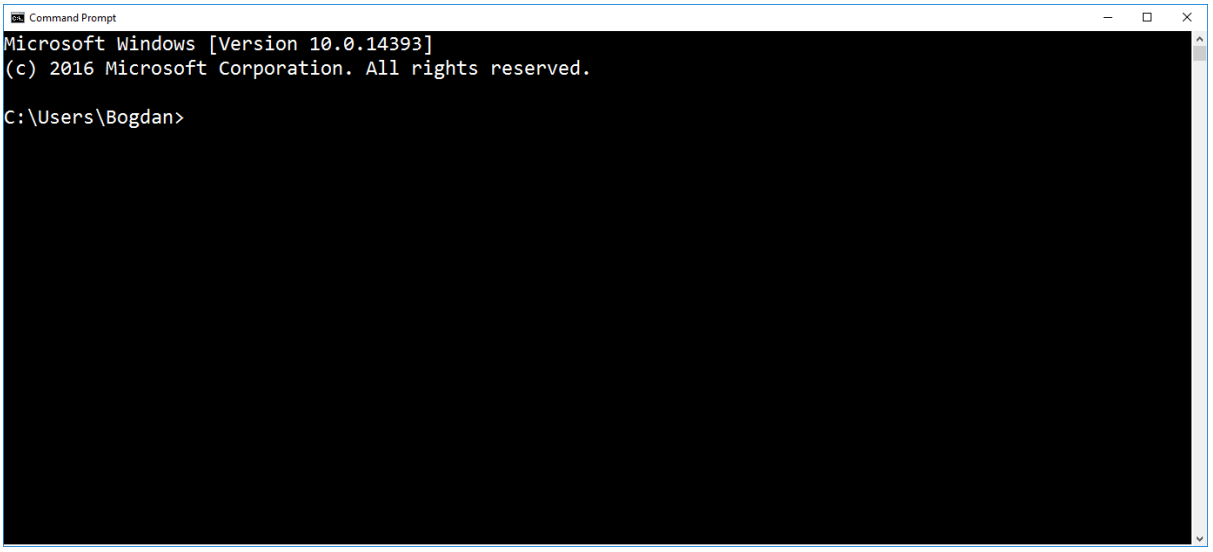
Utilizarea comenzilor text pentru operații legate de rețele de calculatoare și comunicații de date

Cuprins:

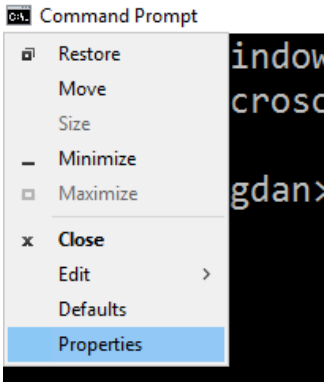
1. Accesarea interfeței text de comandă în Windows:

Windows XP, Vista, 7:
Start->Run->tastăm comanda cmd->Enter
Sau în
Windows 8, 10:
Search (Cortana)->tastăm comanda cmd->Enter

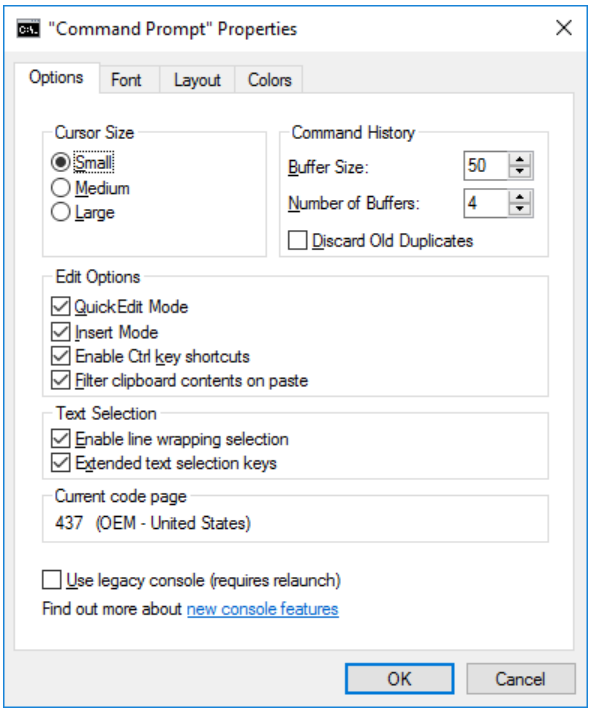


2. Modificarea aspectului și funcționalității interfeței text

Clic pe colțul din stânga sus al ferestrei->Properties

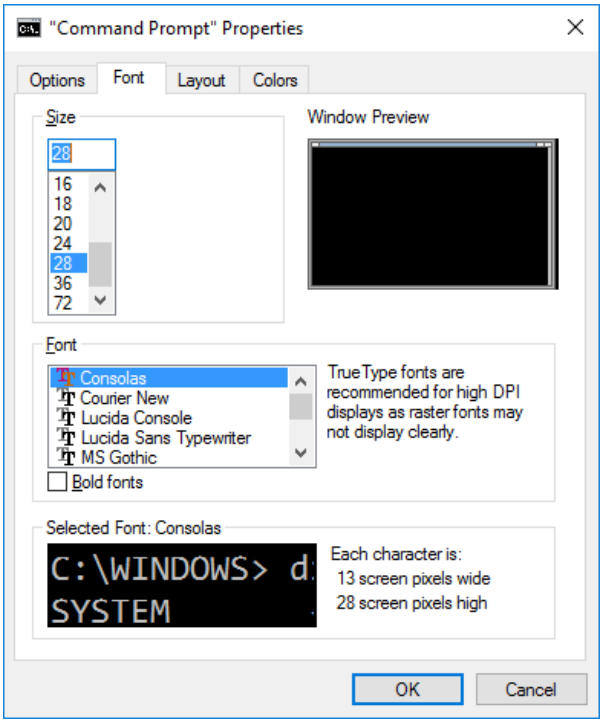


2.1. Tab-ul Options



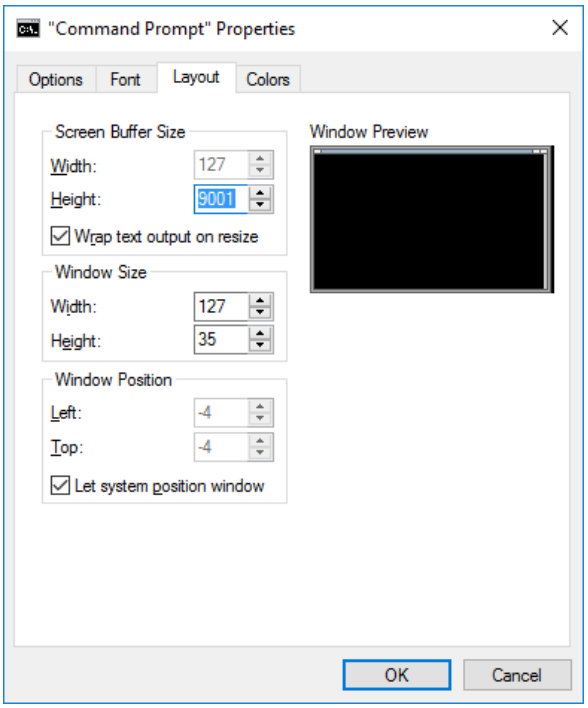
Setări semnificative:
Buffer size: numărul de linii afișate anterior păstrate într-un buffer
Number of buffers: numărul de buffer-e utilizate

2.2. Tab-ul Fonts



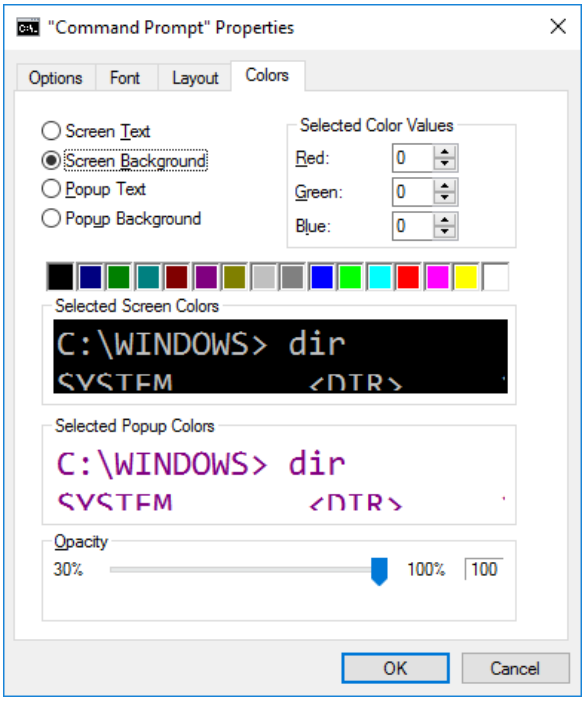
Se recomandă utilizarea unui bine dimensionat și ușor lizibil.

2.3. Tab-ul Layout

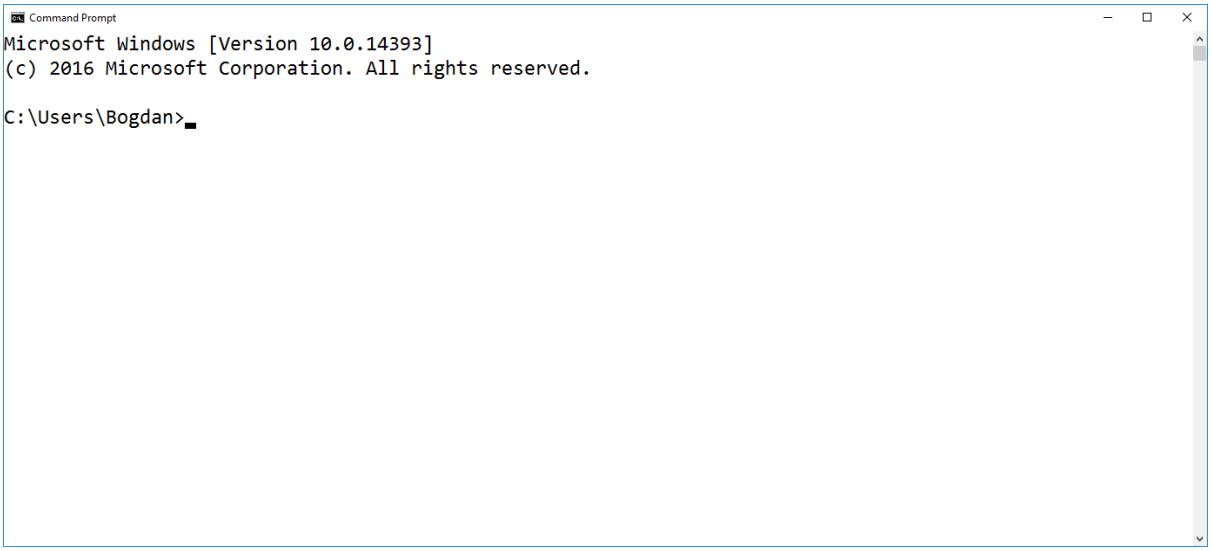


Dimensiunea și poziția ferestrei …

2.4. Tab-ul Colors



Se recomandă alegerea unei scheme de culori ușor lizibile. În cazul în care se dorește realizarea de capturi de ecran pentru tutoriale se recomandă o schemă având culoarea alb drept fundal.



3. Pentru ce poate fi folosită interfața în linie de comandă (CLI)?

Pe scurt:

- Pentru a executa aplicații și pentru a administra foldere, fișiere, echipamente. Un tutorial foarte scurt privind acest subiect se găsește aici: http://www.cs.princeton.edu/courses/archive/spr05/cos126/cmd-prompt.html sau aici: https://www.digitalcitizen.life/command-prompt-how-use-basic-commands
  Un tutorial video mai complet privind acest subiect se găsește aici: https://www.youtube.com/watch?v=MBBWVgE0ewk&list=PL6gx4Cwl9DGDV6SnbINlVUd0o2xT4JbMu
  Un curs de command prompt se găsește aici: https://www.codecademy.com/learn/learn-the-command-line
- Pentru a administra și diagnostica conexiunile de rețea (detalii în secțiunile următoare din acest document)
- Pentru a administra servere și servicii. Mai multe informații se găsesc în diverse cursuri de administrare Windows. O colecție consistentă de astfel de cursuri se găsește spre exemplu, pe Microsoft Virtual Academy aici: https://docs.microsoft.com/en-us/learn/browse/?expanded=windows&products=windows-server

Obs: În afară de Command Prompt, începând cu Windows 7 este disponibilă și o interfață în linie de comandă mult modernizată, denumită Windows PowerShell.
Un curs complet de Windows PowerShell se găsește spre exemplu aici:
https://www.edx.org/course/windows-powershell-basics-microsoft-inf210x-1
Alte cursuri se găsesc pe Microsoft Virtual Academy:
https://docs.microsoft.com/en-us/learn/browse/?expanded=windows&terms=power%20shell
Obs. Microsoft Learn - cursuri gratuite de la Microsoft:
https://docs.microsoft.com/en-us/learn/browse/

4. Comenzi pentru obținerea de informații despre adaptoarele de rețea și comunicațiile desfășurate

4.1. Ipconfig

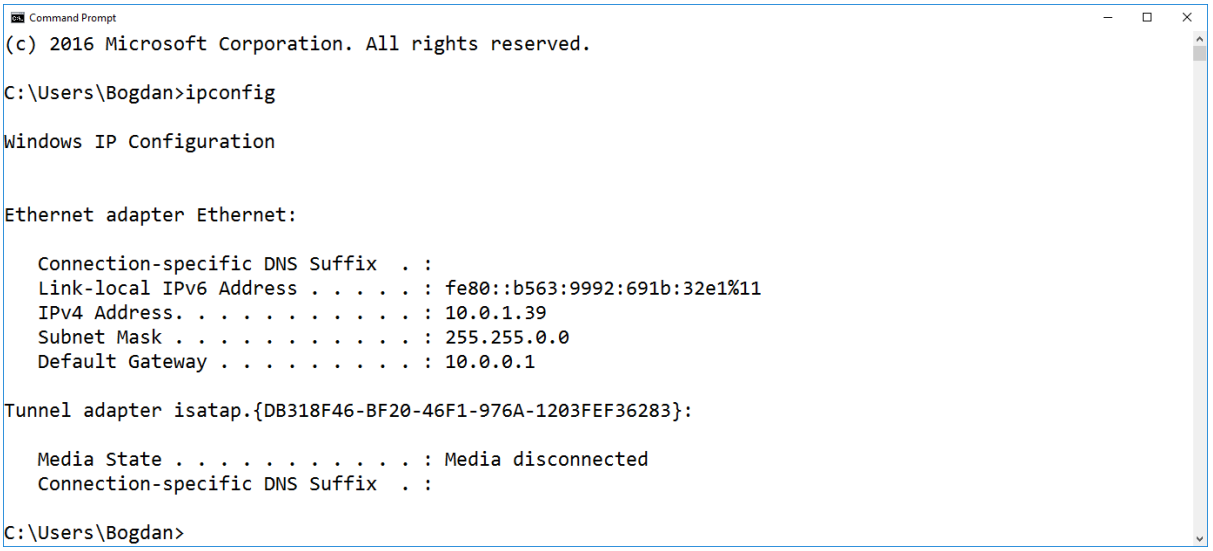
https://www.youtube.com/watch?v=ZKhorleA5aA

IPConfig afișează configurația curentă a stivei IP instalate pe un calculator.

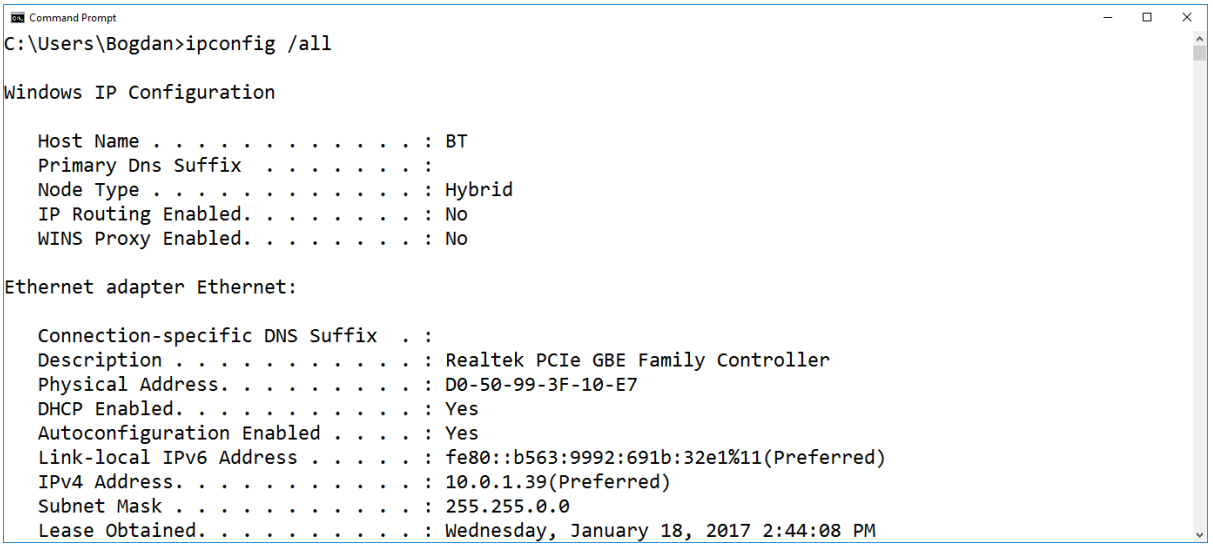
Obs. Comanda cls golește buffer-ul de afișare al interfeței text (șterge tot textul afișat).
Obs. La toate comenzile din această secțiune a tutorialului, precum și la alte comenzi utilizabile în interfața text, informațiile afișate pot fi redirectate către un fișier și copiate ulterior în diverse documente în care e nevoie de ele.
Redirectarea afișării se face utilizând fie simbolul > (dacă se dorește suprascrierea fișierului destinație) fie simbolurile >> (dacă se dorește adăugarea la fișierul destinație).
Ex.
ipconfig >d:\configuratie.txt
va trimite datele furnizate de către ipconfig în fișierul configuratie.txt aflat pe drive-ul D
Obs. sunt disponibile multiple opțiuni de redirectare atât a datelor afișate cât și a datelor de intrare pentru o comandă oarecare:

Operator de redirectare	Descriere
>	Scrie rezultatul execuției comenzii într-un fișier sau îl trimite la un echipament (ex. o imprimantă) în loc să îl afișeze în fereastra Command Prompt.
<	Citește parametrii comenzii dintr-un fișier, în loc să îi citească de la tastatură.
>>	Adaugă rezultatul execuției comenzii la sfârșitul fișierului specificat, fără să șteargă informația deja existentă în fișier.
>&	Scrie ieșirea unui handle în intrarea unui alt handle. Obs. Se numesc handle locația la care se face ieșirea datelor (ex. monitor, imprimantă) și respectiv locația de la care se face intrarea datelor (ex. tastatură).
<&	Citește intrarea unui handle și o scrie în ieșirea unui alt handle.
|	Citește ieșirea unei comenzi și o scrie în intrarea unei alte comenzi. Această metodă de îmbinare a execuției mai multor comenzi (dar și caracterul folosit pentru îmbinare) se mai numește și pipe (conductă :D).

Fără parametri, ipconfig afișează informații de bază despre toate adaptoarele de rețea instalate pe sistemul de calcul respectiv:



Cu parametrul **/all** afișează un raport detaliat de configurație despre toate interfețele, inclusiv cele virtuale.



Parametrii utilizabili sunt:

Parametru	Efect
/all	Produce un raport detaliat de configurare pentru toate interfețele.
/flushdns	Elimină toate intrările din memoria cache de nume DNS.
/registerdns	Reîmprospătează toate lease-urile DHCP și înregistrează nume DNS
/displaydns	Afișează conținutul cache-ului deDNS.

/release < *adapter* >	Eliberează adresa IP pentru o interfață specificată.
/renew < *adapter* >	Reînnoiește adresa IP pentru o interfață specificată.
/showclassid < *adapter* >	Afișează toate ID-urile de clasă DHCP permise pentru adaptorul specificat.
/setclassid < *adapter* > < *classID to set* >	Modifică ID-ul clasei DHCP pentru adaptorul specificat.
/?	Afișează această listă.

A number of other useful parameters for Ipconfig include:

- **/flushdns** , which deletes the DNS name cache;
- **/registerdns** , which refreshes all DHCP leases and re-registers DNS names;
- **/displaydns** which displays the contents of the DNS resolver cache.

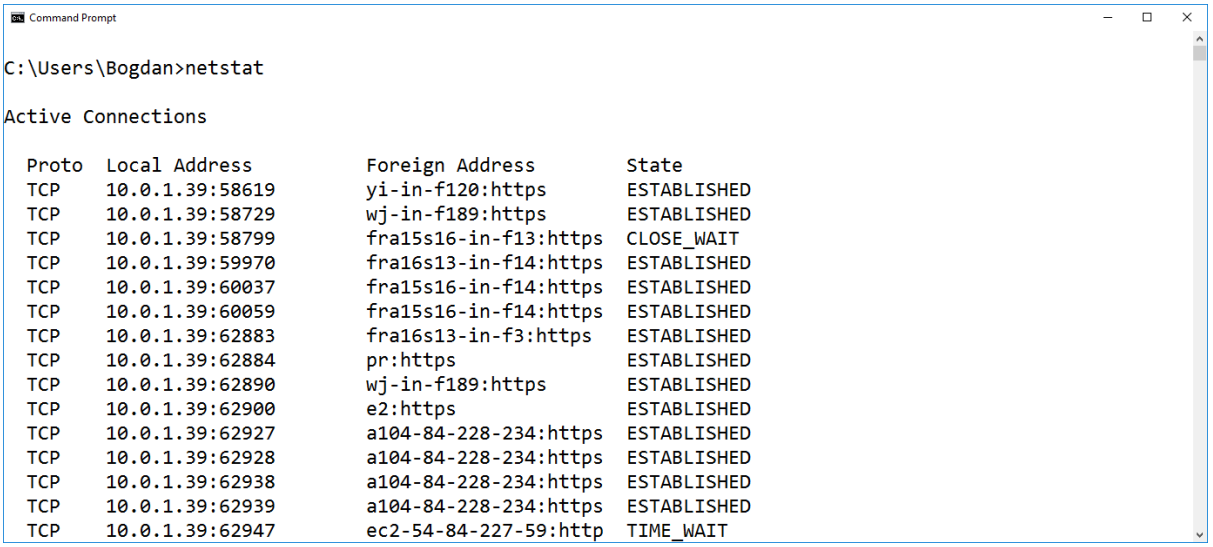
The **/release** < *adapter* > and **/renew** < *adapter* > options release and renew the DHCP-allocated IP address for a specified adapter. If no adapter name is specified, the DHCP leases for all adapters bound to TCP/IP are released or renewed.
For **/setclassid** , if no class ID is specified, then the Class ID is removed.
The /showclassid and /setclassid options allow you to manipulate user class IDs from the command line. The user class IDs are options that a system administrator may set on the DHCP server to configure a client computer to identify itself with the server. Issuing the command **ipconfig /showclassid** < *adapter* > sends a query to the client's server; the server responds by providing the available classes. Once you know which classes are available, you can issue a command like **ipconfig /setdhcpclassid** < *adapter* > < *class ID to set on the server* > to set the class ID that the client will use from that point on. For more information about DHCP and class IDs, see "Dynamic Host Configuration Protocol".

4.2. netstat

Afișează conexiunile TCP active, porturile pe care ascultă computerul, statisticile Ethernet, tabelul de rutare IP, statisticile IPv4 (pentru protocoalele IP, ICMP, TCP și UDP) și statisticile IPv6 (pentru IPv6, ICMPv6, TCP peste IPv6 și UDP prin protocoale IPv6). Folosit fără parametri, **netstat** afișează conexiunile TCP active.

Syntax

netstat [**-a**] [**-e**] [**-n**] [**-o**] [**-p** *Protocol*] [**-r**] [**-s**] [*Interval*]



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.

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

4.3. tracert

The TRACERT diagnostic utility determines the route to a destination by sending Internet Control Message Protocol (ICMP) echo packets to the destination. In these packets, TRACERT uses varying IP Time-To-Live (TTL) values. Because each router along the path is required to decrement the packet's TTL by at least 1 before forwarding the packet, the TTL is effectively a hop counter. When the TTL on a packet reaches zero (0), the router sends an ICMP "Time Exceeded" message back to the source computer.

TRACERT sends the first echo packet with a TTL of 1 and increments the TTL by 1 on each subsequent transmission, until the destination responds or until the maximum TTL is reached. The ICMP "Time Exceeded" messages that intermediate routers send back show the route. Note however that some routers silently drop packets that have expired TTLs, and these packets are invisible to TRACERT.

TRACERT prints out an ordered list of the intermediate routers that return ICMP "Time Exceeded" messages. Using the -d option with the tracert command instructs TRACERT not to perform a DNS lookup on each IP address, so that TRACERT reports the IP address of the near-side interface of the routers.
Sintaxă:
tracert -d -h maximum_hops -j host-list -w timeout target_host

What the parameters do:

-d
Specifies to not resolve addresses to host names

-h maximum_hops
Specifies the maximum number of hops to search for the target

-j host-list
Specifies loose source route along the host-list

-w timeout
Waits the number of milliseconds specified by timeout for each reply

target_host
Specifies the name or IP address of the target host

4.4. ping

Verifică conectivitatea la nivel de IP la un alt computer TCP/IP prin trimiterea de mesaje Echo Request Protocol ICMP (Internet Control Message Protocol). Sunt afișate primirea mesajelor Echo Reply corespunzătoare, împreună cu timpii de călătorie dus-întors. Ping este comanda principală TCP/IP utilizată pentru a depana conexiunea, accesibilitatea și rezoluția numelor. Folosit fără parametri, **ping** afișează textul help-ului.

Syntax

ping [**-t**] [**-a**] [**-n** *Count*] [**-l** *Size*] [**-f**] [**-i** *TTL*] [**-v** *TOS*] [**-r** *Count*] [**-s** *Count*] [{**-j** *HostList* | **-k** *HostList*}] [**-w** *Timeout*] [*TargetName*]

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.

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

4.5. pathping

Instrumentul PathPing este un instrument de urmărire a rutei care combină caracteristicile Ping și Tracert cu informații suplimentare pe care niciunul dintre aceste instrumente nu le oferă. PathPing trimite pachete către fiecare router pe drumul către o destinație finală într-o perioadă de timp și apoi calculează rezultatele pe baza pachetelor returnate de la fiecare

hop. Deoarece PathPing arată gradul de pierdere a pachetelor la orice router sau legătură dat, puteți identifica cu precizie care routere sau legături ar putea cauza probleme de rețea.

Switch	Name	Function
-n	Host names	Does not resolve addresses to host names.
-h < *Max hops* >	Maximum hops	Maximum number of hops to search for target.
-g < *destination address* > < *router IP addresses or NetBIOS names* >	Router -list	Use a loose source route along host-list.
-p < *milliseconds* >	Period	Number of milliseconds to wait between pings.
-q < *Number queries* >	Num_queries	Number of queries per hop.
-R	RSVP test	Checks to see if each router in the path supports the Resource Reservation Protocol (RSVP), which allows the host computer to reserve a certain amount of bandwidth for a data stream. The -R switch is used to test for Quality of Service (QoS) connectivity.
-T	Layer 2 tag	Attaches a layer 2 priority tag (for example, for IEEE 802.1p) to the packets and sends it to each of the network devices in the path. This helps in identifying the network devices that do not have layer 2 priority configured properly. The -T switch is used to test for Quality of Service (QoS) connectivity.
-w < *milliseconds* >	Time-out	Waits this many milliseconds for each reply.

The default number of hops is 30, and the default wait time before a time-out is three seconds (3000 milliseconds). The default period is 250 milliseconds, and the default number of queries to each router along the path is 100.
When PathPing is run, the first results you see list the route as it is tested for problems. This is the same path that is shown via Tracert. PathPing then displays a busy message for the next 125 seconds (this time varies by the hop count, requiring 25 seconds per hop). During this time PathPing gathers information from all the routers previously listed and from the links between them. At the end of this period, it displays the test results.

The two rightmost columns — "This Node/Link Lost/Sent=%" and "Address" — contain the most useful information. The link between 172.16.87.218 (hop 1), and 192.168.52.1 (hop 2) is dropping 13 percent of the packets. All other links are working normally. The routers at hops 2 and 4 also drop packets addressed to them (as shown in the "This Node/Link" column), but this loss does not affect their forwarding path.
The loss rates displayed for the links (marked as a "|" in the rightmost column) indicate losses of packets being forwarded along the path. This loss indicates link congestion. The loss rates displayed for routers (indicated by their IP addresses in the rightmost column) indicate that those routers' CPUs or packet buffers might be overloaded. These congested routers might also be a factor in end-to-end problems, especially if packets are forwarded by software routers.

Loss Calculation

The raw data that PathPing obtains describes how many ICMP Echo Requests are lost between the source and an intermediate router. Figure 3.1 shows how PathPing estimates the per-hop loss statistics. While at first this calculation might seem trivial, it is complicated by differences between the forwarding code path and the code path taken in responding to ping packets (ICMP Echo Requests/Replies).

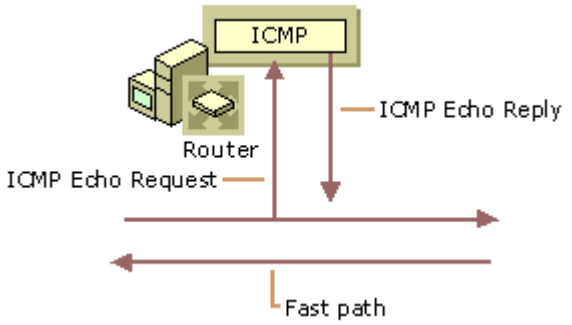


Figure 4.1 Packet Delivery Paths

The horizontal lines indicate the "fast path" of a router, which is taken by packets that are not sent to or from the local computer. That is, the fast path is the code path taken by transit packets that require no special processing other than forwarding, and is highly optimized for such packets.
In the diagram, the vertical lines indicate the extra processing taken when an ICMP Echo Request is sent to the local computer. This kicks it out of the fast path and delivers it to an ICMP module (often using separate queues and processors). Assuming no packets are dropped due to queue overflows, the ICMP module then generates an ICMP Echo Reply, which is forwarded back to the original sender.
Since packet loss can occur in the path indicated by the vertical lines (but such loss does not necessarily imply loss on the horizontal forwarding path itself), the raw numbers obtained from pings do not by themselves determine end-to-end packet loss. For example, pinging an intermediate router might create a 10 percent loss even though no end-to-end packet loss is occurring. PathPing's algorithm uses the change in values from hop-to-hop to estimate actual per hop loss rather than losses in the higher-level router components. This actual per hop loss is the result provided in the "This Node/Link" column of the final PathPing report.

4.6. route

Afișează și modifică intrările din tabelul local de rutare IP. Folosit fără parametri, **route** afișează textul help-ului.

Syntax

route [**-f**] [**-p**] [*Command* [*Destination*] [**mask** *Netmask*] [*Gateway*] [**metric** *Metric*]] [**if** *Interface*]]

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.

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

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

4.7. arp

Arp allows you to view and modify the ARP cache. If two hosts on the same subnet cannot ping each other successfully, try running the **arp -a** command on each computer to see whether the computers have the correct media access control (MAC) addresses listed for each other. You can use Ipconfig to determine a host's correct MAC address.
You can also use Arp to view the contents of the ARP cache by typing **arp -a** at a command prompt. This displays a list of the ARP cache entries, including their MAC addresses.

- If another host with a duplicate IP address exists on the network, the ARP cache might have the MAC address for the other computer placed in it, and this can lead to intermittent problems with address resolution. When a computer on the local network

sends an ARP Request to resolve the address, it forwards its data to the MAC address corresponding to the first ARP Reply it receives. **Arp** can help by listing, adding, and removing the relevant entries.

- You can use **arp -d** < *IP address* > to delete incorrect entries. Use **arp -s <** *MAC address* **>** (where the MAC address is formatted as hexadecimal bytes separated by dashes) to add new static entries; these static entries do not expire from the ARP cache. However, static entries do not persist after a reboot. For persistent static ARP cache entries, you must create a batch file run from the Startup group.
- Use **arp -N** < *IP address* > to list all the ARP entries for the network interface specified by < *IP address*>. Table 3.2 lists all Arp switches.

Parameter	Description
/a [<InetAddr>] [/n <IfaceAddr>]	Displays current ARP cache tables for all interfaces. The /n parameter is case-sensitive. To display the ARP cache entry for a specific IP address, use **arp /a** with the *InetAddr* parameter, where *InetAddr* is an IP address. If *InetAddr* is not specified, the first applicable interface is used. To display the ARP cache table for a specific interface, use the **/n** *IfaceAddr* parameter in conjunction with the **/a** parameter where *IfaceAddr* is the IP address assigned to the interface.
/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.
-N < *Interface IP address* >	Lists all ARP entries for the interface specified
/?	Displays Help at the command prompt.

- 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.

4.8. nslookup

Afișează informații pe care le puteți utiliza pentru a diagnostica infrastructura DNS (Domain Name System). Înainte de a utiliza acest instrument, ar trebui să vă familiarizați cu modul în care funcționează DNS. Instrumentul de linie de comandă Nslookup este disponibil numai dacă ați instalat protocolul TCP/IP.

Parameters

Parameter	Description
Nslookup -exit	Exits **nslookup**.
Nslookup -finger	Connects with the finger server on the current computer.
Nslookup -help	Displays a short summary of **nslookup** subcommands.
Nslookup -ls	Lists information for a DNS domain.
Nslookup -lserver	Changes the default server to the specified DNS domain.
Nslookup -root	Changes the default server to the server for the root of the DNS domain name space.
Nslookup -server	Changes the default server to the specified DNS domain.
Nslookup -set	Changes configuration settings that affect how lookups function.
Nslookup -set all	Prints the current values of the configuration settings.
Nslookup -set class	Changes the query class. The class specifies the protocol group of the information.
Nslookup -set d2	Turns exhaustive Debugging Mode on or off. All fields of every packet are printed.
Nslookup -set debug	Turns Debugging Mode on or off.
Nslookup -set defname	Appends the default DNS domain name to a single component

	lookup request. A single component is a component that contains no periods.
Nslookup -set domain	Changes the default DNS domain name to the name specified.
Nslookup -set ignore	Ignores packet truncation errors.
Nslookup -set port	Changes the default TCP/UDP DNS name server port to the value specified.
Nslookup -set querytype	Changes the resource record type for the query.
Nslookup -set recurse	Tells the DNS name server to query other servers if it does not have the information.
Nslookup -set retry	Sets the number of retries.
Nslookup -set root	Changes the name of the root server used for queries.
Nslookup -set search	Appends the DNS domain names in the DNS domain search list to the request until an answer is received. This applies when the set and the lookup request contain at least one period, but do not end with a trailing period.
Nslookup -set srchlist	Changes the default DNS domain name and search list.
Nslookup -set timeout	Changes the initial number of seconds to wait for a reply to a request.
Nslookup -set type	Changes the resource record type for the query.
Nslookup -set vc	Specifies to use or not use a virtual circuit when sending requests to the server.
Nslookup -view	Sorts and lists the output of the previous **ls** subcommand or commands.

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**.

4.9. 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. For examples of how this command can be used, see Examples.

Syntax

Nbtstat [/a <RemoteName>] [/A <IPAddress>] [/c] [/n] [/r] [/R] [/RR] [/s] [/S] [<Interval>]

Parameters

Parameter	Description
/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 computer running Windows XP or Windows Server 2003 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.

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.

4.10. 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.

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

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.

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.

4.11. 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**]

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.

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.

5. Programe utilitare

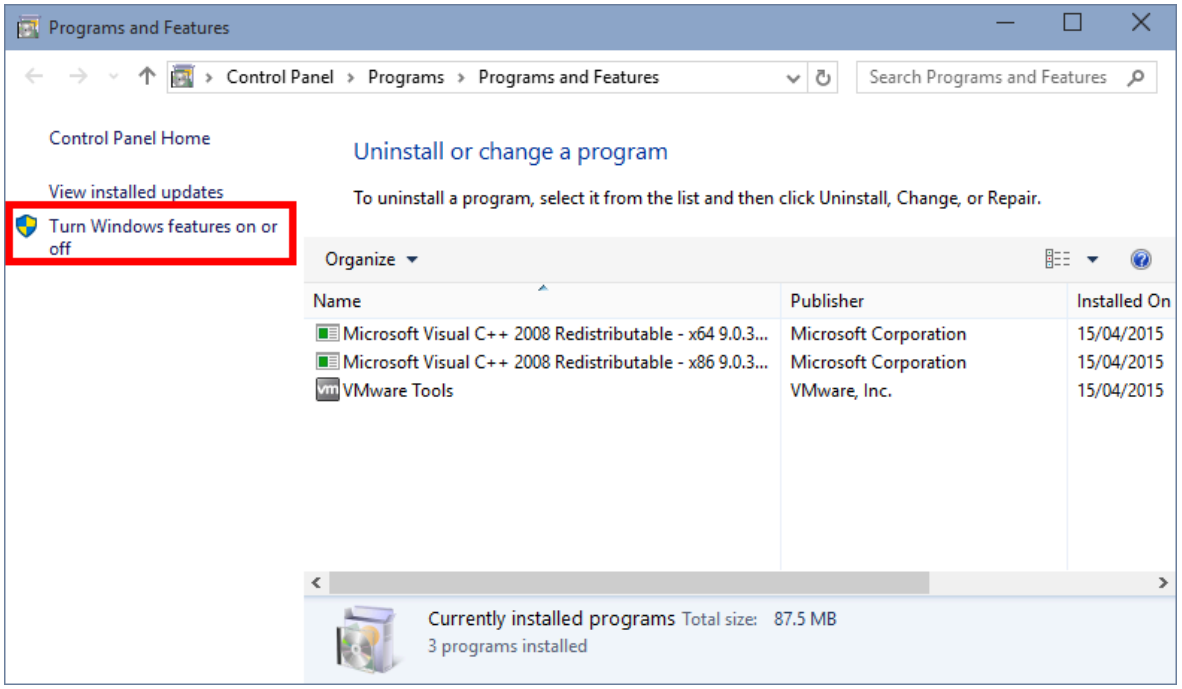
5.1. telnet

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

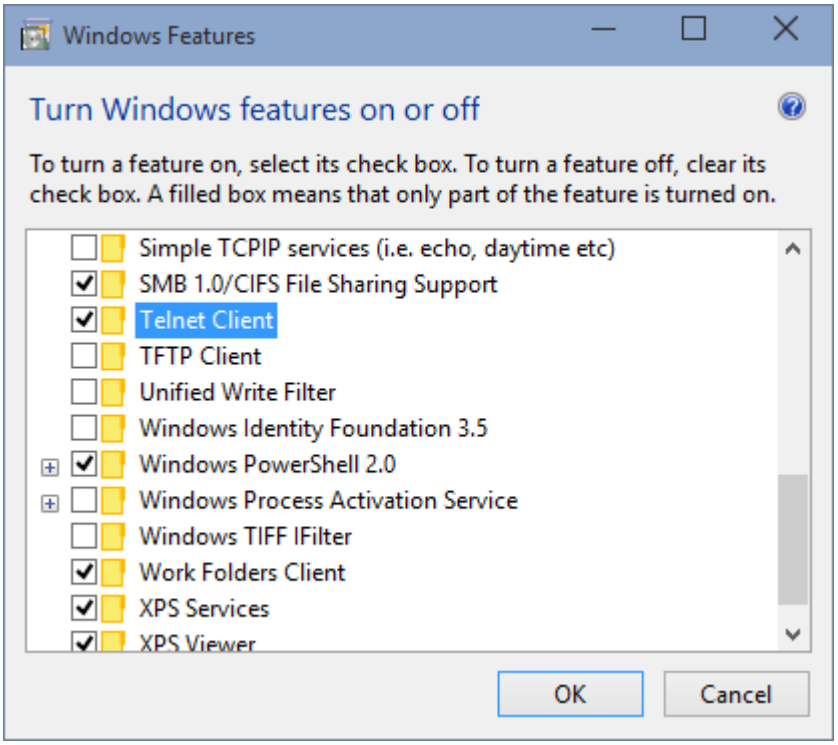
Obs. Clientul telnet este in mod implicit neinstalat / dezactivat in Windows 7 si respectiv in Windows 10.

Pentru a instala clientul telnet în Windows 7 sau 10, utilizând interfața grafica (sunt necesare drepturi de aministrator):

- Control Panel->Programs and Features->Turn Windows features on or off:



- Se selecteaza Telnet Client si se da OK:



Pentru a instala clientul telnet în Windows 7, utilizând interfața în linie de comandă (sunt necesare drepturi de administrator):

- Comanda pkgmgr /iu:"TelnetClient" urmată de repornirea Command prompt-ului

Pentru a instala clientul telnet în Windows 10, utilizând interfața în linie de comandă (sunt necesare drepturi de administrator):

- Comanda dism /online /Enable-Feature /FeatureName:TelnetClient urmată de repornirea Command prompt-ului

Obs. Serverul telnet a fost eliminat din Windows 10 si respectiv a fost marcat ca fiind depășit (deși poate fi instalat) pe serverele MS începând cu MS Windows Server 2012. El este în schimb disponibil pe toate versiunile de Linux, și poate fi activat pe toate versiunile de MS Windows client, până la Windows 8.1 și pe toate versiunile de MS Windows Server (mai multe informații aici: https://technet.microsoft.com/en-us/library/cc754837(v=ws.10).aspx)

Syntax

telnet [*RemoteServer*]

Parameters

\\ *RemoteServer* : Specifies the name of the server to which you want to connect.
/? : Displays help at the command prompt.

Remarks

- Used without parameters, **telnet** starts Telnet Client.
- When you are at the Telnet prompt, you must use Telnet commands.

To stop Telnet Client

Syntax

quit

Parameters

none

Remarks

- You can abbreviate this command to **q**.

To connect Telnet Client to a remote computer

Syntax

open [*RemoteServer*] [*Port*]

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

Syntax

close [*RemoteServer*]

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

Syntax

set [*RemoteServer*] [**ntlm**] [**localecho**] [**term** {**ansi** | **vt100** | **vt52** | **vtnt**}] [**escape** *Character*] [**logfile** *FileName*] [**logging**] [**bsasdel**] [**crlf**] [**delasbs**] [**mode** {**console** | **stream**}] [**?**]

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

Syntax

send [*RemoteServer*] [**ao**] [**ayt**] [**esc**] [**ip**] [**synch**] [**?**]

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

Syntax

display

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

Syntax

tlntadmn [*****RemoteServer*] [**-s**] [**-k**{*SessionID* | **all**}] [**-m** {*SessionID* | **all**} **"***Message***"**]

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

Syntax

tlntadmn [*****RemoteServer*] **config** [**auditlocation=**{**eventlog** | **file** | **both**}] [**audit=**[{**+** | **-**}**admin**][{**+** | **-**}**user**][{**+** | **-**}**fail**]]

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.

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**dom=***DomainName*]

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.

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**ctrlakeymap=**{**yes** | **no**}]

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**maxconn=***PositiveInteger*]

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**maxfail=***PositiveInteger*]

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**mode=**{**console** | **stream**}]

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**port=***IntegerValue*]

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**sec=**[{**+** | **-**}**ntlm**][{**+** | **-**}**passwd**]]

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.

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

Syntax

tlntadmn [*RemoteServer*] **config** [**timeout=***hh***:***mm***:***ss*]

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.

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.

Remarks

- To switch from Telnet Client to command mode, at the Telnet prompt, press CTRL+]. To switch back to Telnet Client, press ENTER.

5.2. ftp

5.3. putty