ru Administrator /rp PassW@rd3

After the **change** command completes, the **run** command is repeated. This time, the Update.exe program starts and SchTasks.exe displays the following message:

```
SUCCESS: The Scheduled Task "Update" is running......
```

schtasks end

Stops a program started by a task.

⇡Top of page

Syntax

schtasks /end /tn *TaskName* [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]] **/?**

⇡Top of page

Parameters

/tn ***TaskName*** **:** Identifies the task that started the program. This parameter is required.

/s ***Computer*** **:** Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [*domain*\]*user* **:** Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p ***password*** **:** Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/? : Displays help.

⇡Top of page

Remarks

- Using the **/s**, **/u**, and **/p** command-line options

  The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.
- SchTasks.exe ends only the instances of a program started by a scheduled task. To stop other processes, use TaskKill, a tool included in Windows XP Professional. For more information about TaskKill, see Taskkill.

⇡Top of page

Examples

To end a task on a local computer

The following command stops the instance of Notepad.exe that was started by the My Notepad task:

schtasks /end /tn "My Notepad"

In response, SchTasks.exe stops the instance of Notepad.exe that the task started, and it displays the following success message:

```
SUCCESS: The Scheduled Task "My Notepad" has been terminated
successfully.
```

To end a task on a remote computer

The following command stops the instance of Internet Explorer that was started by the InternetOn task on the remote computer, Svr01:

schtasks /end /tn InternetOn /s Svr01

In response, SchTasks.exe stops the instance of Internet Explorer that the task started, and it displays the following success message:

```
SUCCESS: The Scheduled Task "InternetOn" has been terminated
successfully.
```

schtasks delete

Deletes a scheduled task.

⇡Top of page

Syntax

schtasks /delete /tn {*TaskName* | *} [**/f**] [**/s** *computer* [**/u** [*domain*\\]*user* **/p** *password*]] [**/?**]

⇡Top of page

Parameters

/tn {*TaskName* | *} : Identifies the task being deleted. This parameter is required.

Value	Description
TaskName	Deletes the named task.
*	Deletes all scheduled tasks on the computer.

/f : Suppresses the confirmation message. The task is deleted without warning.

/s *Computer* : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [*domain*\\]*user* : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p *password* : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Using the **/s**, **/u**, and **/p** command-line options

  The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.
- The **delete** operation deletes the task from the schedule. It does not delete the program that the task runs or interrupt a running program.
- The **delete *** command deletes all tasks scheduled for the computer, not just the tasks scheduled by the current user.

⇡Top of page

Examples

To delete a task from the schedule of a remote computer

The following command deletes the "Start Mail" task from the schedule of a remote computer. It uses the **/s** parameter to identify the remote computer.

schtasks /delete /tn "Start Mail" /s Svr16

In response, SchTasks.exe displays the following confirmation message. To delete the task, type **y**. To cancel the command, type **n**:

```
WARNING: Are you sure you want to remove the task "Start Mail" (Y/
N )? y
SUCCESS: The Scheduled Task "Start Mail" was successfully deleted.
```

To delete all tasks scheduled for the local computer

The following command deletes all tasks from the schedule of the local computer, including tasks scheduled by other users. It uses the **/tn *** parameter to represent all tasks on the computer and the **/f** parameter to suppress the confirmation message.

schtasks /delete /tn * /f

In response, SchTasks.exe displays the following success messages indicating that the only task scheduled, SecureScript, is deleted.

SUCCESS: The Scheduled Task "SecureScript" was successfully deleted.

schtasks query

Displays all tasks scheduled to run on the computer, including those scheduled by other users.

⇡Top of page

Syntax

schtasks [**/query**] [**/fo** {**TABLE** | **LIST** | **CSV**}] [**/nh**] [**/v**] [**/s** *computer* [**/u** [*domain*\]*user* **/p** *password*]]

⇡Top of page

Parameters

[/query] : The operation name is optional. Typing **schtasks** without any parameters performs a query.

/fo {TABLE | LIST | CSV} : Specifies the output format. **TABLE** is the default.

/nh : Omits column headings from the table display. This parameter is valid with the **TABLE** and **CSV** output formats.

/v : Adds advanced properties of the tasks to the display.

Queries using **/v** should be formatted as **LIST** or **CSV**.

/s *Computer* : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [*domain*\]*user* : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p *password* : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

•Using the **/s**, **/u**, and **/p** command-line options

The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.

•The **query** operation lists all tasks scheduled for the computer, not just the tasks scheduled by the current user.

⇡Top of page

Examples

To display the scheduled tasks on the local computer

The following commands display all tasks scheduled for the local computer. These commands produce the same result and can be used interchangeably.

schtasks

schtasks /query

In response, SchTasks.exe displays the tasks in the default, simple table format, as shown in the following table:

```
TaskName                  Next Run Time            Status
========================= ======================== ==============
Microsoft Outlook         At logon time
SecureScript              14:42:00 PM , 2/4/2001
```

To display advanced properties scheduled tasks

The following command requests a detailed display of the tasks on the local computer. It uses the **/v** parameter to request a detailed (verbose) display and the **/fo LIST** parameter to format the display as a list for easy reading. You can use this command to verify that a task you created has the intended recurrence pattern.

schtasks /query /fo LIST /v

In response, SchTasks.exe displays a detailed property list for all tasks. The following display shows the task list for a task scheduled to run at 4:00 A.M. on the last Friday of every month:

```
HostName:                                  RESKIT01
TaskName:                                  SecureScript
Next Run Time:                             4:00:00 AM , 3/30/2001
Status:                                    Not yet run
Last Run Time:                             Never
Last Result:                               0
Creator:                                   user01
Schedule:                                  At 4:00 AM on the last
Fri of every month, starting 3/24/2001
Task To Run:                               C:\WINDOWS\system32
\notepad.exe
Start In:                                  notepad.exe
Comment:                                   N/A
Scheduled Task State:                      Enabled
Scheduled Type:                            Monthly
```

```
Modifier:                                  Last FRIDAY
Start Time:                                4:00:00 AM
Start Date:                                3/24/2001
End Date:                                  N/A
Days:                                      FRIDAY
Months:                                    JAN,FEB,MAR,APR,MAY,
JUN,JUL,AUG,SEP,OCT,NOV,DEC
Run As User:                               RESKIT\user01
Delete Task If Not Rescheduled:            Enabled
Stop Task If Runs X Hours and X Mins:      72:0
Repeat: Until Time:                        Disabled
Repeat: Duration:                          Disabled
Repeat: Stop If Still Running:             Disabled
Idle: Start Time(For IDLE Scheduled Type): Disabled
Idle: Only Start If Idle for X Minutes:    Disabled
Idle: If Not Idle Retry For X Minutes:     Disabled
Idle: Stop Task If Idle State End:         Disabled
Power Mgmt: No Start On Batteries:         Disabled
Power Mgmt: Stop On Battery Mode:          Disabled
```

To log tasks scheduled for a remote computer

The following command requests a list of tasks scheduled for a remote computer, and adds the tasks to a comma-separated log file on the local computer. You can use this command format to collect and track tasks that are scheduled for multiple computers.

The command uses the **/s** parameter to identify the remote computer, Reskit16, the **/fo** parameter to specify the format and the **/nh** parameter to suppress the column headings. The >> append symbol redirects the output to the task log, p0102.csv, on the local computer, Svr01. Because the command runs on the remote computer, the local computer path must be fully qualified.

schtasks /query /s Reskit16 /fo csv /nh >> **\\svr01\data\tasklogs\p0102.csv**

In response, SchTasks.exe adds the tasks scheduled for the Reskit16 computer to the p0102.csv file on the local computer, Svr01.

Top of page

Remarks

- SchTasks.exe performs operations similar to those in Scheduled Tasks in Control Panel. You can use either tool to create, delete, configure, or display scheduled tasks.
- Typing **schtasks** without any parameters performs a query.
- The user must be a member of the Administrators group on the computer that the command affects.
- To verify that a scheduled task ran or to find out why a scheduled task did not run, see the Task Scheduler service transaction log, *Systemroot*\SchedLgU.txt. This log records attempted runs initiated by all tools that use the service, including Scheduled Tasks and SchTasks.exe.
- On rare occasions, task files become corrupted. Corrupted tasks do not run. When you try to perform an operation on corrupted tasks, SchTasks.exe displays the following error message:

  ```
  ERROR: The data is invalid.
  ```

  You cannot recover corrupted tasks. To restore the task scheduling features of the system, use SchTasks.exe or Scheduled Tasks to delete the tasks from the system and reschedule them.
- SchTasks.exe replaces At.exe, a tool included in previous versions of Windows.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Taskkill

Secedit

Configures and analyzes system security by comparing your current configuration to at least one template.

To view the command syntax, click a command:

secedit /analyze

Syntax

secedit /analyze /db *FileName* [**/cfg** *FileName*] [**/log** *FileName*] [**/quiet**]

⇡Top of page

Parameters

/db *FileName* : Required. Specifies the path and file name of a database that contains the stored configuration against which the analysis will be performed. If *FileName* specifies a new database, the **/cfg** *FileName* command-line option must also be specified.

/cfg *FileName* : Specifies the path and file name for the security template that will be imported into the database for analysis. This command-line option is only valid when used with the **/db** parameter. If this is not specified, the analysis is performed against any configuration already stored in the database.

/log *FileName* : Specifies the path and file name of the log file for the process. If this is not provided, the default log file is used.

/quiet : Suppresses screen and log output. You can still view analysis results by using Security Configuration and Analysis.

secedit /configure

Configures system security by applying a stored template.

Top of page

Syntax

secedit /configure /db *FileName* [**/cfg** *FileName*] [**/overwrite**][**/areas** *area1 area2...*] [**/log** *FileName*] [**/quiet**]

Top of page

Parameters

/db *FileName* : Required. Provides the file name of a database that contains the security template that should be applied.

/cfg *FileName* : Specifies the file name of the security template that will be imported into the database and applied to the system. This command-line option is only valid when used with the **/db** parameter. If this is not specified, the template that is already stored in the database is applied.

/overwrite : Specifies whether the security template in the **/cfg** parameter should overwrite any template or composite template that is stored in the database instead of appending the results to the stored template. This command-line option is only valid when the **/cfg** parameter is also used. If this is not specified, the template in the **/cfg** parameter is appended to the stored template.

/areas *area1 area2...* : Specifies the security areas to be applied to the system. If an area is not specified, all areas are applied to the system. Each area should be separated by a space.

Area name	Description
SECURITYPOLICY	Local policy and domain policy for the system, including account policies, audit policies, and so on.
GROUP_MGMT	Restricted group settings for any groups specified in the security template
USER_RIGHTS	User logon rights and granting of privileges
REGKEYS	Security on local registry keys
FILESTORE	Security on local file storage
SERVICES	Security for all defined services

/log *FileName* : Specifies the file name of the log file for the process. If it is not specified, the default path is used.

/quiet : Suppresses screen and log output.

secedit /export

Exports a stored template from a security database to a security template file.

⇡Top of page

Syntax

secedit /export [**/mergedpolicy**] [**/DB** *FileName*] [**/CFG** *FileName*] [**/areas** *area1 area2...*] [**/log** *FileName*] [**/quiet**]

⇡Top of page

Parameters

/mergedpolicy : Merges and exports domain and local policy security settings.

/db *FileName* : Specifies the database file that contains the template that will be exported. If the name of a database file is not provided, the system policy database is used.

/db *FileName* : Specifies the file name where the template should be saved.

/areas *area1 area2...* : Specifies the security areas to be exported to a template. If an area is not specified, all areas are exported. Each area should be separated by a space.

Area name	Description
SECURITYPOLICY	Specifies local policy and domain policy for the system, including account policies, audit policies, and so on.
GROUP_MGMT	Specifies restricted group settings for any groups specified in the security template.
USER_RIGHTS	Specifies user logon rights and granting of privileges
REGKEYS	Specifies the security on local registry keys

FILESTORE	Specifies the security on local file storage
SERVICES	Specifies security for all defined services

/log *FileName* : Specifies the file name of the log file for the process. If not specified, the default path is used.

/quiet : Suppresses screen and log output.

secedit /validate

Validates the syntax of a security template to be imported into a database for analysis or application to a system.

⇡Top of page

Syntax

secedit /validate *FileName*

⇡Top of page

Parameter

***FileName* :** Specifies the file name of the security template you have created with Security Templates.

⇡Top of page

Remarks

- **secedit /refreshpolicy** has been replaced with **gpupdate**. For information on how to refresh security settings, see gpupdate

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Automating security configuration tasks

Top of page

Manage Your Profile |Legal |Contact Us

Set

Displays, sets, or removes environment variables. Used without parameters, **set** displays the current environment settings.

Syntax

set [[**/a** [*expression*]] [**/p** [*variable*=]] *string*]

Top of page

Parameters

/a : Sets *string* to a numerical expression that is evaluated.

/p : Sets the value of *variable* to a line of input.

***variable* :** Specifies the variable you want to set or modify.

***string* :** Specifies the string you want to associate with the specified variable.

/? : Displays help at the command prompt.

Top of page

Remarks

- Using **set** at the Recovery Console

  The **set** command, with different parameters, is available from the Recovery Console.

- Using special characters

  The characters <, >, |, &, ^ are special command shell characters and must be either preceded by the escape character (^) or enclosed in quotation marks when used in *string* (that is, "*StringContaining&Symbol*". If you use quotation marks to enclose a string containing one of the special characters, the quotation marks are set as part of the environment variable value.
- Using environment variables

  Use environment variables to control the behavior of some batch files and programs and to control the way Windows XP and the MS-DOS subsystem appears and works. The **set** command is often used in the Autoexec.nt file to set environment variables.
- Displaying the current environment settings

  When you type the **set** command alone, the current environment settings are displayed. These settings usually include the COMSPEC and PATH environment variables that are used to help find programs on disk. Two other environment variables used by Windows XP are PROMPT and DIRCMD.
- Using parameters

  When you specify values for variable and string, the specified variable value is added to the environment and the string is associated with that variable. If the 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, the string value associated with the variable is cleared (as if the variable is not there at all).
- Using **/a**

  The following table lists the operators supported for **/a** in descending order of precedence.

Operator	Operation performed
< >	Grouping
* / % + -	Arithmetic
<< >>	Logical shift
&	Bitwise AND
^	Bitwise exclusive OR
|	Bitwise OR

= *= /= %= += -= &= ^= |= <<= >>=	Assignment
,	Expression separator

If you use logical (&& ||) or modulus (%) operators, enclose the expression string in quotation marks. Any non-numeric strings in the expression are considered environment variable names whose values are converted to numbers before being processed. If you specify an environment variable name that is not defined in the current environment, a value of zero is allotted, which allows you to do arithmetic with environment variable values without using the % to retrieve a value.

If you run **set /a** from the command line outside of a command script, it displays the final value of the expression.

Numeric values are decimal numbers unless prefixed by 0× for hexadecimal numbers or 0 for octal numbers. Therefore, 0×12 is the same as 18 is the same as 022. The octal notation can be confusing. For example, 08 and 09 are not valid numbers because 8 and 9 are not valid octal digits.

- Using **/p**

You are not required to include a prompt string.

- Supporting delayed environment variable expansion

Delayed environment variable expansion support has been added. This support is disabled by default, but you can enable or disable it by using **cmd /v**.

- Working with command extensions

When command extensions are enabled (that is, the default) and you run **set** alone, it displays all current environment variables. If you run **set** with a value, it displays the variables that match that value.

- Using **set** in batch files

When creating batch files, you can use **set** to create variables and use them in the same way that you would the numbered variables *%0* through *%9*. You can also use the variables *%0* through *% 9* as input for **set**.

- Calling a **set** variable from a batch file

When you call a variable value from a batch file, 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 typing **%baud%** at the command line.

Top of page

Examples

To set an environment variable named TEST^1, type:

set testVar=test^^1

To set an environment variable named TEST&1, type:

set testVar=test^&1

Set sets the variable value as everything following the equals sign (=). If you type:

set testVar="test^1"

You get the following result:

testVar="test^1"

To set an environment variable named INCLUDE so that the string C:\Inc (the \Inc directory on drive C) is associated with it, type:

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 so that you can display the contents of the directory associated with the INCLUDE environment variable:

dir %include%

When this command is processed, the string C:\Inc replaces **%include%**.

You can also use **set** in a batch program that adds a new directory to the PATH environment variable. For example:

```
@echo off
rem ADDPATH.BAT adds a new directory
rem to the path environment variable.
set path=%1;%path%
set
```

When command extensions are enabled (that is, the default) and you run **set** with a value, it displays the variables that match that value. For example, if you type **set p** at the command prompt, you get the following results:

```
Path=C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 8 Stepping 1, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0801
ProgramFiles=C:\Program Files
PROMPT=$P$G
```

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command shell overview

Cmd

Setlocal

Starts localization of environment variables in a batch file. Localization continues until a matching **endlocal** command is encountered or the end of the batch file is reached.

Syntax

setlocal {**enableextension** | **disableextensions**} {**enabledelayedexpansion** | **disabledelayedexpansion**}

⇧Top of page

Arguments

enableextension : Enables the command extensions until the matching **endlocal** command is encountered, regardless of the setting prior to the **setlocal** command.

disableextensions : Disables the command extensions until the matching **endlocal** command is encountered, regardless of the setting prior to the **setlocal** command.

enabledelayedexpansion : Enables the delayed environment variable expansion until the matching **endlocal** command is encountered, regardless of the setting prior to the **setlocal** command.

disabledelayedexpansion : Disables the delayed environment variable expansion until the matching **endlocal** command is encountered, regardless of the setting prior to the **setlocal** command.

/? : Displays help at the command prompt.

⇧Top of page

Remarks

- Using **setlocal**

  When you use **setlocal** outside of a script or batch file, it has no effect.
- Changing environmental variables

  Use **setlocal** to change environment variables when you run a batch file. Environment changes made after you run **setlocal** are local to the batch file. Cmd.exe restores previous settings when it either encounters an **endlocal** command or reaches the end of the batch file.
- You can have more than one **setlocal** or **endlocal** command in a batch program (that is, nested commands).
- Testing for command extensions in batch files

  The **setlocal** command sets the ERRORLEVEL variable. If you pass either {**enableextension** | **disableextensions**} or {**enabledelayedexpansion** | **disabledelayedexpansion**}, the ERRORLEVEL variable is set to zero (0). Otherwise, it is set to one (1). You can use this in batch scripts to determine whether the extensions are available, for example:

  ```
      verify other 2>nul
      setlocal enableextensions
      if errorlevel 1 echo Unable to enable extensions
  ```

  Because **cmd** does not set the ERRORLEVEL variable when command extensions are disabled, the **verify** command initializes the ERRORLEVEL variable to a nonzero value when you use it with an invalid argument. Also, if you use the **setlocal** command with arguments {**enableextension** | **disableextensions**} or {**enabledelayedexpansion** | **disabledelayedexpansion**} and it does not set the ERRORLEVEL variable to one (1), command extensions are not available.

  For more information about enabling and disabling command extensions, see **cmd** in Related Topics.

⇡Top of page

Examples

You can localize environment variables in a batch file, as follows:

```
  rem *******Begin Comment**************
  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.
```

```
rem *******End Comment**************
    @echo off
    setlocal
    path=g:\programs\superapp;%path%
    call superapp>c:\superapp.out
    endlocal
    start notepad c:\superapp.out
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Set

Endlocal

Verify

Command-line reference A-Z

Top of page

Shift

Changes the position of batch parameters in a batch file.

Syntax

shift

⇡Top of page

Parameters

none

⇡Top of page

Remarks

- Using the **shift** command-line option with command extensions

  When command extensions are enabled (that is, the default), the **shift** command supports the **/n** command-line option, which tells the command to start shifting at the *n*th argument, where *n* can be a value from zero to eight. For example,

  SHIFT /2

  would shift %3 to %2, %4 to %3, and so on, and leave %0 and %1 unaffected.
- How the shift command works

  The **shift** command changes the values of the batch 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 batch parameters

  You can also use the **shift** command to create a batch file that can accept more than 10 batch 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.
- Using %* with **shift**

  Shift has no affect on the %* batch parameter.
- Shifting parameters back

  There is no backward **shift** command. After you carry out the **shift** command, you cannot recover the first batch parameter (%0) that existed before the shift.

Top of page

Examples

The following batch file, Mycopy.bat, shows how to use **shift** with any number of batch parameters. It copies a list of files to a specific directory. The batch parameters are represented by the directory and file name arguments.

```
@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
```

Top of page

Formatting legend

Format	Meaning

Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Cmd

Using batch parameters

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Shutdown

Allows you to shut down or restart a local or remote computer. Used without parameters, **shutdown** will logoff the current user.

Syntax

shutdown [{**-l**|**-s**|**-r**|**-a**}] [**-f**] [**-m** [\\ComputerName]] [**-t** *xx*] [**-c** "*message*"] [**-d**[**u**][**p**]**:***xx***:***yy*]

⇡Top of page

Parameters

-l : Logs off the current user, this is also the defualt. -m *ComputerName* takes precedence.

-s : Shuts down the local computer.

-r : Reboots after shutdown.

-a : Aborts shutdown. Ignores other parameters, except **-l** and *ComputerName*. You can only use **-a** during the time-out period.

-f : Forces running applications to close.

-m [\\*ComputerName*] : Specifies the computer that you want to shut down.

-t *xx* : Sets the timer for system shutdown in *xx* seconds. The default is 20 seconds.

-c "*message*" : Specifies a message to be displayed in the Message area of the System Shutdown window. You can use a maximum of 127 characters. You must enclose the message in quotation marks.

-d [u][p]:*xx*:*yy* : Lists the reason code for the shutdown. The following table lists the different values.

Value	Description
u	Indicates a user code.
p	Indicates a planned shutdown code.
xx	Specifies the major reason code (0-255).
yy	Specifies the minor reason code (0-65536).

/? : Displays help at the command prompt.

Top of page

Remarks

- If you indicate a major and minor reason code, you must first define these reason codes on each computer for which you plan to use the particular reason. If the reason codes are not defined on the target computer, Event Viewer cannot log the correct reason text.

Top of page

Examples

To shut down \\MyServer in 60 seconds, force running applications to close, restart the computer after shutdown, indicate a user code, indicate that the shutdown is planned, log major reason code 125, and log minor reason code 1, type:

```
shutdown -r -f -m \\MyServer -t 60 -d up:125:1
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Shutdown Event Tracker

Command-line reference A-Z

↑Top of page

Start

Starts a separate Command Prompt window to run a specified program or command. Used without parameters, **start** opens a second command prompt window.

Syntax

start [**"***title***"**] [**/d***Path*] [**/i**] [**/min**] [**/max**] [{**/separate** | **/shared**}] [{**/low** | **/normal** | **/high** | **/realtime** | **/abovenormal** | **belownormal**}] [**/wait**] [**/b**] [*FileName*] [*parameters*]

⇡Top of page

Parameters

"*title*" : Specifies the title to display in Command Prompt window title bar.

/d*Path* : Specifies the startup directory.

/i : Passes the Cmd.exe startup environment to the new Command Prompt window.

/min : Starts a new minimized Command Prompt window.

/max : Starts a new maximized Command Prompt window.

/separate : Starts 16-bit programs in a separate memory space.

/shared : Starts 16-bit programs in a shared memory space.

/low : Starts an application in the idle priority class.

/normal : Starts an application in the normal priority class.

/high : Starts an application in the high priority class.

/realtime : Starts an application in the realtime priority class.

/abovenormal : Starts an application in the abovenormal priority class.

/belownormal : Starts an application in the belownormal priority class.

/wait : Starts an application and waits for it to end.

/b : Starts an application without opening a new Command Prompt 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.

⇡Top of page

Remarks

- You can run nonexecutable files through their file association by typing the name of the file as a command. For more information about creating these associations in a command script by using **assoc** and **ftype**, see Related Topics.
- When you run a command that contains a the string "CMD" as the first token without an extension or path qualifier, "CMD" is replaced with the value of the COMSPEC variable. This prevents users from picking up **cmd** from the current directory.
- When you run a 32-bit graphical user interface (GUI) application, **cmd** does not wait for the application to quit before returning to the command prompt. This new behavior does not occur if you run the application from a command script.
- When you run a command that uses a first token that does not contain an extension, Cmd.exe uses the value of the PATHEXT environment variable to determine which extensions to look for and in what order. The default value for the PATHEXT variable is: .COM;.EXE;.BAT;.CMD (that is, the syntax is the same as the PATH variable, with semicolons separating the different elements).
- When you search for an executable and there is no match on any extension, **start** searches directory name. If it does, **start** opens Explorer.exe on that path.

⇡Top of page

Examples

To start the Myapp program at the command prompt and retain use of the current Command Prompt window, type:

start myapp

Assoc

Ftype

Cmd

Command-line reference A-Z

Top of page

Subst

Associates a path with a drive letter. Used without parameters, **subst** displays the names of the virtual drives in effect.

Syntax

subst [*drive1***:** [*drive2***:**]*Path*]

subst *drive1***: /d**

⇧Top of page

Parameters

***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.

/? : Displays help at the command prompt.

⇧Top of page

Remarks

- The following commands do not work, or should not be used, on drives used in the **subst** command:

  - **chkdsk**
  - **diskcomp**
  - **diskcopy**
  - **format**
  - **label**
  - **recover**
- The *drive1* parameter must be within the range specified by the **lastdrive** command. If not, **subst** displays the following error message:

  Invalid parameter - drive1:

⇡Top of page

Examples

To create a virtual drive Z for the path B:\User\Betty\Forms, type:

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 follows:

z:

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Systeminfo

Displays detailed configuration information about a computer and its operating system, including operating system configuration, security information, product ID, and hardware properties, such as RAM, disk space, and network cards.

Syntax

systeminfo[**.exe**] [**/s** *Computer* [**/u** *Domain\User* [**/p** *Password*]]] [**/fo** {**TABLE**|**LIST**|**CSV**}] [**/nh**]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fo {TABLE|LIST|CSV} : Specifies the format to use for the output. Valid values are **TABLE**, **LIST**, and **CSV**. The default format for output is **LIST**.

/nh : Suppresses column headers in the output. Valid when the **/fo** parameter is set to **TABLE** or **CSV**.

/?: Displays help at the command prompt.

⇡Top of page

Examples

The following examples show how you can use the **systeminfo** command:

```
systeminfo.exe /s srvmain
systeminfo.exe /s srvmain /u maindom\hiropln
systeminfo /s srvmain /u maindom\hiropln /p p@ssW23 /fo table
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

System File Checker (sfc)

Scans and verifies the versions of all protected system files after you restart your computer.

Syntax

sfc [**/scannow**] [**/scanonce**] [**/scanboot**] [**/revert**] [**/purgecache**] [**/cachesize=x**]

Top of page

Parameters

/scannow : Scans all protected system files immediately.

/scanonce : Scans all protected system files once.

/scanboot : Scans all protected system files every time the computer is restarted.

/revert : Returns the scan to its default operation.

/purgecache : Purges the Windows File Protection file cache and scans all protected system files immediately.

/cachesize=x : Sets the size, in MB, of the Windows File Protection file cache.

/? : Displays help at the command prompt.

Top of page

Remarks

- You must be logged on as a member of the Administrators group to run **sfc**.
- If **sfc** discovers that a protected file has been overwritten, it retrieves the correct version of the file from the %systemroot%\system32\dllcache folder, and then replaces the incorrect file.

- If the %systemroot%\system32\dllcache folder becomes corrupt or unusable, use **sfc /scannow**, **sfc /scanonce**, or **sfc /scanboot** to repair the contents of the Dllcache directory.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Windows File Protection

Using File Signature Verification

Driver Signing for Windows

Command-line reference A-Z

⇡Top of page

Manage Your Profile |Legal |Contact Us

Taskkill

Ends one or more tasks or processes. Processes can be killed by process ID or image name.

Syntax

taskkill [**/s** *Computer*] [**/u** *Domain\User* [**/p** *Password*]]] [**/fi** *FilterName*] [**/pid** *ProcessID*]|[**/im** *ImageName*] [**/f**][**/t**]

Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fi *FilterName* : Specifies the types of process(es) to include in or exclude from termination. The following are valid filter names, operators, and values.

Name	Operators	Value
Hostname	**eq**, **ne**	Any valid string.
Status	**eq**, **ne**	**RUNNING**|**NOT RESPONDING**
Imagename	**eq**, **ne**	Any valid string.
PID	**eg**, **ne**, **gt**, **lt**, **ge**, **le**	Any valid positive integer.
Session	**eg**, **ne**, **gt**, **lt**, **ge**, **le**	Any valid session number.

CPUTime	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	Valid time in the format of *hh*:*mm*:*ss*. The *mm* and *ss* parameters should be between 0 and 59 and *hh* can be any valid unsigned numeric value.
Memusage	**eg**, **ne**, **gt**, **lt**, **ge**, **le**	Any valid integer.
Username	**eq**, **ne**	Any valid user name ([*Domain*\]*User*).
Services	**eq**, **ne**	Any valid string.
Windowtitle	**eq**, **ne**	Any valid string.

/pid *ProcessID* : Specifies the process ID of the process to be terminated.

/im *ImageName* : Specifies the image name of the process to be terminated. Use the wildcard (*) to specify all image names.

/f : Specifies that process(es) be forcefully terminated. This parameter is ignored for remote processes; all remote processes are forcefully terminated.

/t : Specifies to terminate all child processes along with the parent process, commonly known as a tree kill.

/?: Displays help at the command prompt.

Top of page

Remarks

- The wildcard character (*) is accepted only when specified along with the filters.
- Termination for remote processes will always be done forcefully regardless of whether the **/f** parameter is specified.
- Supplying a computer name to the HOSTNAME filter will cause a shutdown and all processes will be stopped.
- Use **tasklist** to determine the Process ID (PID) for the process to be terminated.
- **Taskkill** is a replacement for the **Kill** tool.

Top of page

Examples

The following examples show how you can use the **taskkill** command:

```
taskkill /pid 1230 /pid 1241 /pid 1253
taskkill /f /fi "USERNAME eq NT AUTHORITY\SYSTEM" /im notepad.exe
taskkill /s srvmain /f /im notepad.exe
taskkill /s srvmain /u maindom\hiropln /p p@ssW23 /fi "IMAGENAME eq note*" /im *
taskkill /s srvmain /u maindom\hiropln /fi "USERNAME ne NT*" /im *
taskkill /f /fi "PID ge 1000" /im *
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Manage Your Profile |Legal |Contact Us

Tasklist

Displays a list of applications and services with their Process ID (PID) for all tasks running on either a local or a remote computer.

Syntax

tasklist[**.exe**] [**/s** *computer*] [**/u** *domain\user* [**/p** *password*]] [**/fo** {**TABLE**|**LIST**|**CSV**}] [**/nh**] [**/fi** *FilterName* [**/fi** *FilterName2* [...]]] [**/m** [*ModuleName*] | **/svc** | **/v**]

⇡Top of page

Parameters

/s *Computer* : Specifies the name or IP address of a remote computer (do not use backslashes). The default is the local computer.

/u *Domain\User* : Runs the command with the account permissions of the user specified by *User* or *Domain\User*. The default is the permissions of the current logged on user on the computer issuing the command.

/p *Password* : Specifies the password of the user account that is specified in the **/u** parameter.

/fo {TABLE|LIST|CSV} : Specifies the format to use for the output. Valid values are **TABLE**, **LIST**, and **CSV**. The default format for output is **TABLE**.

/nh : Suppresses column headers in the output. Valid when the **/fo** parameter is set to **TABLE** or **CSV**.

/fi *FilterName* : Specifies the types of process(es) to include in or exclude from the query. The following table lists valid filter names, operators, and values.

Name	Operators	Value
Status	**eq**, **ne**	**RUNNING**|**NOT RESPONDING**

Imagename	**eq**, **ne**	Any valid string.
PID	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	Any valid positive integer.
Session	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	Any valid session number.
SessionName	**eq**, **ne**	Any valid string.
CPUTime	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	Valid time in the format of *hh*:*mm*:*ss*. The *mm* and *ss* parameters should be between 0 and 59 and *hh* can be any valid unsigned numeric value.
Memusage	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	Any valid integer.
Username	**eq**, **ne**	Any valid user name ([*Domain*\\]*User*).
Services	**eq**, **ne**	Any valid string.
Windowtitle	**eq**, **ne**	Any valid string.
Modules	**eq**, **ne**	Any valid string.

/m [*ModuleName*] : Specifies to show module information for each process. When a module is specified, all the processes using that module are shown. When a module is not specified, all the processes for all the modules are shown. Cannot be used with the **/svc** or the **/v** parameter.

/svc : Lists all the service information for each process without truncation. Valid when the **/fo** parameter is set to **TABLE**. Cannot be used with the **/m** or the **/v** parameter.

/v : Specifies that verbose task information be displayed in the output. Cannot be used with the **/svc** or the **/m** parameter.

/?: Displays help at the command prompt.

↑Top of page

Remarks

- **Tasklist** is a replacement for the **TList** tool.

↑Top of page

Examples

The following examples show how you can use the **tasklist** command:

```
tasklist /v /fi "PID gt 1000" /fo csv
tasklist /fi "USERNAME ne NT AUTHORITY\SYSTEM" /fi "STATUS eq running"
tasklist /v /fi "STATUS eq running"
tasklist /s srvmain /nh
tasklist /s srvmain /s srvny
tasklist /s srvmain /u maindom\hiropln /p p@ssW23 /nh
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

Tcmsetup

Sets up or disables the TAPI client.

Syntax

tcmsetup [**/q**] [**/x**] **/c** *Server1* [*Server2...*]

tcmsetup [**/q**] **/c /d**

⇡Top of page

Parameters

/q : Prevents the display of message boxes.

/x : Specifies that connection-oriented callbacks will be used for heavy traffic networks where packet loss is high. When this parameter is omitted, connectionless callbacks will be used.

/c : Required. Specifies client setup.

***Server1* :** Required. Specifies the name of the remote server that has the TAPI service providers that the client will use. The client will use the service providers' lines and phones. The client must be in the same domain as the server or in a domain that has a two-way trust relationship with the domain that contains the server.

***Server2...* :** Specifies any additional server or servers that will be available to this client. If you specify a list of servers is, use a space to separate the server names.

/d : Clears the list of remote servers. Disables the TAPI client by preventing it from using the TAPI service providers that are on the remote servers.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- In order for TAPI to function correctly, you must run **tcmsetup** to specify the remote servers that will be used by TAPI clients.
- You must be in the Administrators group to run this command. If you are logged on in another group, see Use the runas command to start a program as an administrator
- Before a client user can use a phone or line on a TAPI server, the telephony server administrator must assign the user to the phone or line.
- The list of telephony servers that is created by this command replaces any existing list of telephony servers available to the client. You cannot use this command to add to the existing list.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Specify telephony servers on a client computer

Runas

TCP/IP utilities and services

TCP/IP utilities offer network connections to other computers, such as UNIX workstations. You must have the TCP/IP network protocol installed to use the TCP/IP utilities. For information about starting TCP/IP services from the command prompt, see Net start For information about stopping TCP/IP services from the command prompt, see Net stop

For more information, click a utility or service:

- Arp
- Nslookup
- Finger
- Ping
- Ftp
- Rcp
- Hostname
- Rexec
- Ipconfig
- Route
- Lpq
- Rsh
- Lpr
- Tftp
- Nbtstat
- Tracert
- Netstat

Top of page

Telnet commands

The **telnet** commands allow you to communicate with a remote computer that is using the Telnet protocol. You can run **telnet** without parameters in order to enter the telnet context, indicated by the Telnet prompt (**telnet>**). From the Telnet prompt, use the following commands to manage a computer running Telnet Client.

The **tlntadmn** commands allow you to remotely manage a computer running Telnet Server. These commands are run from the command prompt. Used without parameters, **tlntadmn** displays local server settings.

To use **telnet** commands at the Telnet prompt

To start Telnet Client and to enter the Telnet prompt

Syntax

telnet [\\RemoteServer]

⇡Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server to which you want to connect.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- Used without parameters, **telnet** starts Telnet Client.
- When you are at the Telnet prompt, you must use Telnet commands.

To stop Telnet Client

⇡Top of page

Syntax

quit

⇡Top of page

Parameters

none

⇡Top of page

Remarks

- You can abbreviate this command to **q**.

To connect Telnet Client to a remote computer

⇡Top of page

Syntax

open [\\RemoteServer] [*Port*]

⇡Top of page

Parameters

\\RemoteServer **:** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

Port **:** Specifies the port that you want to use. If you do not specify a port, the default port is assumed.

Remarks

- You can abbreviate this command to **o**.

Examples

To connect to remote server Redmond through port number 44, type:
o redmond 44

To disconnect Telnet Client from a remote computer

Top of page

Syntax

close [\\RemoteServer]

Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

Remarks

- You can abbreviate this command to **c**.

Examples

To disconnect from remote server Redmond, type:
c redmond 44

To set Telnet Client options

Top of page

Syntax

set [\\RemoteServer] [**ntlm**] [**localecho**] [**term** {**ansi** | **vt100** | **vt52** | **vtnt**}] [**escape** *Character*] [**logfile** *FileName*] [**logging**] [**bsasdel**] [**crlf**] [**delasbs**] [**mode** {**console** | **stream**}] [**?**]

⇡Top of page

Parameters

\\*RemoteServer* : Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

ntlm : Turns on NTLM authentication if it is available on the remote server.

localecho : Turns on local echo.

term {ansi | vt100 | vt52 | vtnt} : Sets the terminal to the specified type.

escape *Character* : Sets the escape character. The escape character can be a single character, or it can be a combination of the CTRL key plus a character. To set a control-key combination, hold down CTRL while you type the character that you want to assign.

logfile *FileName* : Sets the file to be used for logging Telnet activity. The log file must be on your local computer. Logging begins automatically when you set this option.

logging : Turns on logging. If no log file is set, an error message appears.

bsasdel : Sets BACKSPACE to be sent as delete.

crlf : Sets the new line mode, which causes the ENTER key to send 0x0D, 0x0A.

delasbs : Sets DELETE to be sent as backspace.

mode {console | stream} : Sets the mode of operation.

? : Allows you to view the complete syntax for this command.

Remarks

- To turn off an option that was previously set, at the Telnet prompt, type:

  unset [*Option*]
- To set the escape character, type:

  e *Character*
- On non-English versions of Telnet, the **codeset** *Option* is available. **Codeset** *Option* sets the current code set to an option, which can be any one of the following: **Shift JIS**, **Japanese EUC**, **JIS Kanji, JIS Kanji (78)**, **DEC Kanji**, **NEC Kanji**. You should set the same code set on the remote computer.

To send Telnet Client commands

⇡Top of page

Syntax

send [\\RemoteServer] [**ao**] [**ayt**] [**esc**] [**ip**] [**synch**] [**?**]

⇡Top of page

Parameters

\\*RemoteServer* : Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

ao : Aborts output command.

ayt : Sends an "Are you there?" command.

esc : Sends the current escape character.

ip : Interrupts the process command.

synch : Performs the Telnet sync operation.

? : Allows you to view the complete syntax for this command.

To view the current settings for the Telnet client

Top of page

Syntax

display

Top of page

Parameters

none

Remarks

- The display command lists the currently operating parameters for the Telnet client. If you are in a Telnet session (in other words, if you are connected to a Telnet server), you can exit the Telnet session to modify the parameters by pressing CTRL+]. To return to the Telnet session, press ENTER.

To use **tlntadmn** commands at the command prompt

To administer a computer running Telnet Server

Syntax

tlntadmn [\\RemoteServer] [**start**] [**stop**] [**pause**] [**continue**]

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

start : Starts Telnet Server.

stop : Stops Telnet Server.

pause : Interrupts Telnet Server.

continue : Resumes Telnet Server.

/?: Displays help at the command prompt.

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

To administer Telnet sessions

⇡Top of page

Syntax

tlntadmn [\\RemoteServer] [**-s**] [**-k**{*SessionID* | **all**}] [**-m** {*SessionID* | **all**} **"***Message***"**]

⇡Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

-s : Displays active Telnet sessions.

-k{*SessionID* | all} : Terminates sessions. Type the session ID to terminate a specific session, or type **all** to terminate all sessions.

-m {*SessionID* | all} "*Message*" : Sends a message to one or more sessions. Type the session ID to send a message to a specific session, or type **all** to send a message to all sessions. Type the message that you want to send between quotation marks (that is, **"***Message***"**).

/?: Displays help at the command prompt.

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

To set logging options on a computer running Telnet Server

Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**auditlocation=**{**eventlog** | **file** | **both**}] [**audit=**[{**+** | **-**}**admin**][{**+** | **-**}**user**][{**+** | **-**}**fail**]]

Top of page

Parameters

\\*RemoteServer* : Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

auditlocation={eventlog | file | both} : Specifies whether to send event information to Event Viewer, to a file, or to both.

audit=[{+ | -}admin][{+ | -}user][{+ | -}fail] : Specifies which events you want to audit (administrative logon events, user logon events, or failed logon attempts). To audit events of a particular type, type a plus sign (+) before that event type. To stop auditing events of a particular type, type a minus sign (-) before that event type.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

- If you specify where to send event information without specifying which type or types of information to audit, only information about administrative logon events will be audited and sent to the location that you specified.

⇡Top of page

Examples

To send event information to Event Viewer, type:

tlntadmn config auditlocation=eventlog

To audit administrative logon events and failed logon attempts, type:

tlntadmn config audit=+admin +fail

To set the default domain on a computer running Telnet Server

⇡Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**dom=***DomainName*]

⇡Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

dom=*DomainName* : Specifies the domain that you want to make the default domain.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

Top of page

Examples

To make Redmond the default domain on your local server, type:

tlntadmn config dom=Redmond

To map the Alt key on a computer running Telnet Server

Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**ctrlakeymap=**{**yes** | **no**}]

Top of page

Parameters

\\RemoteServer **:** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

ctrlakeymap={yes | no} : Specifies whether you want Telnet Server to interpret CTRL+A as ALT. Type **yes** to map the shortcut key, or type **no** to prevent mapping.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.
- If you do not map the ALT key, Telnet Server does not send the ALT key to applications that might rely on that key.

To set the maximum number of connections on a computer running Telnet Server

Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**maxconn=***PositiveInteger*]

Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

maxconn=*PositiveInteger* : Sets the maximum number of connections. You must specify this number with a positive integer that is smaller than 10 million.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

To set the maximum number of failed logon attempts on a computer running Telnet Server

Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**maxfail=***PositiveInteger*]

⇡Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

maxfail=*PositiveInteger* : Sets the maximum number of failed logon attempts that a user is allowed. You must specify this number with a positive integer that is smaller than 100.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

To set the mode of operation on a computer running Telnet Server

⇡Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**mode=**{**console** | **stream**}]

⇡Top of page

Parameters

\\***RemoteServer* :** Specifies the name for the server that you want to manage. If you do not specify a server, the local server is assumed.

mode={console | stream} : Specifies the mode of operation.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

To set the Telnet port on a computer running Telnet Server

Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**port=***IntegerValue*]

Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

port=*IntegerValue* : Sets the Telnet port. You must specify the port with an integer smaller than 1,024.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

To set the methods of authentication on a computer running Telnet Server

Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**sec=**[{**+** | **-**}**ntlm**][{**+** | **-**}**passwd**]]

Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

sec=[{+ | -}ntlm][{+ | -}passwd] : Specifies whether you want to use NTLM, a password, or both to authenticate logon attempts. To use a particular type of authentication, type a plus sign (+) before that type of authentication. To prevent using a particular type of authentication, type a minus sign (-) before that type of authentication.

/?: Displays help at the command prompt.

Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.
- NTLM is the authentication protocol for transactions between two computers where one or both computers is running Windows NT 4.0 or an earlier version. In addition, NTLM is the authentication protocol for computers that are not participating in a domain, such as stand-alone servers and workgroups.

To set the time-out for idle sessions on a computer running Telnet Server

⇡Top of page

Syntax

tlntadmn [\\RemoteServer] **config** [**timeout=***hh***:***mm***:***ss*]

⇡Top of page

Parameters

\\***RemoteServer* :** Specifies the name of the server that you want to manage. If you do not specify a server, the local server is assumed.

timeout=*hh*:*mm*:*ss* : Sets the time-out period in hours, minutes, and seconds.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- You can remotely administer a computer running Telnet Server using the **tlntadmn** commands if both computers are running Windows XP. You can not use the **tlntadmn** commands to remotely administer a computer running Windows 2000 and Telnet Server from a computer that is running Windows XP.

⇡Top of page

Remarks

- To switch from Telnet Client to command mode, at the Telnet prompt, press CTRL+]. To switch back to Telnet Client, press ENTER.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Telnet commands

Session mode and command mode

Authentication

Telnet Client overview

Top of page

Terminal Services commands

For more information, click a command:

- change logon
- change port
- change user
- cprofile
- flattemp
- logoff
- msg
- mstsc
- query process
- query session
- query termserver
- query user
- register
- reset session
- shadow
- tscon
- tsdiscon
- tskill
- tsprof
- tsshutdn

Top of page

Manage Your Profile |Legal |Contact Us

Tftp

Transfers files to and from a remote computer, typically a computer running UNIX, that is running the Trivial File Transfer Protocol (TFTP) service or daemon. Used without parameters, **tftp** displays help.

Syntax

tftp [**-i**] [*Host*] [{**get** | **put**}] [*Source*] [*Destination*]

⇡Top of page

Parameters

-i : Specifies binary image transfer mode (also called octet mode). In binary image mode, the file is transferred in one-byte units. 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 end-of-line (EOL) characters to an appropriate format for the specified computer. Use this mode when transferring text files. If a file transfer is successful, the data transfer rate is displayed.

***Host* :** Specifies the local or remote computer.

put : Transfers the file *Destination* on the local computer to the file *Source* on the remote computer. Because the TFTP protocol does not support user authentication, the user must be logged onto the remote computer, and the files must be writable on the remote computer.

get : Transfers the file *Destination* on the remote computer to the file *Source* on the local computer.

***Source* :** Specifies the file to transfer.

***Destination* :** Specifies where to transfer the file. If *Destination* is omitted, it is assumed to have the same name as *Source*.

/? : Displays help at the command prompt.

Top of page

Remarks

- Using the **get** parameter

  Specify **put** if transferring file *FileTwo* on the local computer to file *FileOne* on remote computer. Specify **get** if transferring file *FileTwo* on the remote computer to file *FileOne* on the remote computer.
- Windows XP or Windows 2000 does not provide a general purpose TFTP server. Windows 2000 provides a TFTP server service only to provide remote boot capabilities to Windows XP and Windows 2000 client computers.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To transfer the file Users.txt from the local computer to the file Users19.txt on a remote computer named vax1, type:

tftp vax1 put users.txt users19.txt

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one

Time

Displays or sets the system time. Used without parameters, **time** displays the system time and prompts you to enter a new time.

Syntax

time [**/t**] [**/time**] [*hours***:**[*minutes*[**:***seconds*[**.***hundredths*]]][{**A**|**P**}]]

⇡Top of page

Parameters

/t : Displays the current system time, without prompting you to enter a new time.

/time : Same as **/t**.

***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.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

•When using **time** without parameters, press ENTER to keep the same time.

- Using the **/t** parameter

  The **/t** command-line option is only available if command extensions are enabled (that is, the default). For more information about enabling and disabling command extensions, see **cmd** in Related Topics.
- Specifying an invalid time format

  If you specify the time in an invalid format the following message is displayed and the system then waits for you to enter the time in the correct format:

  ```
  Invalid time
  Enter new time:_
  ```
- Changing the time format

  You can change the **time** format by changing the setting in **Date and Time** in Control Panel. For just the MS-DOS subsystem, change the **country** setting in your Config.nt file. Depending on the Country/Region selected, time is displayed in the 12-hour format or the 24-hour format. If you are setting the time in the 12-hour format, specify **P** for hours after noon.

↑Top of page

Examples

To set your computer clock to 1:36 P.M., type either of the following commands:

time 13:36

time 1:36P

↑Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown

Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Date

Command-line reference A-Z

Cmd

Top of page

Manage Your Profile |Legal |Contact Us

Title

Creates a title for the command prompt window.

Syntax

title [*string*]

[Top of page](#)

Parameters

***string* :** Specifies the title for the command prompt window.

/? : Displays help at the command prompt.

[Top of page](#)

Remarks

- To use the window title for batch programs, include the **title** command at the beginning of a batch program.
- Once set, the window title can be reset with the **title** command only.

[Top of page](#)

Examples

To set the window title for a batch program, type:

```
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
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Tracerpt

Processes event trace logs or real-time data from instrumented event trace providers and allows you to generate trace analysis reports and **CSV** (comma-delimited) files for the events generated.

Syntax

tracerpt [*FileName* [*filename* ...]] [**-o** [*FileName*]] [**-report** [*FileName*]] [**-rt** *session_name* [*session_name* ...]] [**-summary** [*FileName*]] [**-config** [*FileName*]

⇡Top of page

Parameters

***FileName* [*filename* ...] :** Specifies the name of the file for the event trace session. You can specify multiple files.

-o [*FileName*] : Specifies the name of the **CSV** (comma-delimited) file. If no files are specified, then the default is *dumpfile.csv* and not *summary.txt*.

-report [*FileName*] : Specifies the name of the output report file. Default is *workload.txt*.

-rt *session_name* [*session_name* ...] : Gets data from the realtime data source. To use this option, include the event trace session.

-summary [*FileName*] : Specifies name of output summary file. Default is *summary.txt.*

-config *FileName* : Specifies the pathname of the settings file that contains command line parameters. Use this to enter your command line options into a file.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Viewing *dumpfile.csv* files in Microsoft Excel

  Opening the *dumpfile.csv* format in Microsoft Excel allows you to view events in chronological order. The files include header records followed by comma-delimited text. The header fields are listed below:

Field	Description
TID	Thread identifier
Clock time	Event timestamp
Kernel (ms)	Processor time in kernel mode
User (ms)	Processor time in user mode
User data	Variable piece of header data based on the Managed Object Format (MOF) structure.
IID	Instance ID
PIID	Parent Instance ID

- Using the **-report** option

  To use the report option, you must include a "Windows kernel trace" file, and any other event trace file in your command line. You will receive an error message without this in your command line.

- Using the **-summary** option

  When you use the **-summary** option, the following file is generated:

Files processed	list of files
Total buffers processed	N
Total events processed	N
Total events lost	N
Start time	dd MMM yyyy hh:mm:ss.ttt
End time	dd MMM yyyy hh:mm:ss.ttt
Elapsed time	N sec

- Using the **-config** option

  The contents of the setting file used with the **-config** option should have the following format:

  [*command_option*]

  value

  where *command_option* is a command line option and *value* specifies its value. For example:

  [o]

  output.txt

  [report]

  report.txt

  [summary]

  summary.txt
- For more information about incorporating **Tracerpt** into your Windows Management Instrumentation (WMI) scripts, see Scripting Logs and Monitoring at the Microsoft Windows Resource Kits Web site.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one

Courier font	Code or program output

Logman

Typeperf

Relog

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Tracert

Determines the path taken to a destination by sending Internet Control Message Protocol (ICMP) Echo Request messages to the destination with incrementally increasing Time to Live (TTL) field values. The path displayed is the list of near-side router interfaces of the routers in the path between a source host and a destination. The near-side interface is the interface of the router that is closest to the sending host in the path. Used without parameters, **tracert** displays help.

Syntax

tracert [**-d**] [**-h** *MaximumHops*] [**-j** *HostList*] [**-w** *Timeout*] [*TargetName*]

⇡Top of page

Parameters

-d : Prevents **tracert** from attempting to resolve the IP addresses of intermediate routers to their names. This can speed up the display of **tracert** results.

-h *MaximumHops* : Specifies the maximum number of hops in the path to search for the target (destination). The default is 30 hops.

-j *HostList* : Specifies that Echo Request messages use the Loose Source Route option in the IP header with the set of intermediate destinations specified in *HostList*. With loose source routing, successive intermediate destinations can be separated by one or multiple routers. The maximum number of addresses or names in the host list is 9. The *HostList* is a series of IP addresses (in dotted decimal notation) separated by spaces.

-w *Timeout* : Specifies the amount of time in milliseconds to wait for the ICMP Time Exceeded or Echo Reply message corresponding to a given Echo Request message to be received. If not received within the time-out, an asterisk (*) is displayed. The default time-out is 4000 (4 seconds).

***TargetName* :** Specifies the destination, identified either by IP address or host name.

-? : Displays help at the command prompt.

Top of page

Remarks

- This diagnostic tool determines the path taken to a destination by sending ICMP Echo Request messages with varying Time to Live (TTL) values to the destination. Each router along the path is required to decrement the TTL in an IP packet by at least 1 before forwarding it. Effectively, the TTL is a maximum link counter. When the TTL on a packet reaches 0, the router is expected to return an ICMP Time Exceeded message to the source computer. Tracert determines the path by sending the first Echo Request message with a TTL of 1 and incrementing the TTL by 1 on each subsequent transmission until the target responds or the maximum number of hops is reached. The maximum number of hops is 30 by default and can be specified using the **-h** parameter. The path is determined by examining the ICMP Time Exceeded messages returned by intermediate routers and the Echo Reply message returned by the destination. However, some routers do not return Time Exceeded messages for packets with expired TTL values and are invisible to the tracert command. In this case, a row of asterisks (*) is displayed for that hop.
- To trace a path and provide network latency and packet loss for each router and link in the path, use the **pathping** command.
- This command is available only if the **Internet Protocol (TCP/IP)** protocol is installed as a component in the properties of a network adapter in Network Connections

Top of page

Examples

To trace the path to the host named corp7.microsoft.com, type:

tracert corp7.microsoft.com

To trace the path to the host named corp7.microsoft.com and prevent the resolution of each IP address to its name, type:

tracert -d corp7.microsoft.com

To trace the path to the host named corp7.microsoft.com and use the loose source route 10.12.0.1-10.29.3.1-10.1.44.1, type:

tracert -j 10.12.0.1 10.29.3.1 10.1.44.1 corp7.microsoft.com

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Pathping

Top of page

Manage Your Profile |Legal |Contact Us

Tree

Graphically displays the directory structure of a path or of the disk in a drive.

Syntax

tree [*Drive***:**][*Path*] [**/f**] [**/a**]

⇡Top of page

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.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- The structure displayed by **tree** 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.

⇡Top of page

Examples

To display the names of all the subdirectories on the disk in your current drive, type:

```
tree \
```

To display, one screen at a time, the files in all the directories on drive C, type:

```
tree c:\ /f | more
```

To print the same list that the previous example displayed, type:

```
tree c:\ /f prn
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Dir

Command-line reference A-Z

Top of page

Type

Displays the contents of a text file. Use the **type** command to view a text file without modifying it.

Syntax

type [*Drive*:][*Path*] *FileName*

↑Top of page

Parameters

[*Drive*:][*Path*] *FileName* : Specifies the location and name of the file or files that you want to view. Separate multiple file names with spaces.

/? : Displays help at the command prompt.

↑Top of page

Remarks

- If you are using an NTFS drive and *FileName* contains spaces, use quotation marks around the text (that is, "*File Name*").
- 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, avoid using the **type** command to display binary files.

↑Top of page

Examples

To display the contents of a file named Holiday.mar, type:

type holiday.mar

To display the contents of a lengthy file one screen at a time, type:

type holiday.mar | more

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Using filters

Using command redirection operators

Command-line reference A-Z

⇡Top of page

Typeperf

Writes performance counter data to the command window, or to a supported log file format. To stop **Typeperf**, press CTRL+C.

Syntax

Typeperf [*Path* [*path* ...]] [**-cf** *FileName*] [**-f** {**csv**|**tsv**|**bin**}] [**-si** *interval*] [**-o** *FileName*] [**-q** [*object*]] [**-qx** [*object*]] [**-sc** *samples*] [**-config** *FileName*] [**-s** *computer_name*]

⇡Top of page

Parameters

-c {*Path* [*path* ...] | -cf *FileName*} : Specifies the performance counter path to log. To list multiple counter paths, separate each command path by a space.

-cf *FileName* : Specifies the file name of the file that contains the counter paths that you want to monitor, one per line.

-f {csv|tsv|bin} : Specifies the output file format. File formats are *csv* (comma-delimited), *tsv* (tab-delimited), and *bin* (binary). Default format is *csv*.

-si *interval* [*mm:*] *ss* : Specifies the time between samples, in the [*mm:*] *ss* format. Default is one second.

-o *FileName* : Specifies the pathname of the output file. Defaults to *stdout*.

-q [*object*] : Displays and queries available counters without instances. To display counters for one object, include the object name.

-qx [*object*] : Displays and queries all available counters with instances. To display counters for one object, include the object name.

-sc *samples* : Specifies the number of samples to collect. Default is to sample until you press CTRL+C.

-config *FileName* : Specifies the pathname of the settings file that contains command line parameters.

-s *computer_name* : Specifies the system to monitor if no server is specified in the counter path.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- Counter path format

The general format for counter paths is as follows: [\\Computer]*object*[*parent*/*instance*#*index*]*counter*] where:

The parent, instance, index, and counter components of the format may contain either a valid name or a wildcard character. The computer, parent, instance, and index components are not necessary for all counters.

You determine the counter paths to use based on the counter itself. For example, the *LogicalDisk* object has an instance *index*, so you must provide the *#index* or a wildcard. Therefore, you could use the following format:

\LogicalDisk(*/*#*)*

In comparison, the *Process* object does not require an instance *index*. Therefore, you could use the following format:

\Process(*)\ID Process

The following is a list of the possible formats:

- \\machine\object(parent/instance#index)\counter
- \\machine\object(parent/instance)\counter
- \\machine\object(instance#index)\counter
- \\machine\object(instance)\counter
- \\machine\object\counter
- \object(parent/instance#index)\counter

- \object(parent/instance)\counter
- \object(instance#index)\counter
- \object(instance)\counter
- \object\counter

If a wildcard character is specified in the parent name, all instances of the specified object that match the specified instance and counter fields will be returned.

If a wildcard character is specified in the instance name, all instances of the specified object and parent object will be returned if all instance names corresponding to the specified index match the wildcard character.

If a wildcard character is specified in the counter name, all counters of the specified object are returned.

Partial counter path string matches (for example, pro*) are not supported.

- Query formats

For counter path queries use this format **Typeperf** [{**-q** | **-qx** [\\Computer\] [*object*] [**-o** *outputfile*}].

Use the following command lines for complete queries:

Typeperf [\\Computer]*object*[*instance*]*counter*]

Typeperf -cf *input file*

- Using the **-config** option

The contents of the setting file used with the **-config** option should have the following format:

[*command_option*]

value

where *command_option* is a command line option and *value* specifies its value. For example:

[-c]

\Windows\mypath

[-o]

report.csv

[-s]

mysystem

- For more information about incorporating **Typeperf** into your Windows Management Instrumentation (WMI) scripts, see Scripting Logs and Monitoring at the Microsoft Windows Resource Kits Web site.

⇡Top of page

Examples

To display processor and memory counters, type:

```
typeperf "\Memory\Available bytes" "\processor(_total)\% processor time"
```

To display Explorer process counters every three seconds in a CSV-output file format, type:

```
typeperf "\Process(Explorer)\Thread Count" -si 3 -o typeperf.csv
```

To display 50 samples of RAS counters on computer Server_name, type:

```
typeperf "\RAS port(LPT1)\Bytes Transmitted" -sc 50 -s Server_name
```

To display counter values for 50 samples of the counters listed in the file input.txt, at sample intervals of 120 seconds, type:

```
typeperf -cf input.txt -si 120 -sc 50 -f TSV -o domain2.tsv
```

To query installed counters with instances, type the following command. Be aware that the **-qx** parameter will return a far greater number of performance counters than the **-q** parameter.

```
typeperf -qx
```

To query physical disk counters without instances on computer Server5 and write them to a file called diskctrs.txt, type:

```
typeperf -q \\Server5\PhysicalDisk -o diskctrs.txt
```

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Logman

Tracerpt

Relog

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Unlodctr

Removes Performance counter names and Explain text for a service or device driver from the system registry.

Syntax

Unlodctr [\\ComputerName] *DriverName*

⇡Top of page

Parameters

[\\*ComputerName*] *DriverName* : Removes the Performance counter name settings and Explain text for driver or service *DriverName* from the Windows XP system registry. If you do not specify the *ComputerName*, Windows XP uses the local computer by default.

/?: Displays help at the command prompt.

⇡Top of page

Remarks

- If the information that you supply contains spaces, use quotation marks around the text (for example, **"***Computer Name***"**).

⇡Top of page

Examples

To remove the current Performance registry settings and counter Explain text for the RSVP service on computer comp1, type:

unlodctr \\comp1 RSVP

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Add counters to System Monitor

Troubleshooting

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Ver

Displays the Windows XP version number.

Syntax

ver

⇡Top of page

Parameters

/? : Displays help at the command prompt.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Verify

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

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

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.

Syntax

vol [*Drive***:**]

⇡Top of page

Parameters

Drive**:** **:** Specifies the drive that contains the disk for which you want to display the volume label and serial number.

/? **:** Displays help at the command prompt.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Format

Label

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Vssadmin

Displays current volume shadow copy backups and all installed shadow copy writers and providers in the command window.

Syntax

Vssadmin list {**shadows** [**/set=** [*shadow copy set GUID*]] | **writers** | **providers**}

⇡Top of page

Parameters

list shadows [/set= [*shadow copy set GUID*]] : Lists all shadow copies on the system, grouped by the Globally Unique Identifier (GUID) for the shadow copy set (a group of shadow copies created at the same time).

list writers : Lists the name, GUID, instance ID, and status of all installed shadow copy writers.

list providers : Lists the name, type, GUID, and version of all installed shadow copy providers.

/?: Displays help at the command prompt.

⇡Top of page

Examples

To display all installed volume shadow copy providers, type the following command:

vssadmin list providers

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Volume shadow copy overview

Command-line reference A-Z

Top of page

W32tm

A tool used to diagnose problems occurring with Windows Time

Syntax

{**/config** [**/computer:***ComputerName*] [[**/update**] [**/manualpeerlist:***ListOfComputerNames*]] [**/syncfromflags:***ListOfFlags*]]|**/monitor**|**/ntte**|**/ntpte**|**/register**|**/resync** [{**:***ComputerName*] [**/nowait**]|[**/rediscover**}]|**/tz**|**/unregister**}

⇧Top of page

Parameters

/config [/computer:*ComputerName*] [[/update] [/manualpeerlist:*ListOfComputerNames*]] [/syncfromflags:*ListOfFlags*] : Adjusts the time settings on the local or target computer. Time synchronization peers can be set with the **/manualpeerlist** switch. Changes to configuration are not used by Windows Time unless the service is restarted or the **/update** switch is used. **/syncfromflags** can be used to set the types of sources used for synchronization, and can be set to either **MANUAL** to use the manual peer list or **DOMHIER** to synchronize from a domain controller.

/monitor : Monitors the target computer or list of computers.

/ntte : Converts an NT system time into a readable format.

/ntpte : Converts an NTP time into a readable format.

/register : Register to run as a service and add default configuration to the registry.

/resync [{:*ComputerName*] [/nowait]|[/rediscover}] : Resynchronize the clock as soon as possible, disregarding all accumulated error statistics. If no computer is specified, the local computer will resynchronize. The command will wait for resynchronization unless the **/nowait** switch is used. Currently used time resources will be used unless **/rediscover** is used, which will force redetection of network resourced before resynchronization.

/tz : Display the current time zone settings.

/unregister : Unregister service and remove all configuration information from the registry.

/? : Displays help at the command prompt.

⇡Top of page

Remarks

- This tool is designed for network administrators to use for diagnosing problems with Windows Time.

  For more information, see **net time** in Related Topics.
- For the Windows Time service to use the changed made with W32tm, it must be notified of the changes. To notify Windows Time, at the command prompt, type **w32tm /config /update**.

⇡Top of page

Examples

To display the current time zone settings, type:

w32tm /tz

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Net time

Top of page

Manage Your Profile |Legal |Contact Us

Winnt

Performs an installation of or upgrade to Windows XP. If you have hardware that is compatible with Windows XP, you can run **winnt** at a Windows 3.*x* or MS-DOS command prompt.

Syntax

winnt [**/s:***SourcePath*] [**/t:***TempDrive*] [**/u:***answer file*][**/udf:***ID* [**,***UDB_file*]] [**/r:***folder*][**/rx:***folder*][**/e:***command*][**/a**]

⇡Top of page

Parameters

/s:*SourcePath* : Specifies the source location of the Windows XP files. The location must be a full path of the form *x***:**\[*Path*] or \\*server**share*[*Path*].

/t:*TempDrive* : Directs Setup to place temporary files on the specified drive and to install Windows XP on that drive. If you do not specify a location, Setup attempts to locate a drive for you.

/u:*answer file* : Performs an unattended Setup using an answer file. The answer file provides answers to some or all of the prompts that the end user normally responds to during Setup. If you use **/u**, you must also use **/s**.

/udf:*ID* [,*UDB_file*] : Indicates an identifier (*ID*) that Setup uses to specify how a Uniqueness Database (UDB) file modifies an answer file (see **/u**). The UDB overrides values in the answer file, and the identifier determines which values in the UDB file are used. If no *UDB_file* is specified, Setup prompts you to insert a disk that contains the $Unique$.udb file.

/r:*folder* : Specifies an optional folder to be installed. The folder remains after Setup finishes.

/rx:*folder* : Specifies an optional folder to be copied. The folder is deleted after Setup finishes.

/e:*command* : Specifies a command to be carried out just before the final phase of Setup.

/a : Enables accessibility options.

/? : Displays help at the command prompt.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command shell overview

Top of page

Manage Your Profile |Legal |Contact Us

Winnt32

Performs an installation of or upgrade to Windows XP. You can run **winnt32** at the command prompt on a computer running Windows 95, Windows 98, Windows Millennium Edition, Windows NT, Windows 2000, or Windows XP.

Note

- If you run **winnt32** on an Itanium-based computer, the command can be run from the Extensible Firmware Interface (EFI) or from Windows XP (not from an earlier operating system). Also, on an Itanium-based computer, **/cmdcons** and **/syspart** are not available, and options relating to upgrades are also not available.

⇡Top of page

Syntax

winnt32 [**/checkupgradeonly**] [**/cmd:***command_line*] [**/cmdcons**] [**/copydir:**{**i386**|**ia64**}*FolderName*] [**/copysource:***FolderName*] [**/debug**[*Level*]**:**[*FileName*]] [**/dudisable**] [**/duprepare:***pathname*] [**/dushare:***pathname*] [**/m:***FolderName*] [**/makelocalsource**] [**/noreboot**] [**/s:***SourcePath*] [**/syspart:***DriveLetter*] [**/tempdrive:***DriveLetter*] [**/udf:***id* [**,***UDB_file*]] [**/unattend**[*num*]**:**[*answer_file*]]

⇡Top of page

Parameters

/checkupgradeonly : Checks your computer for upgrade compatibility with Windows XP.

If you use this option with **/unattend**, no user input is required. Otherwise, the results are displayed on the screen, and you can save them under the file name you specify. The default file name is Upgrade.txt in the *systemroot* folder.

/cmd:*command_line* : Instructs Setup to carry out a specific command before the final phase of Setup. This would occur after your computer has restarted and after Setup has collected the necessary configuration information, but before Setup is complete.

/cmdcons : Installs the Recovery Console as a startup option on a functioning x86-based computer. The Recovery Console is a command-line interface from which you can perform tasks such as starting and stopping services and accessing the local drive (including drives formatted with NTFS). You can only use the **/cmdcons** option after normal Setup is finished.

/copydir:{i386|ia64}*FolderName* : Creates an additional folder within the folder in which the Windows XP files are installed. *Folder_name* refers to a folder that you have created to hold modifications just for your site. For example, for x86-based computers, you could create a folder called Private_drivers within the i386 source folder for your installation, and place driver files in the folder. Then you could type **/copydir:i386\Private_drivers** to have Setup copy that folder to your newly installed computer, making the new folder location *systemroot*\Private_drivers. You can use **/copydir** to create as many additional folders as you want.

/copysource:*FolderName* : Creates a temporary additional folder within the folder in which the Windows XP files are installed. *Folder_name* refers to a folder that you have created to hold modifications just for your site. For example, you could create a folder called Private_drivers within the source folder for your installation, and place driver files in the folder. Then you could type **/copysource:Private_drivers** to have Setup copy that folder to your newly installed computer and use its files during Setup, making the temporary folder location *systemroot* \Private_drivers. You can use **/copysource** to create as many additional folders as you want. Unlike the folders **/copydir** creates, **/copysource** folders are deleted after Setup completes.

/debug[*Level*]:[*FileName*] : Creates a debug log at the level specified, for example, **/debug4:Debug.log**. The default log file is C:\systemroot\Winnt32.log, and the default debug level is 2. The log levels are as follows: 0 represents severe errors, 1 represents errors, 2 represents warnings, 3 represents information, and 4 represents detailed information for debugging. Each level includes the levels below it.

/dudisable : Prevents Dynamic Update from running. Without Dynamic Update, Setup runs only with the original Setup files. This option will disable Dynamic Update even if you use an answer file and specify Dynamic Update options in that file.

/duprepare:*pathname* : Carries out preparations on an installation share so that it can be used with Dynamic Update files that you downloaded from the Windows Update Web site. This share can then be used for installing Windows XP for multiple clients.

/dushare:*pathname* : Specifies a share on which you previously downloaded Dynamic Update files (updated files for use with Setup) from the Windows Update Web site, and on which you previously ran **/duprepare:***pathname*. When run on a client, specifies that the client installation will make use of the updated files on the share specified in *pathname*.

/m:*FolderName* : Specifies that Setup copies replacement files from an alternate location.

Instructs Setup to look in the alternate location first, and if files are present, to use them instead of the files from the default location.

/makelocalsource : Instructs Setup to copy all installation source files to your local hard disk. Use **/makelocalsource** when installing from a CD to provide installation files when the CD is not available later in the installation.

/noreboot : Instructs Setup to not restart the computer after the file copy phase of Setup is completed so that you can run another command.

/s:*SourcePath* : Specifies the source location of the Windows XP files. To simultaneously copy files from multiple servers, type the **/s:***SourcePath* option multiple times (up to a maximum of eight). If you type the option multiple times, the first server specified must be available, or Setup will fail.

/syspart:*DriveLetter* : On an x86-based computer, specifies that you can copy Setup startup files to a hard disk, mark the disk as active, and then install the disk into another computer. When you start that computer, it automatically starts with the next phase of Setup. You must always use the **/tempdrive** parameter with the **/syspart** parameter. You can start **Winnt32** with the **/syspart** option on an x86-based computer running Windows NT 4.0, Windows 2000, or Windows XP. The computer cannot be running Windows 95, Windows 98, or Windows Millennium Edition.

/tempdrive:*DriveLetter* : Directs Setup to place temporary files on the specified partition. For a new installation, Windows XP will also be installed on the specified partition. For an upgrade, the **/tempdrive** option affects the placement of temporary files only; the operating system will be upgraded in the partition from which you run **winnt32**.

/udf:*id* [,*UDB_file*] : Indicates an identifier (*id*) that Setup uses to specify how a Uniqueness Database (UDB) file modifies an answer file (see the **/unattend** entry). The UDB overrides values in the answer file, and the identifier determines which values in the UDB file are used. For example, **/udf:RAS_user,Our_company.udb** overrides settings specified for the RAS_user identifier in the Our_company.udb file. If no *UDB_file* is specified, Setup prompts the user to insert a disk that contains the $Unique$.udb file.

/unattend : Upgrades your previous version of Windows 98, Windows Millennium Edition, Windows NT 4.0, or Windows 2000 in unattended Setup mode. All user settings are taken from the previous installation, so no user intervention is required during Setup.

/unattend[*num*]:[*answer_file*] : Performs a fresh installation in unattended Setup mode. The specified *answer_file* provides Setup with your custom specifications. *Num* is the number of seconds between the time that Setup finishes copying the files and when it restarts your computer. You can use *num* on any computer running Windows 98, Windows Millennium Edition, Windows NT, Windows 2000, or Windows XP.

Using the **/unattend** command-line option to automate Setup affirms that you have read and accepted the Microsoft License Agreement for Windows XP. Before using this command-line option to install Windows XP on behalf of an organization other than your own, you must confirm that the end user (whether an individual, or a single entity) has received, read, and accepted the terms of the Microsoft License Agreement for Windows XP. OEMs may not specify this key on machines being sold to end users.

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Command-line reference A-Z

Command shell overview

Top of page

Using the Windows Management Instrumentation Command-line (WMIC) tool

The Windows Management Instrumentation Command-line (WMIC) is a command-line and scripting interface that simplifies the use of Windows Management Instrumentation (WMI) and systems managed through WMI.

WMIC is based on aliases. Aliases make the primary data provided by WMI available without having to understand WMI-specific concepts. WMI data and many WMI features are also accessible through WMI without aliases.

You can list the available aliases by using WMIC **/?** help. WMIC has a progressive help system. You can use **/?** at any time and at any depth to discover the additional options that are available in the current context. **/?** lists the currently available aliases, commands, and the global switches (that is, switches that apply to WMIC overall). To list the verbs and switches available for an alias, type the name of the alias and **/?**. To list the parameters available for a particular verb, type the name of the alias, the verb, and then **/?**.

For example, the processes running on the current system are available from the PROCESS alias. To view all of the processes that are currently running on the computer, type **PROCESS** in the WMIC utility. To list a specific process, type a command such as **PROCESS WHERE (Description="explorer.exe")**. To receive specific properties for the processes, type a command such as **PROCESS GET Name, Handle, PageFaults**.

Without using aliases, you can use the same options with the **CLASS** command. For example, **CLASS Win32_Process GET Name, Handle, PageFaults**. However, you must determine the name of the class from other sources. To do the equivalent of the alias **Where** clause, you must use **PATH Win32_Process.Description="explorer.exe"**.

For more information about using and extending WMIC, see WMI Command-line in Windows XP Help. This documentation includes information about using WMIC in batch files, for reporting, with remote and multiple computers, and for system management.

Command	Result
/?	Displays help.

CLASS	Escapes from the default alias mode of WMIC to access classes in the WMI schema directly.
PATH	Escapes from the default alias mode of WMIC to access instances in the WMI schema directly.
CONTEXT	Displays the current values of all global switches.
QUIT	Exits WMIC.
EXIT	Exits WMIC.

Top of page

Xcopy

Copies files and directories, including subdirectories.

Syntax

xcopy *Source* [*Destination*] [**/w**] [**/p**] [**/c**] [**/v**] [**/q**] [**/f**] [**/l**] [**/g**] [**/d**[**:***mm-dd-yyyy*]] [**/u**] [**/i**] [**/s** [**/e**]] [**/t**] [**/k**] [**/r**] [**/h**] [{**/a**|**/m**}] [**/n**] [**/o**] [**/x**] [**/exclude:***file1*[+[*file2*]][+[*file3*]] [{**/y**|**/-y**}] [**/z**]

⇡Top of page

Parameters

Source **:** Required. Specifies the location and names of the files you want to copy. This parameter must include either a drive or a path.

Destination **:** Specifies the destination of the files you want to copy. This parameter can include a drive letter and colon, a directory name, a file name, or a combination of these.

/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.

/q : Suppresses the display of **xcopy** messages.

/f : Displays source and destination file names while copying.

/l : Displays a list of files that are to be copied.

/g : Creates decrypted destination files.

/d[:*mm-dd-yyyy*] : Copies source files changed on or after the specified date only. If you do not include a *mm-dd-yyyy* value, **xcopy** copies all *Source* files that are newer than existing *Destination* files. This command-line option allows you to update files that have changed.

/u : Copies files from *Source* that exist on *Destination* only.

/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, **xcopy** copies all specified files into the new directory. By default, **xcopy** prompts you to specify whether *Destination* is a file or a directory.

/s : Copies directories and subdirectories, unless they are empty. If you omit **/s**, **xcopy** works within a single directory.

/e : Copies all subdirectories, even if they are empty. Use **/e** with the **/s** and **/t** command-line options.

/t : Copies the subdirectory structure (that is, the tree) only, not files. To copy empty directories, you must include the **/e** command-line option.

/k : Copies files and retains the read-only attribute on destination files if present on the source files. By default, **xcopy** removes the read-only attribute.

/r : Copies read-only files.

/h : Copies files with hidden and system file attributes. By default, **xcopy** does not copy hidden or system files.

/a : Copies only source files that have their archive file attributes set. **/a** does not modify the archive file attribute of the source file. For information about how to set the archive file attribute by using **attrib**, see Related Topics.

/m : Copies source files that have their archive file attributes set. Unlike **/a**, **/m** turns off archive file attributes in the files that are specified in the source. For information about how to set the archive file attribute by using **attrib**, see Related Topics.

/n : Creates copies by using the NTFS short file or directory names. **/n** is required when you copy files or directories from an NTFS volume to a FAT volume or when the FAT file system naming convention (that is, 8.3 characters) is required on the destination file system. The destination file system can be FAT or NTFS.

/o : Copies file ownership and discretionary access control list (DACL) information.

/x : Copies file audit settings and system access control list (SACL) information (implies **/o**).

/exclude:*filename1*[+[*filename2*]][+[*filename3*]] : Specifies a list of files containing strings.

/y : Suppresses prompting to confirm that you want to overwrite an existing destination file.

/-y : Prompts to confirm that you want to overwrite an existing destination file.

/z : Copies over a network in restartable mode.

/? : Displays help at the command prompt.

Top of page

Remarks

- Using **/v**

  Windows XP does not use this command. It is accepted only for compatibility with MS-DOS files.
- Using **/exclude**

  List each string in a separate line in each file. If any of the listed strings match any part of the absolute path of the file to be copied, that file is then excluded from the copying process. For example, if you specify the string "\Obj\", you exclude all files underneath the Obj directory. If you specify the string ".obj", you exclude all files with the .obj extension.
- Using **/z**

  If you lose your connection during the copy phase (for example, if the server going offline severs the connection), it resumes after you reestablish the connection. **/z** also displays the percentage of the copy operation completed for each file.
- Using **/y** in the COPYCMD environment variable

  You can use **/y** in the COPYCMD environment variable. You can override this command by using **/-y** on the command line. By default, you are prompted to overwrite, unless you run **copy** from within a batch script.

- Copying encrypted files

  Copying encrypted files to a volume that does not support EFS results in an error. Decrypt the files first or copy the files to a volume that does support EFS.
- Appending files

  To append files, specify a single file for destination, but multiple files for source (that is, by using wildcards or file1+file2+file3 format).
- 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 (\), the following message appears:

  ```
  Does destination specify a file name
  or directory name on the target
  (F = file, D = directory)?
  ```

  Press F if you want the file or files to be copied to a file. Press D if you want the file or files to be copied to a directory.

  You can suppress this message by using the **/i** command-line option, which causes **xcopy** to assume that the destination is a directory if the source is more than one file or a directory.
- Using the **xcopy** command to set archive attribute for *Destination* files

  The **xcopy** command creates files with the archive attribute set, whether or not this attribute was set in the source file. For more information about file attributes and **attrib**, see Related Topics.
- Comparing **xcopy** and **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, use the **xcopy** command instead of **diskcopy**. Because the **diskcopy** command copies disks track by track, your source and destination disks must have the same format. The **xcopy** command does not have this requirement. Use **xcopy** unless you need a complete disk image copy.

- Exit codes for **xcopy**

To process exit codes returned by **xcopy**, use the *errorlevel* parameter on the **if** command line in a batch program. For an example of a batch program that processes exit codes using **if**, see Related Topics. The following table lists each exit code and a description.

Exit code	Description
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.

Top of page

Examples

To copy all the files and subdirectories (including any empty subdirectories) from drive A to drive B, type:

```
xcopy a: b: /s /e
```

To include any system or hidden files in the previous example, add the**/h** command-line option as follows:

```
xcopy a: b: /s /e /h
```

To update files in the \Reports directory with the files in the \Rawdata directory that have changed since December 29, 1993, type:

```
xcopy \rawdata \reports /d:12-29-1993
```

To update all the files that exist in \Reports in the previous example, regardless of date, type:

```
xcopy \rawdata \reports /u
```

To obtain a list of the files to be copied by the previous command (that is, without actually copying the files), type:

xcopy \rawdata \reports /d:12-29-1993 /l > xcopy.out

The file Xcopy.out lists every file that is to be copied.

To copy the \Customer directory and all subdirectories to the directory \\Public\Address on network drive H:, retain the read-only attribute, and be prompted when a new file is created on H:, type:

xcopy \customer h:\public\address /s /e /k /p

To issue the previous command, ensure that **xcopy** creates the \Address directory if it does not exist, and suppress the message that appears when you create a new directory, add the **/i** command-line option as follows:

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 if 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:

copyit c:\prgmcode b:

The command interpreter substitutes C:\Prgmcode for **%1** and B: for **%2**, then uses **xcopy** with the **/e** and **/s** command-line options. If **xcopy** encounters an error, the batch program reads the exit code and goes to the label indicated in the appropriate **IF ERRORLEVEL** statement, then displays the appropriate message and exits from the batch program.

⇡Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Copy

Move

Dir

Attrib

Diskcopy

If

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Using batch parameters

You can use batch parameters anywhere within a batch file to extract information about your environment settings.

Cmd.exe provides the batch parameter expansion variables %0 through %9. When you use batch parameters in a batch file, %0 is replaced by the batch file name, and %1 through %9 are replaced by the corresponding arguments that you type at the command line. To access arguments beyond %9, you need to use the **shift** command. For more information about the **shift** command, see Shift
The *%** batch parameter is a wildcard reference to all the arguments, not including %0, that are passed to the batch file.

For example, to copy the contents from Folder1 to Folder2, where %1 is replaced by the value Folder1 and %2 is replaced by the value Folder2, type the following in a batch file called Mybatch.bat:

```
xcopy %1\*.* %2
```

To run the file, type:

```
mybatch.bat C:\folder1 D:\folder2
```

This has the same effect as typing the following in the batch file:

```
xcopy C:\folder1 \*.* D:\folder2
```

You can also use modifiers with batch parameters. Modifiers use current drive and directory information to expand the batch parameter as a partial or complete file or directory name. To use a modifier, type the percent (%) character followed by a tilde (~) character, and then type the appropriate modifier (that is, **%~***modifier*).

The following table lists the modifiers you can use in expansion.

Modifier	Description
%~1	Expands %1 and removes any surrounding quotation marks ("").

%~f1	Expands %1 to a fully qualified path name.
%~d1	Expands %1 to a drive letter.
%~p1	Expands %1 to a path.
%~n1	Expands %1 to a file name.
%~x1	Expands %1 to a file extension.
%~s1	Expanded path contains short names only.
%~a1	Expands %1 to file attributes.
%~t1	Expands %1 to date and time of file.
%~z1	Expands %1 to size of file.
%~$PATH:1	Searches the directories listed in the PATH environment variable and expands %1 to the fully qualified name of the first one found. If the environment variable name is not defined or the file is not found, this modifier expands to the empty string.

The following table lists possible combinations of modifiers and qualifiers that you can use to get compound results.

Modifier	Description
%~dp1	Expands %1 to a drive letter and path.
%~nx1	Expands %1 to a file name and extension.
%~dp$PATH:1	Searches the directories listed in the PATH environment variable for %1 and expands to the drive letter and path of the first one found.
%~ftza1	Expands %1 to a **dir**-like output line.

Note

•In the previous examples, you can replace **%1** and **PATH** with other batch parameter values.

The %* modifier is a unique modifier that represents all arguments passed in a batch file. You cannot use this modifier in combination with the %~ modifier. The %~ syntax must be terminated by a valid argument value.

You cannot manipulate batch parameters in the same manner that you can manipulate environment variables. You cannot search and replace values or examine substrings. However, you can assign the parameter to an environment variable, and then manipulate the environment variable.

Top of page

Manage Your Profile |Legal |Contact Us

Fsutil: behavior

Queries the current settings for generating 8.3 character-length file names, allowing extended characters in 8.3 character-length file names on NTFS volumes, updating the last access timestamp on NTFS volumes, how often quota events are written to the system log, and the size of the master file table (MFT) Zone. Enables or disables the use of 8.3 character-length file names, allowing extended characters in 8.3 character-length file names on NTFS volumes, and updating the last access timestamp on NTFS volumes. Enables you to change how often quota events are written to the system log and to change the amount of disk space reserved for the MFT Zone.

Syntax

fsutil behavior query {**disable8dot3**|**allowextchar**|**disablelastaccess**|**quotanotify**|**mftzone**}

fsutil behavior set [{**disable8dot3** {**1**|**0**}|**allowextchar** {**1**|**0**}|**disablelastaccess** {**1**|**0**}|**quotanotify** *frequency*|**mftzone** *value*}]

Top of page

Parameters

query : Queries the file system behavior parameters.

set : Changes the file system behavior parameters.

disable8dot3 {1|0} : Disables creation of 8.3 character-length file names on FAT- and NTFS-formatted volumes.

allowextchar {1|0} : Determines whether characters from the extended character set, including diacritic characters, can be used in 8.3 short file names on NTFS volumes.

disablelastaccess {1|0} : Determines whether NTFS updates the last access timestamp on each directory when it lists the directories on an NTFS volume.

quotanotify *frequency* : Configures how frequently NTFS quota violations are reported in the system log. Enter a frequency 0 through 4294967295 seconds for how often quota violations are written to the system log. Default is 1 hour (3600 seconds).

mftzone ***value*** **:** The master file table (MFT) Zone is a reserved area that enables the MFT to expand as needed, in order to prevent MFT fragmentation. Set the *value* from 1 (default) to 4 (maximum). The *value* is in 8ths of the disk.

⇡Top of page

Remarks

- The **behavior** subcommand writes changes to the registry, so you must restart the computer for changes to take effect.
- Using **disable8dot3** {**1**|**0**}

  When **disable8dot3** is set to **0**, every time you create a file with a long file name, NTFS creates a second file entry that has an 8.3 character-length file name. When NTFS creates files in a folder, it must look up the 8.3 character-length file names associated with the long file names.

  This parameter updates the **HKLM\SYSTEM\CurrentControlSet\Control\FileSystem \NtfsDisable8dot3NameCreation** registry key.
- Using **allowextchar** {**1**|**0**}

  This parameter updates the **HKLM\SYSTEM\CurrentControlSet\Control\FileSystem \NtfsAllowExtendedCharacterIn8dot3Name** registry key.
- Using **disablelastaccess** {**1**|**0**}

  The **disablelastaccess** parameter is designed to reduce the logging impact of updating the last access timestamp on folders and directories. Disabling the Last Access Time improves the speed of folder and file access.

  Each file and folder on an NTFS volume contains an attribute called Last Access Time. This attribute defines when the file or folder was last accessed, such as when a user lists folders, adds files to a folder, reads a file, or makes changes to a file. The most up-to-date Last Access Time is stored in memory and is eventually written to the disk in two different locations. One is within the file's attribute, which is part of its MFT record. The second is in the index of the directory that contains the file.

  The Last Access Time on disk is not always current. This lag occurs because NTFS delays writing the Last Access Time to disk when users or programs perform read-only operations on a file or folder, such as listing the folder's contents or reading (but not changing) a file in the folder. If the Last Access Time is kept current on disk for read operations, all read operations become write operations, which impacts NTFS performance.

Note that file-based queries of Last Access Time are accurate even if all on-disk values are not current. NTFS returns the correct value on queries because the accurate value is stored in memory.

NTFS typically updates a file's attribute on disk if the current Last Access Time in memory differs by more than an hour from the Last Access Time stored on disk, or when all in-memory references to that file are gone, whichever is more recent. For example, if a file's current Last Access Time is 1:00 P.M., and you read the file at 1:30 P.M., NTFS does not update the Last Access Time. If you read the file again at 2:00 P.M., NTFS updates the Last Access Time in the file's attribute to reflect 2:00 P.M. because the file's attribute shows 1:00 P.M. and the in-memory Last Access Time shows 2:00 P.M.

NTFS updates the index of the directory that contains the file when NTFS updates the file's Last Access Time and detects that the Last Access Time for the file differs by more than an hour from the Last Access Time stored in the directory's index. This update typically occurs after a program closes the handle used to access a file within the directory. If the user holds the handle open for an extended time, a lag occurs before the change appears in the index entry of the directory.

Note that one hour is the maximum time that NTFS defers updating the Last Access Time on disk. If NTFS updates other file attributes such as Last Modify Time, and a Last Access Time update is pending, NTFS updates the Last Access Time along with the other updates without additional performance impact.

Note that using the **disablelastaccess** parameter can affect programs such as backup and Remote Storage that rely on this feature.

This parameter updates the **HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate** registry key.

- Using **mftzone** *value*

The *value* is an approximation of the initial size of the MFT plus the MFT Zone on a new volume and is set at mount time for each file system. As space on the volume is used, NTFS adjusts the space reserved for future MFT growth. If the MFT Zone is already large, the full MFT Zone size is not reserved again. Since the MFT Zone is based on the contiguous range past the end of the MFT, it shrinks as the space is used.

The file system does not redetermine the MFT Zone location until the current MFT Zone is completely used. This never occurs on a typical system.

Fsutil

Fsutil: dirty

Fsutil: file

Fsutil: fsinfo

Fsutil: hardlink

Fsutil: objectid

Fsutil: quota

Fsutil: reparsepoint

Fsutil: sparse

Fsutil: usn

Fsutil: volume

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Fsutil: dirty

Queries to see whether a volume's dirty bit is set. Sets a volume's dirty bit. When a volume's dirty bit is set, **autochk** automatically checks the volume for errors the next time the computer is restarted.

Syntax

fsutil dirty {**query**|**set**} *PathName*

⇡Top of page

Parameters

query : Queries the dirty bit.

set : Sets a volume's dirty bit.

PathName **:** Specifies the drive letter (followed by a colon), mount point, or volume name.

⇡Top of page

Remarks

- If a volume's dirty bit is set, this indicates that the file system may be in an inconsistent state. The dirty bit can be set because the volume is online and has outstanding changes, because changes were made to the volume and the computer shutdown before the changes were committed to disk, or because corruption was detected on the volume. If the dirty bit is set when the computer restarts, **chkdsk** runs to verify the consistency of the volume.

  Every time Windows XP starts, Autochk.exe is called by the Kernel to scan all volumes to check if the volume dirty bit is set. If the dirty bit is set, autochk performs an immediate **chkdsk /f** on that volume. **Chkdsk /f** verifies file system integrity and attempts to fix any problems with the volume.

⇡Top of page

Examples

To query the dirty bit on drive C, type:

fsutil dirty query C:

Sample output:

```
Volume C: is dirty
```

or

```
Volume C: is not dirty
```

To set the dirty bit on drive C, type:

fsutil dirty set C:

Fsutil

Fsutil: behavior

Fsutil: file

Fsutil: fsinfo

Fsutil: hardlink

Fsutil: objectid

Fsutil: quota

Fsutil: reparsepoint

Fsutil: sparse

Fsutil: usn

Fsutil: volume

Command-line reference A-Z

Chkdsk

Chkdsk

Top of page

Manage Your Profile |Legal |Contact Us

Fsutil: file

Typically used by support professionals. Finds a file by security identifier, queries allocated ranges for a file, sets a file's short name, sets a file's valid data length, sets zero data for a file, or creates a new file.

Syntax

fsutil file [**createnew**] *PathName length*

fsutil file [**findbysid**] *User PathName*

fsutil file [**queryallocranges**] **offset=***offset* **length=***length PathName*

fsutil file [**setshortname**] *PathName shortname*

fsutil file [**setvaliddata**] *PathName datalength*

fsutil file [**setzerodata**] **offset=***offset* **length=***length PathName*

⇡Top of page

Parameters

createnew : Creates a file of the specified name and size, whose content consists of zeroes.

***PathName* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

***length* :** Specifies the file's valid data length.

findbysid : Finds files on NTFS volumes that belong to a specified user. The user is identified by their SID (security identifier).

***User* :** Specifies the user's user name or logon name.

***PathName* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

queryallocranges : Queries the allocated ranges for a file on an NTFS volume. Useful for determining whether a file has sparse regions.

offset=*offset* : Specifies the start of the range to set to zeroes.

length=*length* : Specifies the length of the range, in bytes.

***PathName* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

setshortname : Sets the short name (8.3 character-length file name) for a file on an NTFS volume.

***PathName* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

***shortname* :** Specifies the file's shortname.

setvaliddata : Set the valid data length for a file on an NTFS volume.

***PathName* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

***datalength* :** Specifies the length of the file in bytes.

setzerodata : Sets a range (specified by *offset* and *length*) of the file to zeroes, which empties the file. If the file is a sparse file, the underlying allocation units are decommitted.

offset=*offset* : Specifies the file offset, the start of the range to set to zeroes.

length=*length* : Specifies the length of the range to set to zero.

***PathName* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

Top of page

Remarks

•Using **setvaliddata**

In NTFS, there are two important concepts of file length: the End of File (EOF) marker and the Valid Data Length (VDL). The EOF indicates the actual length of the file. The VDL identifies the length of valid data on disk. Any reads between VDL and EOF automatically return 0 in order to preserve the C2 object reuse requirement.

The **setvaliddata** parameter is only available for administrators because it requires the Manage Volume privilege. This feature is only required for advanced multimedia and system area network (SAN) scenarios.

Top of page

Examples

To find user scottb on drive C by his SID, type:

fsutil file findbysid scottb C:\users

To query the allocated ranges for a file on an NTFS volume, type:

fsutil file queryallocranges offset=1024 length=64 C:\Temp\sample.txt

To set the short name for the file *longfilename.txt* on drive C to *longfile.txt*, type:

fsutil file setshortname C:\longfilename.txt longfile.txt

To set the valid data length for a file on an NTFS volume, type:

fsutil file setvaliddata C:\testfile.txt 4096

To set a range of a file on an NTFS volume to zeros and thus empty it, type:

fsutil file setzerodata offset=100 length=150 C:\Temp\sample.txt

Fsutil

Fsutil: behavior

Fsutil: dirty

[Fsutil: fsinfo]

[Fsutil: hardlink]

[Fsutil: objectid]

[Fsutil: quota]

[Fsutil: reparsepoint]

[Fsutil: sparse]

[Fsutil: usn]

[Fsutil: volume]

[Command-line reference A-Z]

Top of page

Manage Your Profile |Legal |Contact Us

Fsutil: fsinfo

Typically used by support professionals. Lists all drives, queries the drive type, queries volume information, queries NTFS-specific volume information, or queries file system statistics.

Syntax

fsutil fsinfo [**drives**]

fsutil fsinfo [**drivetype**] *PathName*

fsutil fsinfo [**ntfsinfo**] *RootPathname*

fsutil fsinfo [**statistics**] *VolumePathname*

fsutil fsinfo [**volumeinfo**] *RootPathname*

⇡Top of page

Parameters

drives : Lists all drives in the computer.

drivetype : Queries a drive and lists its type, for example CD-ROM drive.

ntfsinfo : Lists NTFS specific volume information for the specified volume, such as the number of sectors, total clusters, free clusters, and the start and end of the MFT Zone.

statistics : Lists file system statistics for the specified volume, such as metadata, log file, and MFT reads and writes.

volumeinfo : Lists information for the specified volume, such as the file system, and whether the volume supports case-sensitive file names, unicode in file names, or disk quotas.

***PathName* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

***RootPathname* :** Specifies the drive letter (followed by a colon) of the root drive.

***VolumePathname* :** Specifies the drive letter (followed by a colon), mount point, or volume name.

Top of page

Examples

To query the drive type of drive C, type:

fsutil fsinfo drivetype C:

Possible results:

- Unknown Drive
- No such Root Directory
- Removable Drive, for example floppy
- Fixed Drive
- Remote/Network Drive
- CD-ROM Drive
- Ram Disk

To query the volume information for volume E, type:

fsinfo volumeinfo E:\

Sample output:

```
Volume Name :
Volume Serial Number : 0xd0b634d9
Max Component Length : 255
File System Name : NTFS
Supports Case-sensitive filenames
Preserves Case of filenames
Supports Unicode in filenames
Preserves & Enforces ACL's
Supports file-based Compression
Supports Disk Quotas
Supports Sparse files
Supports Reparse Points
Supports Object Identifiers
Supports Encrypted File System
Supports Named Streams
```

To query drive F for NTFS-specific volume information, type:

fsutil fsinfo ntfsinfo F:\

Sample output:

```
NTFS Volume Serial Number : 0xe660d46a60d442cb
Number Sectors : 0x00000000010ea04f
Total Clusters : 0x000000000021d409
Free Clusters : 0x00000000000f222a
Total Reserved : 0x0000000000003550
Bytes Per Sector : 512
Bytes Per Cluster : 4096
Bytes Per FileRecord Segment : 1024
Clusters Per FileRecord Segment : 0
Mft Valid Data Length : 0x0000000003eab000
Mft Start Lcn : 0x0000000000000004
Mft2 Start Lcn : 0x000000000010ea04
Mft Zone Start : 0x0000000000003ea0
Mft Zone End : 0x0000000000004700
```

To query the file system statistics for drive E, type:

fsinfo statistics E:

Sample output:

```
File System Type : NTFS
Version : 1
UserFileReads : 75021
UserFileReadBytes : 1305244512
UserDiskReads : 71213
UserFileWrites : 8037
UserFileWriteBytes : 207685440
UserDiskWrites : 5278
MetaDataReads : 14377
MetaDataReadBytes : 58888192
MetaDataDiskReads : 14773
```

```
MetaDataWrites : 18705
MetaDataWriteBytes : 138108928
MetaDataDiskWrites : 21464

MftReads : 9394
MftReadBytes : 38477824
MftWrites : 17061
MftWriteBytes : 115359744
Mft2Writes : 0
Mft2WriteBytes : 0
RootIndexReads : 0
RootIndexReadBytes : 0
RootIndexWrites : 0
RootIndexWriteBytes : 0
BitmapReads : 136
BitmapReadBytes : 557056
BitmapWrites : 1180
BitmapWriteBytes : 11161600
MftBitmapReads : 4
MftBitmapReadBytes : 16384
MftBitmapWrites : 118
MftBitmapWriteBytes : 491520
UserIndexReads : 2938
UserIndexReadBytes : 12034048
UserIndexWrites : 2738
UserIndexWriteBytes : 20615168
LogFileReads : 4
LogFileReadBytes : 16384
LogFileWrites : 8407
LogFileWriteBytes : 180936704
```

Fsutil

Fsutil: behavior

Fsutil: dirty

Fsutil: file

Fsutil: hardlink

Fsutil: objectid

Fsutil: quota

Fsutil: reparsepoint

Fsutil: sparse

Fsutil: usn

Fsutil: volume

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us

Fsutil: hardlink

A hard link is a directory entry for a file. Every file can be considered to have at least one hard link. On NTFS volumes, each file can have multiple hard links, and thus a single file can appear in many directories (or even in the same directory with different names). Because all of the links reference the same file, programs can open any of the links and modify the file. A file is deleted from the file system only after all links to it have been deleted. After you create a hard link, programs can use it like any other file name.

Syntax

fsutil hardlink create *NewFilename ExistingFilename*

↑Top of page

Parameters

create : Establishes an NTFS hard link between an existing file and a new file. An NTFS hard link is similar to a POSIX hard link.

***NewFilename* :** Specifies the file to which you want to create a hardlink.

***ExistingFilename* :** Specifies the file from which you want to create a hardlink.

Fsutil

Fsutil: behavior

Fsutil: dirty

Fsutil: file

Fsutil: fsinfo

Fsutil: objectid

Fsutil: quota

Fsutil: reparsepoint

Fsutil: sparse

Fsutil: usn

Fsutil: volume

Command-line reference A-Z

Top of page

Manage Your Profile |Legal |Contact Us