# 参考文档1

Netsh 是命令行脚本实用工具，可让你显示或修改当前正在运行的计算机的网络配置。 可以通过在 netsh 提示符下键入命令来运行 Netsh 命令，并且可以在批处理文件或脚本中使用 Netsh 命令。 可以使用 netsh 命令来配置远程计算机和本地计算机。

Netsh 还提供脚本功能，可让你在批处理模式下对指定的计算机运行一组命令。 你可以使用 Netsh 将配置脚本保存在文本文件中，以便存档或者帮助你配置其他计算机。

## 一、如何正确使用netsh

1. 当我们忘记netsh的相关命令时，可以直接使用netsh自带的帮助功能，就能看到一些可以使用的命令。

shell

netsh /?

C:\Users\ming Ming>netsh /?

Usage: netsh [-a AliasFile] [-c Context] [-r RemoteMachine] [-u [DomainName\]UserName] [-p Password | \*]

[Command | -f ScriptFile]

The following commands are available:

Commands in this context:

? - Displays a list of commands.

add - Adds a configuration entry to a list of entries.

advfirewall - Changes to the `netsh advfirewall' context.

bridge - Changes to the `netsh bridge' context.

delete - Deletes a configuration entry from a list of entries.

dhcpclient - Changes to the `netsh dhcpclient' context.

dnsclient - Changes to the `netsh dnsclient' context.

dump - Displays a configuration script.

exec - Runs a script file.

firewall - Changes to the `netsh firewall' context.

help - Displays a list of commands.

http - Changes to the `netsh http' context.

interface - Changes to the `netsh interface' context.

ipsec - Changes to the `netsh ipsec' context.

lan - Changes to the `netsh lan' context.

mbn - Changes to the `netsh mbn' context.

namespace - Changes to the `netsh namespace' context.

netio - Changes to the `netsh netio' context.

nlm - Changes to the `netsh nlm' context.

p2p - Changes to the `netsh p2p' context.

ras - Changes to the `netsh ras' context.

rpc - Changes to the `netsh rpc' context.

set - Updates configuration settings.

show - Displays information.

trace - Changes to the `netsh trace' context.

wcn - Changes to the `netsh wcn' context.

wfp - Changes to the `netsh wfp' context.

winhttp - Changes to the `netsh winhttp' context.

winsock - Changes to the `netsh winsock' context.

wlan - Changes to the `netsh wlan' context.

The following sub-contexts are available:

advfirewall bridge dhcpclient dnsclient firewall http interface ipsec lan mbn namespace netio nlm p2p ras rpc trace wcn wfp winhttp winsock wlan

To view help for a command, type the command, followed by a space, and then

type ?.

1. 那知道有哪些可选功能后，我们就可以进一步输入二级命令去执行我们将要进行的操作，比如下边我就以查看当前tcp链接状态位列展示：

shell

复制代码

$>netsh

$netsh>int

$netsh interface>ip

$netsh interface ipv4>show tcpstats

TCP Statistics

------------------------------------------------------

Timeout Algorithm: Van Jacobson's Algorithm

Minimum Timeout: 5

Maximum Timeout: 4294967295

Maximum Connections: Dynamic

Active Opens: 3971

Passive Opens: 556

Attempts Failed: 1017

Established Resets: 407

Currently Established: 34

In Segments: 202193

Out Segments: 149657

Retransmitted Segments: 0

In Errors: 0

Out Resets: 845

Fastopen Active Opens: 0

Fastopen Passive Opens: 0

Fastopen Attempts Failed: 0

Retransmits Of First SYN: 644

Retransmits Of First SYN (Fastopen): 0

## 二、netsh命令整理

### 1. 导出导入脚本

1. 导出配置脚本：

netsh -c interface ip dump > c:/interface.txt

1. 导入配置脚本：

netsh -f c:/interface.txt

### 2. 固定IP地址批处理

//SETIPDNS.bat（ “本地连接”换成你网卡的名字, 具体的 ip 地址，网关，请自行修改 ）

@echo off

echo 正在修改本机IP...

netsh interface ip set address name="本地连接" source=static 192.168.1.188 255.255.255.0 192.168.1.1 1

echo 正在添加本机主DNS...

netsh interface ip set dns "本地连接" static 18.0.0.10 primary

echo 正在添加本机副DNS...

netsh interface ip add dns "本地连接" 18.0.0.10

echo 设置完成...

echo 检查当前本机配置...

ipconfig /all

echo 导出配置脚本...

netsh -c interface ip dump

pause

### 3. 自动获取 IP

@echo off

echo 正在修改本机IP...

netsh interface ip set address name="本地连接" source=dhcp

echo 正在添加本机主DNS...

netsh interface ip set dns "本地连接" dhcp

echo 设置完成...

echo 检查当前本机配置...

ipconfig /all

echo 导出配置脚本...

netsh -c interface ip dump 注意：这里的IP默认是值ipv4而不是一个具体的ip地址

pause

### 4. netsh用法

1. 查看网络配置

netsh interface ip show {选项} 注意：这里的IP默认是值ipv4而不是一个具体的ip地址

{选项}可以是：

address-显示IP地址配置。

config-显示IP地址和更多信息。

dns-显示DNS服务器地址。

icmp-显示ICMP统计。

interface-显示IP接口统计。

ipaddress-显示当前IP地址。

ipnet-显示IP的网络到媒体的映射。

ipstats-显示IP统计。

joins-显示加入的多播组。

offload-显示卸载信息。

tcpconn-显示TCP连接。

tcpstats-显示TCP统计。

udpconn-显示UDP连接。

udpstats-显示UDP统计。

wins-显示WINS服务器地址。

1. 查看网络配置

netsh interface ip show {选项}

{选项}可以是：

address-显示IP地址配置。

config-显示IP地址和更多信息。

dns-显示DNS服务器地址。

icmp-显示ICMP统计。

interface-显示IP接口统计。

ipaddress-显示当前IP地址。

ipnet-显示IP的网络到媒体的映射。

ipstats-显示IP统计。

joins-显示加入的多播组。

offload-显示卸载信息。

tcpconn-显示TCP连接。

tcpstats-显示TCP统计。

udpconn-显示UDP连接。

udpstats-显示UDP统计。

wins-显示WINS服务器地址。

1. 配置接口IP/网关IP

netsh interface ip set address "本地连接" static 10.1.2.90 255.255.255.0 10.1.2.254 1

1. 配置自动换取IP地址,DNS地址及wins地址

netsh interface ip set address "本地连接" dhcp

netsh interface ip set dns "本地连接" dhcp

netsh interface ip set wins "本地连接" dhcp

1. 配置静态IP地址，DNS地址及wins地址

netsh interface ip set address "本地连接" static 10.1.2.90

netsh interface ip set dns "本地连接" static 202.99.160.68

netsh interface ip set wins "本地连接" static 10.1.2.200

1. 看网络配置文件

C:>netsh -c interface dump

1. 导出网络配置文件

netsh -c interface dump > d:/1.txt

1. 导入网络配置文件

netsh -f d:/1.txt

netsh exec d:/2.txt

1. netsh 配置防火墙 查看、开启或禁用系统防火墙

# 查看防火墙的状态

netsh firewall show state

# 禁用系统防火墙

netsh firewall set opmode disable

# 启用防火墙

netsh firewall set opmode enable

# 参考文档2

**一、net**[sh命令](https://so.csdn.net/so/search?q=sh%E5%91%BD%E4%BB%A4&spm=1001.2101.3001.7020)**的简介**

netsh（Network Shell）是一个windows系统本身提供的功能强大的[网络配置](https://so.csdn.net/so/search?q=%E7%BD%91%E7%BB%9C%E9%85%8D%E7%BD%AE&spm=1001.2101.3001.7020)命令行工具，可用来修改windows的ip、网关、dns等信息。

netsh是一个上下文命令，它有很多子命令，这些子命令各自又有各自的子命令，这样反复嵌套下来一共可以有4层命令或参数，例如【netsh interface [ipv4](https://so.csdn.net/so/search?q=ipv4&spm=1001.2101.3001.7020) dump】。

如果进入了上下文环境，执行命令【exit】、【exit /b】可直接退出所有层级的上下文环境。

**二、打印netsh命令帮助信息**

**1.1 打印最外层netsh命令的帮助信息**

执行命令【netsh /?】，显示帮助信息

C:\WINDOWS\system32>netsh /?

用法: netsh [-a AliasFile] [-c Context] [-r RemoteMachine] [-u [DomainName\]UserName] [-p Password | \*]

[Command | -f ScriptFile]

下列指令有效:

此上下文中的命令:

? - 显示命令列表。

add - 在项目列表上添加一个配置项目。

advfirewall - 更改到 `netsh advfirewall' 上下文。

branchcache - 更改到 `netsh branchcache' 上下文。

bridge - 更改到 `netsh bridge' 上下文。

delete - 在项目列表上删除一个配置项目。

dhcpclient - 更改到 `netsh dhcpclient' 上下文。

dnsclient - 更改到 `netsh dnsclient' 上下文。

dump - 显示一个配置脚本。

exec - 运行一个脚本文件。

firewall - 更改到 `netsh firewall' 上下文。

help - 显示命令列表。

http - 更改到 `netsh http' 上下文。

interface - 更改到 `netsh interface' 上下文。

ipsec - 更改到 `netsh ipsec' 上下文。

lan - 更改到 `netsh lan' 上下文。

mbn - 更改到 `netsh mbn' 上下文。

namespace - 更改到 `netsh namespace' 上下文。

netio - 更改到 `netsh netio' 上下文。

p2p - 更改到 `netsh p2p' 上下文。

ras - 更改到 `netsh ras' 上下文。

rpc - 更改到 `netsh rpc' 上下文。

set - 更新配置设置。

show - 显示信息。

trace - 更改到 `netsh trace' 上下文。

wcn - 更改到 `netsh wcn' 上下文。

wfp - 更改到 `netsh wfp' 上下文。

winhttp - 更改到 `netsh winhttp' 上下文。

winsock - 更改到 `netsh winsock' 上下文。

wlan - 更改到 `netsh wlan' 上下文。

下列的子上下文可用:

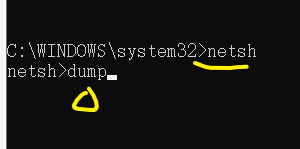
advfirewall branchcache bridge dhcpclient dnsclient firewall http interface ipsec lan mbn namespace netio p2p ras rpc trace wcn wfp winhttp winsock wlan

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

**2.2 netsh上下文机制与子命令上下文的进入方法**

由于netsh命令是一个上下文命令，因此当直接执行netsh时，会进入netsh环境；而进一步使用诸如dump、delete这些命令时，效果和不先进入上下文环境而直接执行【netsh dump】、【netsh delete】的完全一样。



然而！！！

对于命令后带有「更改到 ' xxx ' 上下文」注释信息的命令本身（不带子命令或参数），就必须在netsh上下文环境中执行。

比如说interface命令后就有这种注释信息，因此如果要进入netsh interface上下文，必须先执行【netsh】进入netsh上下文。

再执行【interface】命令（或简写的int）进一步进入netsh interface上下文。

而再进一步执行【ip】或【ipv4】，就可进入netsh interface ipv4上下文；或者执行【ipv6】，进入netsh interface ipv6上下文。

**2.3 打印netsh命令的各个层的子命令帮助信息**

（1）不在上下文中打印的帮助信息

打印帮助信息不管是不是处于上下文环境都可以；下面以netsh——>interface——>ipv4——>dump这个4层深度的命令链为例，讲述该链经过了netsh命令树的哪些“枝干”。

如果不是处于上下文环境，直接执行【netsh interface】，就会显示netsh interface上下文的子命令或参数。

C:\WINDOWS\system32>netsh interface

下列指令有效:

此上下文中的命令:

6to4 - 更改到 `netsh interface 6to4' 上下文。

? - 显示命令列表。

dump - 显示一个配置脚本。

help - 显示命令列表。

httpstunnel - 更改到 `netsh interface httpstunnel' 上下文。

ipv4 - 更改到 `netsh interface ipv4' 上下文。

ipv6 - 更改到 `netsh interface ipv6' 上下文。

isatap - 更改到 `netsh interface isatap' 上下文。

portproxy - 更改到 `netsh interface portproxy' 上下文。

set - 设置配置信息。

show - 显示信息。

tcp - 更改到 `netsh interface tcp' 上下文。

teredo - 更改到 `netsh interface teredo' 上下文。

udp - 更改到 `netsh interface udp' 上下文。

下列的子上下文可用:

6to4 httpstunnel ipv4 ipv6 isatap portproxy tcp teredo udp

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

同理，直接执行【netsh interface ipv4】会显示netsh interface ipv4上下文的子命令或参数。

C:\WINDOWS\system32>netsh interface ipv4

下列指令有效:

此上下文中的命令:

? - 显示命令列表。

add - 在一个表格中添加一个配置项。

delete - 从一个表格中删除一个配置项。

dump - 显示一个配置脚本。

help - 显示命令列表。

install - 安装 IP 协议。

reset - 重置 IP 配置。

set - 设置配置信息。

show - 显示信息。

uninstall - 卸载 IP 协议。

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

（2）在上下文中打印帮助信息，会包括继承部分

 另外一点，我们也可以发现，当我们进入netsh的子命令的上下文后，再执行【?】，就会把自父命令（netsh）继承下来的命令也给打印出来。

netsh interface>?

下列指令有效:

命令从 netsh 上下文继承:

.. - 移到上一层上下文级。

abort - 丢弃在脱机模式下所做的更改。

add - 在项目列表上添加一个配置项目。

advfirewall - 更改到 `netsh advfirewall' 上下文。

alias - 添加一个别名

branchcache - 更改到 `netsh branchcache' 上下文。

bridge - 更改到 `netsh bridge' 上下文。

bye - 退出程序。

commit - 提交在脱机模式中所做的更改。

delete - 在项目列表上删除一个配置项目。

dhcpclient - 更改到 `netsh dhcpclient' 上下文。

dnsclient - 更改到 `netsh dnsclient' 上下文。

exit - 退出程序。

firewall - 更改到 `netsh firewall' 上下文。

http - 更改到 `netsh http' 上下文。

interface - 更改到 `netsh interface' 上下文。

ipsec - 更改到 `netsh ipsec' 上下文。

lan - 更改到 `netsh lan' 上下文。

mbn - 更改到 `netsh mbn' 上下文。

namespace - 更改到 `netsh namespace' 上下文。

netio - 更改到 `netsh netio' 上下文。

offline - 将当前模式设置成脱机。

online - 将当前模式设置成联机。

p2p - 更改到 `netsh p2p' 上下文。

popd - 从堆栈上打开一个上下文。

pushd - 将当前上下文放入堆栈。

quit - 退出程序。

ras - 更改到 `netsh ras' 上下文。

rpc - 更改到 `netsh rpc' 上下文。

set - 更新配置设置。

show - 显示信息。

trace - 更改到 `netsh trace' 上下文。

unalias - 删除一个别名。

wcn - 更改到 `netsh wcn' 上下文。

wfp - 更改到 `netsh wfp' 上下文。

winhttp - 更改到 `netsh winhttp' 上下文。

winsock - 更改到 `netsh winsock' 上下文。

wlan - 更改到 `netsh wlan' 上下文。

此上下文中的命令:

6to4 - 更改到 `netsh interface 6to4' 上下文。

? - 显示命令列表。

dump - 显示一个配置脚本。

help - 显示命令列表。

httpstunnel - 更改到 `netsh interface httpstunnel' 上下文。

ipv4 - 更改到 `netsh interface ipv4' 上下文。

ipv6 - 更改到 `netsh interface ipv6' 上下文。

isatap - 更改到 `netsh interface isatap' 上下文。

portproxy - 更改到 `netsh interface portproxy' 上下文。

set - 设置配置信息。

show - 显示信息。

tcp - 更改到 `netsh interface tcp' 上下文。

teredo - 更改到 `netsh interface teredo' 上下文。

udp - 更改到 `netsh interface udp' 上下文。

下列的子上下文可用:

6to4 httpstunnel ipv4 ipv6 isatap portproxy tcp teredo udp

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

而我们在执行【ip】进入netsh interface ipv4上下文后，用命令【?】得到的帮助信息中，不仅对最顶层的netsh命令进行了继承，也对二层的netsh interface命令进行了继承。

netsh interface ipv4>?

下列指令有效:

命令从 netsh 上下文继承:

.. - 移到上一层上下文级。

abort - 丢弃在脱机模式下所做的更改。

add - 在项目列表上添加一个配置项目。

advfirewall - 更改到 `netsh advfirewall' 上下文。

alias - 添加一个别名

branchcache - 更改到 `netsh branchcache' 上下文。

bridge - 更改到 `netsh bridge' 上下文。

bye - 退出程序。

commit - 提交在脱机模式中所做的更改。

delete - 在项目列表上删除一个配置项目。

dhcpclient - 更改到 `netsh dhcpclient' 上下文。

dnsclient - 更改到 `netsh dnsclient' 上下文。

exit - 退出程序。

firewall - 更改到 `netsh firewall' 上下文。

http - 更改到 `netsh http' 上下文。

interface - 更改到 `netsh interface' 上下文。

ipsec - 更改到 `netsh ipsec' 上下文。

lan - 更改到 `netsh lan' 上下文。

mbn - 更改到 `netsh mbn' 上下文。

namespace - 更改到 `netsh namespace' 上下文。

netio - 更改到 `netsh netio' 上下文。

offline - 将当前模式设置成脱机。

online - 将当前模式设置成联机。

p2p - 更改到 `netsh p2p' 上下文。

popd - 从堆栈上打开一个上下文。

pushd - 将当前上下文放入堆栈。

quit - 退出程序。

ras - 更改到 `netsh ras' 上下文。

rpc - 更改到 `netsh rpc' 上下文。

set - 更新配置设置。

show - 显示信息。

trace - 更改到 `netsh trace' 上下文。

unalias - 删除一个别名。

wcn - 更改到 `netsh wcn' 上下文。

wfp - 更改到 `netsh wfp' 上下文。

winhttp - 更改到 `netsh winhttp' 上下文。

winsock - 更改到 `netsh winsock' 上下文。

wlan - 更改到 `netsh wlan' 上下文。

命令从 netsh interface 上下文继承:

6to4 - 更改到 `netsh interface 6to4' 上下文。

httpstunnel - 更改到 `netsh interface httpstunnel' 上下文。

ipv4 - 更改到 `netsh interface ipv4' 上下文。

ipv6 - 更改到 `netsh interface ipv6' 上下文。

isatap - 更改到 `netsh interface isatap' 上下文。

portproxy - 更改到 `netsh interface portproxy' 上下文。

set - 设置配置信息。

show - 显示信息。

tcp - 更改到 `netsh interface tcp' 上下文。

teredo - 更改到 `netsh interface teredo' 上下文。

udp - 更改到 `netsh interface udp' 上下文。

此上下文中的命令:

? - 显示命令列表。

add - 在一个表格中添加一个配置项。

delete - 从一个表格中删除一个配置项。

dump - 显示一个配置脚本。

help - 显示命令列表。

install - 安装 IP 协议。

reset - 重置 IP 配置。

set - 设置配置信息。

show - 显示信息。

uninstall - 卸载 IP 协议。

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

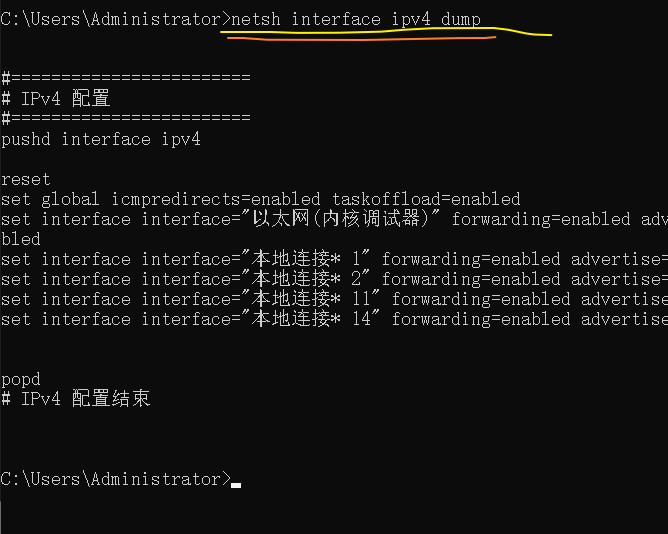
**2.4 执行netsh系列命令的方法**

（1）不进入上下文环境，**直接执行**

对于并非上下文环境的命令，比如【netsh interface】就是上下文命令，而【netsh dump】不是上下文命令，可以不用在上下文环境中执行。



再比如【netsh interface ipv4】是上下文命令，而【netsh interface ipv4 dump】不是，就可以直接执行。



（2）进入上下文环境，再执行

还是以上面的【netsh dump】为例，要实现相同的效果，除了直接执行外，还可以先通过【netsh】进入netsh上下文再执行【dump】。

**三、显示所有配置信息——dump**

**3.1 打印帮助信息**

执行命令【netsh dump /?】，即可显示帮助信息。

C:\WINDOWS\system32>netsh dump /?

用法: dump

说明:

创建一个包含当前配置的脚本文件。如果保存到

文件，此脚本可以用来还原更改的配置设置。

**3.2 显示所有配置信息（netsh dump）**

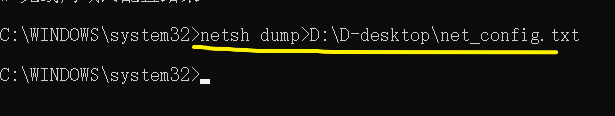
既然可以显示所有配置信息，那么将配置信息重定向至文本文件，也就实现了将所有网络配置信息导出至本地。

执行命令【netsh dump】，显示所有配置信息



**3.3 将网络配置信息导出至本地文件（netsh dump>filepath）**

执行命令【netsh dump>D:\D-desktop\net\_config.txt】，即可将上面这么多行的配置信息全导入至txt文件中。





**四、进入或更改netsh interface——netsh int**

**4.1 打印帮助信息（netsh int /?）**

C:\WINDOWS\system32>netsh int /?

下列指令有效:

此上下文中的命令:

6to4 - 更改到 `netsh interface 6to4' 上下文。

? - 显示命令列表。

dump - 显示一个配置脚本。

help - 显示命令列表。

httpstunnel - 更改到 `netsh interface httpstunnel' 上下文。

ipv4 - 更改到 `netsh interface ipv4' 上下文。

ipv6 - 更改到 `netsh interface ipv6' 上下文。

isatap - 更改到 `netsh interface isatap' 上下文。

portproxy - 更改到 `netsh interface portproxy' 上下文。

set - 设置配置信息。

show - 显示信息。

tcp - 更改到 `netsh interface tcp' 上下文。

teredo - 更改到 `netsh interface teredo' 上下文。

udp - 更改到 `netsh interface udp' 上下文。

下列的子上下文可用:

6to4 httpstunnel ipv4 ipv6 isatap portproxy tcp teredo udp

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

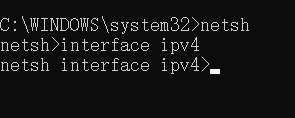
**4.2 进入到ipv4接口环境（int ip）**

因为nterface命令是需要在上下文环境中执行的；必须先执行【netsh】进入上下文。

而后再执行【interface】，即可进入netsh interface的上下文，而再进一步执行【ip】或【ipv4】才可进入netsh interface ipv4上下文。



当然也可在进入netsh上下文后，直接用【int ip】、【int ipv4】、【interface ip】、【interface ipv4】这4个命令之一进入netsh interface ipv4上下文。



进入到ipv4上下文，有什么意义？

这时候，执行命令【dump】，只会显示与ipv4相关的配置信息（截取全部的一部分）。

netsh interface ipv4>dump

#========================

# IPv4 配置

#========================

pushd interface ipv4

reset

set global taskoffload=enabled

set interface interface="以太网(内核调试器)" forwarding=enabled advertise=enabled nud=enabled ignoredefaultroutes=disabled

set interface interface="本地连接\* 1" forwarding=enabled advertise=enabled nud=enabled ignoredefaultroutes=disabled

set interface interface="本地连接\* 2" forwarding=enabled advertise=enabled nud=enabled ignoredefaultroutes=disabled

set interface interface="本地连接\* 11" forwarding=enabled advertise=enabled nud=enabled ignoredefaultroutes=disabled

set interface interface="本地连接\* 14" forwarding=enabled advertise=enabled nud=enabled ignoredefaultroutes=disabled

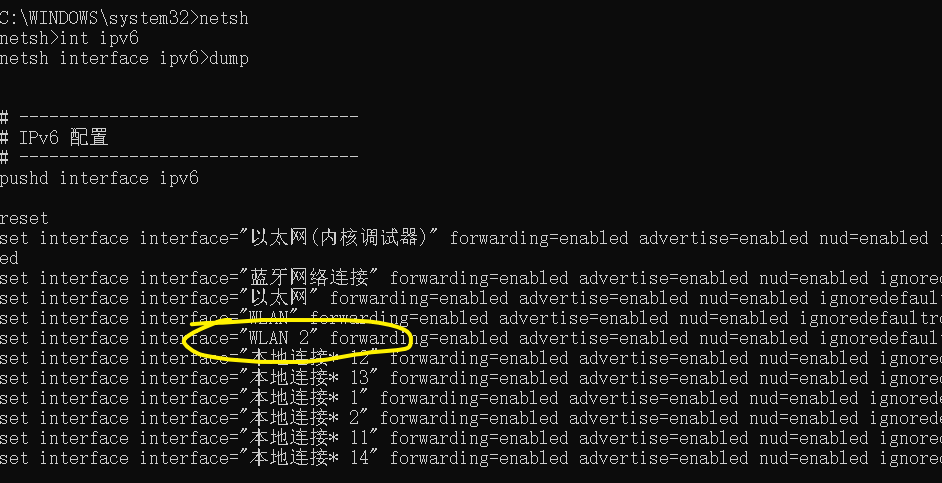
1. popd
2. # IPv4 配置结束

**4.3 进入IPv6接口环境（int ipv6）**

由于现在大部分的网络都已经升级成了IPv6，所以很可能你不会在ipv4接口环境中找到你想要的网络配置信息。

比如我的这台机器上网实际上用的是wlan 2，但是在ipv4接口环境下却输出不出来。

当我进入ipv6环境后，再输出所有配置信息，里面就包括了wlan 2。



而非常诡异的是，虽然是ipv6，但是也可以在ipv4的接口环境中设置IP地址（详见下面的5.3）。

**五、设置IP地址及其获取方式**

**5.1 打印帮助信息（set addr /?）**

在进入了netsh上下文后，如果在ipv4接口环境中，执行命令【set addr /?】，显示帮助

netsh interface ipv4>set addr /?

用法: set address [name=]<字符串>

[[source=]dhcp|static]

[[address=]<IPv4 地址>[/<整数>] [[mask=]<IPv4 掩码>]

[[gateway=]<IPv4 地址>|none [gwmetric=]<整数>]

[[type=]unicast|anycast]

[[subinterface=]<字符串>]

[[store=]active|persistent]

参数:

标记 值

name - 接口名称或索引。

source - 下列值之一:

dhcp: 启用 DHCP，以便为指定的接口

配置 IP 地址。

static: 禁用 DHCP，以便为指定的接口

配置 IP 地址。如果配置了

地址或网关，则必须指定

该值。

address - 要添加或修改的 IPv4 地址，

后边也可以加上子网前缀长度。

mask - 指定的 IP 地址的 IP 子网掩码。

gateway - 下列值之一:

<IPv4 address>: 正在设置的静态 IP 地址的

特定默认网关。

none: 未设置任何默认网关。此为默认值。

gwmetric - 默认网关的跃点数。仅在指定了网关时

才应设置该字段。

type - 下列值之一:

unicast: 将地址标记为单播地址。

此为默认值。

anycast: 将地址标记为任意广播地址。

subinterface - 默认网关存在于其上的子接口的 LUID。

此参数仅用于带有

多个子接口的接口。

store - 下列值之一:

active: 设置仅持续到下一次启动。

persistent: 设置永久有效。此为默认值。

备注: 用来启用或禁用 IP 地址配置的 DHCP。

同时删除所有以前的静态 IP 地址和默认网关，

并可以添加新的静态 IP 地址和默认网关。

示例:

set address name="Wired Ethernet Connection" source=dhcp

set address "Wired Ethernet Connection" static 10.0.0.9 255.0.0.0 10.0.0.1 1

在进入了netsh上下文后，如果在ipv6接口环境中，执行命令【set addr /?】，显示帮助

netsh interface ipv6>set addr /?

用法: set address [interface=]<string> [address=]<IPv6 address>

[[type=]unicast|anycast]

[[validlifetime=]<integer>|infinite]

[[preferredlifetime=]<integer>|infinite]

[[store=]active|persistent]

参数:

标记 值

interface - 接口名称或索引。

address - 要修改的 IPv6 地址。

type - 下列其中一个值:

unicast: 将地址标记为 unicast 地址。

此为默认值。

anycast: 将地址标记为 anycast 地址。

validlifetime - 有效地址的生存时间。

默认值是无限。

preferredlifetime - 首选地址生存时间。

默认值是无限。

store - 下列其中一个值:

active: 设置仅持续到下一次启动。

persistent: 设置永久有效(默认值)。

说明: 修改给定接口上的 IPv6 地址。时间值可以为天、小时、分钟和秒，

例如 1d2h3m4s。

示例:

set address "Private" fe80::2 anycast

**5.2 设置IP地址的基本语法**

从5.1打印出来的帮助信息也可知道，如果是设置ipv4的地址。

在进入netsh interface ipv4上下文后，执行下面的命令可以设置静态地址。

set address name=xx source=static addr=xx mask=xx

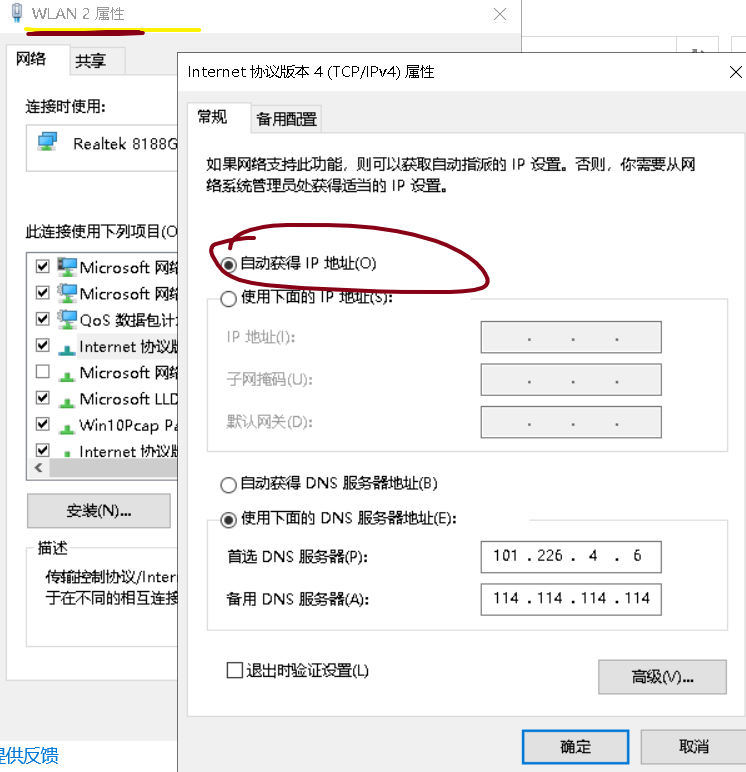
在进入netsh interface ipv4上下文后，执行下面的命令可以设置动态地址。

set address name=xx source=dhcp

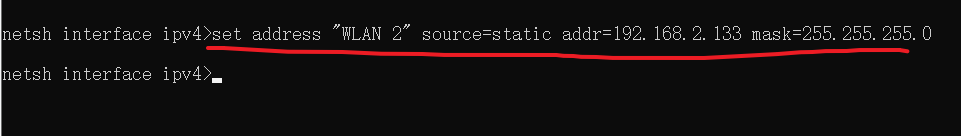
而如果是设置ipv6的地址，本文暂时不涉及，抱歉了！

**5.3 设置静态IP地址**

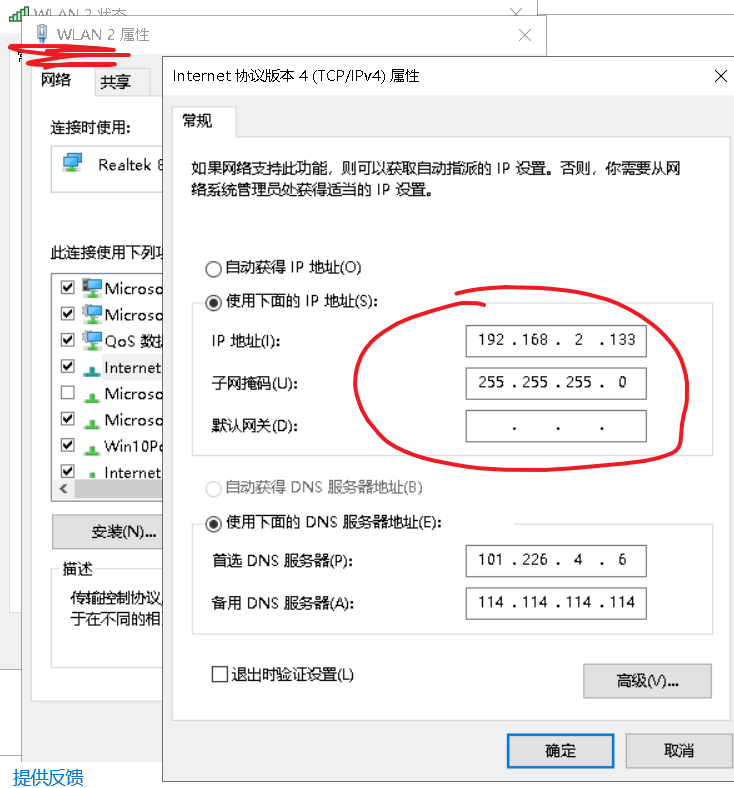
首先，我们可以看到WLAN 2的IP地址是自动获取的。



为了将其调整成静态获取地址，并且手动为其制定IP地址和子网掩码，我们可以执行命令【set address name="WLAN 2" source=static addr=192.168.2.133 mask=255.255.255.0】



 效果如下，修改后能不能正常上网我不知道，反正我是知道修改成功了。

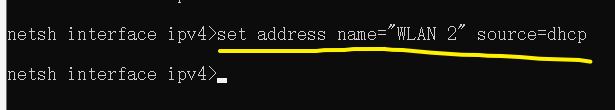


**5.4 设置动态IP地址**

 由于网络和共享中心里，修改成静态地址后，显示无法连接到网络，所以我想改回动态地址。



 在ipv4接口环境下，执行命令【set address name="WLAN 2" source=dhcp】



 碎碎念的自动获取地址，又回来啦！

|  |
| --- |
| https://img-blog.csdnimg.cn/175eeecc2dd8456f8753fa92befc4bbc.png |

**六、使用netsh修复网络故障——winsock**

**6.1 重置Winsock目录**

查看一下winsock命令的帮助信息，执行命令【netsh winsock /?】，可得

C:\Users\Administrator>netsh winsock /?

下列指令有效:

此上下文中的命令:

? - 显示命令列表。

audit - 显示已经安装和删除的 Winsock LSP 列表。

dump - 显示一个配置脚本。

help - 显示命令列表。

remove - 从系统中删除 Winsock LSP。

reset - 重置 Winsock 目录为清除状态。

set - 设置 Winsock 选项。

show - 显示信息。

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

如果计算机Winsock协议配置有问题的话将会导致网络连接等问题，通过重置Winsock目录、初始化网络环境借以恢复网络，可解决由于“软件冲突”、“病毒原因”造成的参数错误问题，可以用命令【netsh winsock reset】。

**6.2 重置TCP/IP协议**

如果我们要重置TCP/IP协议，卸载并重新安装TCP/IP协议，使其恢复到初次安装操作系统时的状态，并且将重置过程日志信息输出到本地，可以用命令【netsh int ip reset>c:\ip\_reset\_log.txt】。

**七、借netsh命令管理防火墙（旧firewall）**

**7.1 管理防火墙的GUI界面**

我们先不着急讲如何通过命令行管理防火墙，因为大多数人恐怕更喜欢用图形界面窗口去进行配置！

在运行窗口或CMD窗口中执行【firewall.cpl】，即可打开「Windows Defender 防火墙」窗口。

|  |
| --- |
| https://img-blog.csdnimg.cn/bf8f117401224680b1551d9ed9c2ecaf.png |

而后在高级设置里，窗口名为「高级安全Vindows Defender防火墙」，我们可以添加各种规则来细化地设置防火墙。

|  |
| --- |
| https://img-blog.csdnimg.cn/16c6d5ce97e04bb1b91eb90c8586443a.png |

**7.2 打印netsh firewall的帮助信息**

执行【netsh firewall /?】，即可大致获得旧防火墙命令的子命令。

C:\WINDOWS\system32>netsh firewall /?

下列指令有效:

此上下文中的命令:

? - 显示命令列表。

add - 添加防火墙配置。

delete - 删除防火墙配置。

dump - 显示一个配置脚本。

help - 显示命令列表。

set - 设置防火墙配置。

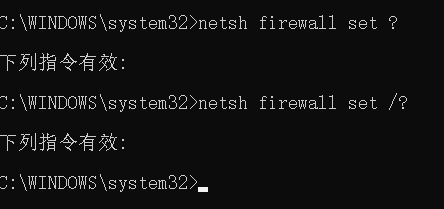
show - 显示防火墙配置。

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

而后我们再看看子命令怎么用？

以set子命令为例！我们执行【netsh firewall set /?】，发现没有帮助信息，从这个方面我们也可看出微软公司已经停止了对该命令的维护。



但是，惊奇的是，我们执行【netsh firewall set portopening /?】 却能打印出set命令的子命令portopenning的帮助信息。

C:\WINDOWS\system32>netsh firewall set portopening /?

set portopening

[ protocol = ] TCP|UDP|ALL

[ port = ] 1-65535

[ [ name = ] name

[ mode = ] ENABLE|DISABLE

[ scope = ] ALL|SUBNET|CUSTOM

[ addresses = ] addresses

[ profile = ] CURRENT|DOMAIN|STANDARD|ALL

设置防火墙端口配置。

参数:

protocol - 端口协议。

TCP - 传输控制协议(TCP)。

UDP - 用户数据报协议(UDP)。

ALL - 所有协议。

port - 端口号。

name - 端口名(可选)。

mode - 端口模式(可选)。

ENABLE - 允许通过防火墙(默认值)。

DISABLE - 不允许通过防火墙。

scope - 端口范围(可选)。

ALL - 允许所有流量通过防火墙(默认值)。

SUBNET - 只允许本地网络(子网)流量通过防火墙。

CUSTOM - 只允许指定的流量通过防火墙。

addresses - 自定义范围地址(可选)。

这个范围可以包含 IPv4 地址、

IPv6 地址、子网、范围或关键字 LocalSubnet，用逗号分隔。

profile - 配置文件(可选)。

CURRENT - 适用于活动配置文件。活动配置文件可以是域、

标准(即专用)，或公用(默认)。

DOMAIN - 适用于域配置文件。

STANDARD - 适用于标准(即专用)配置文件。

ALL - 适用于域和标准(即专用)配置文件。

不适用于公用配置文件。

注释: "scope" 必须是 "CUSTOM" 才能指定 "addresses"。

"addresses" 不能包含未指定的或环回地址。

示例:

set portopening TCP 80 "My Web Port"

set portopening UDP 500 IKE ENABLE ALL

set portopening ALL 53 DNS ENABLE CUSTOM

157.60.0.1,172.16.0.0/16,10.0.0.0/255.0.0.0,

12AB:0000:0000:CD30::/60,LocalSubnet

set portopening protocol=ALL port=53 name=DNS mode=ENABLE scope=CUSTOM

addresses=157.60.0.1,172.16.0.0/16,10.0.0.0/255.0.0.0,

12AB:0000:0000:CD30::/60,LocalSubnet

重要信息: "netsh firewall" 已弃用；

请改用 "netsh advfirewall firewall"。

有关使用 "netsh advfirewall firewall" 命令

而非 "netsh firewall" 的详细信息，请参阅

https://go.microsoft.com/fwlink/?linkid=121488 上的 KB 文章 947709。

我们从帮助信息的最后几行可以看到「重要信息」，也即netsh firewall已经被弃用；但是其中有些命令在win10的系统中还是能用的。

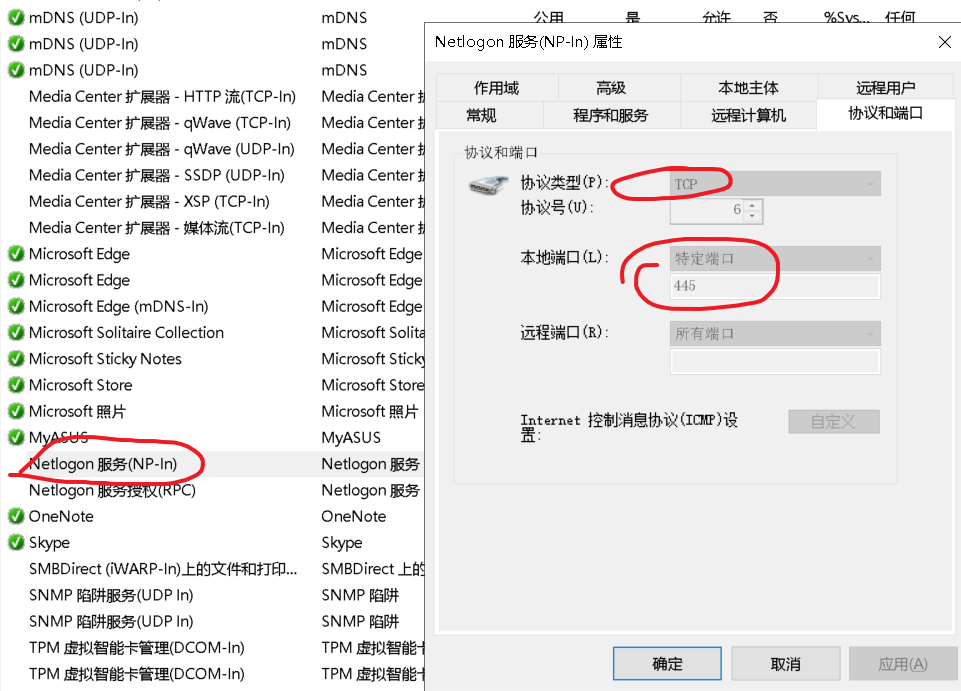
因此我们还是稍微讲讲，（让大家念念旧），哈哈哈。

**7.3 添加、删除防火墙过滤规则端口（set portopenning）**

执行命令【netsh firewall set portopening TCP 445 ENABLE】，结果如下



而后，我们可以在本文的7.1小节中提到的「高级安全Vindows Defender防火墙」窗口中，找到上面设置的这个端口规则。



最后，我们把这个端口过滤规则关掉，毕竟俺就是想测试一下打不打得开总归要关得万一遇到坏人呢。

执行命令【netsh firewall set portopening TCP 445 DISABLE】，果然就找不到那个netlogon服务了。



**7.4 打开或关闭防火墙——set opmode**

 执行命令【netsh firewall set opmode /?】，获得相关帮助信息

C:\WINDOWS\system32>netsh firewall set opmode /?

set opmode

[ mode = ] ENABLE|DISABLE

[ [ exceptions = ] ENABLE|DISABLE

[ profile = ] CURRENT|DOMAIN|STANDARD|ALL

设置防火墙操作配置。

参数:

mode - 操作模式。

ENABLE - 启用防火墙。

DISABLE - 禁用防火墙。

exceptions - 例外模式(可选)。

ENABLE - 允许通过防火墙(默认值)。

DISABLE - 不允许通过防火墙。

profile - 配置文件(可选)。

CURRENT - 适用于活动配置文件。活动配置文件可以是域、

标准(即专用)，或公用(默认)。

DOMAIN - 适用于域配置文件。

STANDARD - 适用于标准(即专用)配置文件。

ALL - 适用于域和标准(即专用)配置文件。

不适用于公用配置文件。

示例:

set opmode ENABLE

set opmode mode=ENABLE exceptions=DISABLE

#### 打开防火墙（旧命令）：netsh firewall set opmode mode=enable

#### 打开防火墙（新命令）：netsh advfirewall set currentprofile state on

#### 关闭防火墙（旧命令）：netsh firewall set opmode mode=disable

#### 关闭防火墙（新命令）：netsh advfirewall set currentprofile state off

重要信息:"netsh firewall" 已弃用；

请改用 "netsh advfirewall firewall"。

有关使用 "netsh advfirewall firewall" 命令

而非 "netsh firewall" 的详细信息，请参阅

https://go.microsoft.com/fwlink/?linkid=121488 上的 KB 文章 947709。

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

在执行下面的命令前，我们看一下当前防火墙的状态

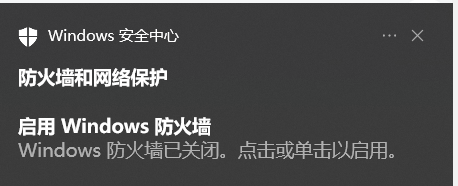


再执行【netsh firewall set opmode mode=enable】，即打开和启用防火墙



最后执行命令【netsh firewall set opmode mode=disable】，即关闭防火墙

执行完后，弹出下面的警示框，也侧面说明成功关闭。



**八、借netsh命令管理防火墙（新firewall）**

**8.1 打印新防火墙命令帮助信息**

执行命令【netsh advfirewall firewall】，得到最外层命令的子命令列表

C:\WINDOWS\system32>netsh advfirewall firewall

下列指令有效:

此上下文中的命令:

? - 显示命令列表。

add - 添加新入站或出站防火墙规则。

delete - 删除所有匹配的防火墙规则。

dump - 显示一个配置脚本。

help - 显示命令列表。

set - 为现有规则的属性设置新值。

show - 显示指定的防火墙规则。

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

接着以子命令set为例，执行【netsh advfirewall firewall set ?】，得到如下帮助信息

C:\WINDOWS\system32>netsh advfirewall firewall set ?

下列指令有效:

此上下文中的命令:

set rule - 为现有规则的属性设置新值。

而后，再执行【netsh advfirewall firewall set rule ?】，得到如下帮助信息 。其中的group字段和name字段表示对规则进行命名，；而dir字段表示规则是入站规则还是出站规则；profile字段表示配置文件，应用于公有网络还是专有网络，还是特定IP。

C:\WINDOWS\system32>netsh advfirewall firewall set rule ?

用法: set rule

group=<string> | name=<string>

[dir=in|out]

[profile=public|private|domain|any[,...]]

[program=<program path>]

[service=service short name|any]

[localip=any|<IPv4 address>|<IPv6 address>|<subnet>|<range>|<list>]

[remoteip=any|localsubnet|dns|dhcp|wins|defaultgateway|

<IPv4 address>|<IPv6 address>|<subnet>|<range>|<list>]

[localport=0-65535|<port range>[,...]|RPC|RPC-EPMap|IPHTTPS|any]

[remoteport=0-65535|<port range>[,...]|any]

[protocol=0-255|icmpv4|icmpv6|icmpv4:type,code|icmpv6:type,code|

tcp|udp|any]

new

[name=<string>]

[dir=in|out]

[program=<program path>

[service=<service short name>|any]

[action=allow|block|bypass]

[description=<string>]

[enable=yes|no]

[profile=public|private|domain|any[,...]]

[localip=any|<IPv4 address>|<IPv6 address>|<subnet>|<range>|<list>]

[remoteip=any|localsubnet|dns|dhcp|wins|defaultgateway|

<IPv4 address>|<IPv6 address>|<subnet>|<range>|<list>]

[localport=0-65535|RPC|RPC-EPMap|any[,...]]

[remoteport=0-65535|any[,...]]

[protocol=0-255|icmpv4|icmpv6|icmpv4:type,code|icmpv6:type,code|

tcp|udp|any]

[interfacetype=wireless|lan|ras|any]

[rmtcomputergrp=<SDDL string>]

[rmtusrgrp=<SDDL string>]

[edge=yes|deferapp|deferuser|no (default=no)]

[security=authenticate|authenc|authdynenc|notrequired]

备注:

- 为已识别的规则设置新的参数值。如果规则不存在，

则该命令失败。若要创建规则，请使用添加命令。

- 会更新规则中 new 关键字后的值。如果

没有值，或缺少关键字 new，则没有任何更改。

- 一组规则只能被启用或禁用。

- 如果多个规则与条件匹配，则会

更新所有匹配规则。

- 规则名称应该是唯一的，并且不能是 "all"。

- 如果指定远程计算机或用户组，security 必须

为 authenticate、authenc 或 authdynenc。

- 为 authdynenc 设置安全性可允许系统动态协商为匹配

给定 Windows Defender 防火墙规则的通信使用加密。

根据现有连接安全规则属性协商加密。

选择此选项后，只要入站 IPsec 连接已设置安全保护，

但未使用 IPsec 进行加密，计算机就能够接收该入站连接的第一个 TCP

或 UDP 包。

一旦处理了第一个数据包，服务器将重新协商连接并对其进行升级，以便

所有后续通信都完全加密。

- Authdynenc 仅当 dir=in 时有效。

- 如果 action=bypass，则当 dir=in 时必须指定远程计算机组。

- 如果 service=any，则规则只适用于服务。

- ICMP 类型或代码可以是 "any"。

- 只能为入站规则指定边缘。

示例:

根据名称为 "allow80" 的规则更改远程 IP 地址:

netsh advfirewall firewall set rule name="allow80" new

remoteip=192.168.0.2

启用带有分组字符串 "Remote Desktop" 的组:

netsh advfirewall firewall set rule group="remote desktop" new

enable=yes

为 udp- 更改规则 "Allow port range" 上的本地端口

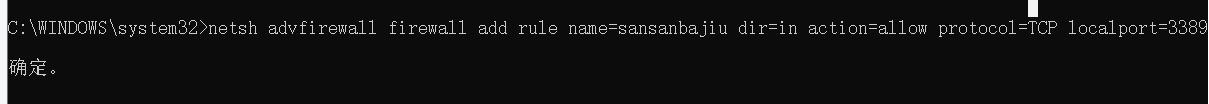
Set rule name="Allow port range" dir=out protocol=udp localport=5000-5020 action=allow

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

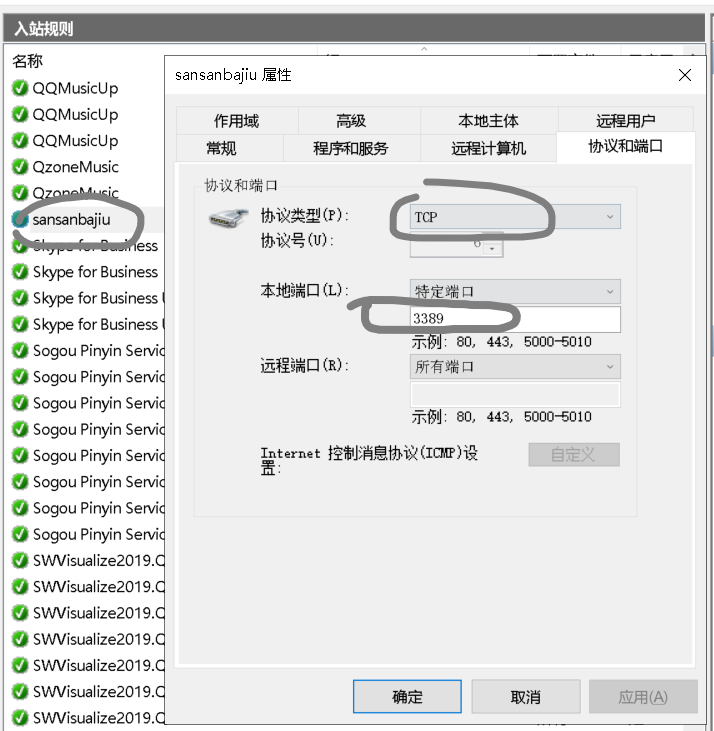
**8.2 利用netsh防火墙命令构建一条防火墙规则**

如果我们要开启端口3389，协议限定为TCP，规则为入站规则。

执行命令【netsh advfirewall firewall add rule name=sansanbajiu dir=in  
action=allow protocol=TCP localport=3389】



成功后，我们在「高级安全Vindows Defender防火墙」中也可以看出来这个名为sansanbajiu的入站规则被开启了。



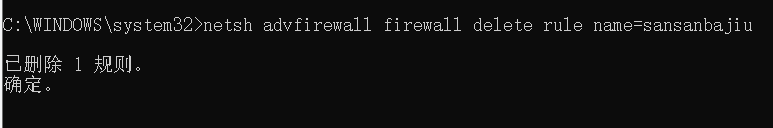
**8.2 利用netsh防火墙命令删除一条规则**

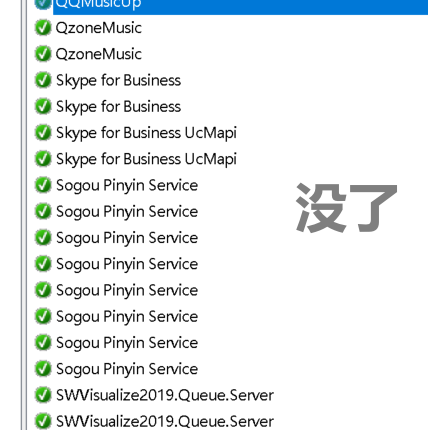
删除规则就没有构建规则那么复杂，因为同名规则只有一个，因此删除规则直接指定名称即可。

执行命令【netsh advfirewall firewall delete rule name=sansanbajiu】

如果您想更精确一些，删除的时候无需指定action字段，其他的端口字段、协议字段都可以添加，以进一步精确地指定欲删除的规则。

执行命令【netsh advfirewall firewall delete rule name=sansanbajiu protocol=TCP localport=3389】





**九、借netsh命令破解WIFI密码**

**9.1 查看netsh wlan命令的帮助信息**

C:\WINDOWS\system32>netsh wlan /?

下列指令有效:

此上下文中的命令:

? - 显示命令列表。

add - 在一个表格中添加一个配置项。

connect - 连接到无线网络。

delete - 从一个表格中删除一个配置项。

disconnect - 从无线网络断开连接。

dump - 显示一个配置脚本。

export - 将 WLAN 配置文件保存为 XML 文件。

help - 显示命令列表。

IHV - 用于 IHV 记录的命令。

refresh - 刷新承载网络设置。

reportissues - 生成 WLAN 智能跟踪报告。

set - 设置配置信息。

show - 显示信息。

start - 启动承载网络。

stop - 停止承载网络。

若需要命令的更多帮助信息，请键入命令，接着是空格，

后面跟 ?。

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

以show子命令为例，查询show子命令怎么用。执行命令【netsh wlan show /?】

C:\WINDOWS\system32>netsh wlan show /?

下列指令有效:

此上下文中的命令:

show all - 显示完整的无线设备和网络信息。

show allowexplicitcreds - 显示允许共享用户凭据设置。

show autoconfig - 显示是否启用或禁用自动配置逻辑。

show blockednetworks - 显示阻止的网络显示设置。

show createalluserprofile - 显示是否允许所有人创建所有

用户配置文件。

show drivers - 显示系统上无线 LAN 驱动程序的属性。

show filters - 显示允许和阻止的网络列表。

show hostednetwork - 显示承载网络的属性和状态。

show interfaces - 显示系统上的无线局域网接口

的列表。

show networks - 显示计算机上可见的网络列表。

show onlyUseGPProfilesforAllowedNetworks - 显示在配置 GP 的网络设置上仅使用 GP 配置文件。

show profiles - 显示计算机上配置的配置文件列表。

show randomization - 显示 MAC 随机化是已启用还是

已禁用。

show settings - 显示无线 LAN 的全局设置。

show tracing - 显示是否启用或禁用无线局域网跟踪。

show wirelesscapabilities - 显示系统的无线功能

show wlanreport - 生成一个报告，显示最新的无线会话信息。

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

再以Profiles为例，进一步想知道show profiles的用法，执行命令【netsh wlan show profiles /?】

C:\WINDOWS\system32>netsh wlan show profiles /?

用法: show profiles [[name=]<string>] [interface=<string>] [key=<string>]

参数:

标记 值

name - 所要显示配置文件的名称。

interface - 已配置此配置文件的接口的名称。

key - 以纯文件显示密钥，设置密钥=clear。

注释:

显示配置文件数据或列出系统上的配置文件。

参数 name 和 interface 都是可选的。

如果已给定配置文件名称，则将会显示配置文件的内容。否则，将仅列出配置文件

名称和描述。

如果已给定接口名称，则将仅列出给定接口上的指定配置文件。否则，将列出

系统中给定名称的所有配置文件。

如果将密钥设置为 "clear"，且呼叫方为本地管理员，

则该密钥将以纯文本显示。

组策略配置文件为只读，而用户配置文件是

可读写的，而且，首选顺序也可以更改。

示例:

show profiles name="profile 1" interface="Wireless Network Connection"

show profiles name="profile 2"

show profiles name="profile 3" key=clear

show profiles

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

这时候显然就是走到头了，已经不能再深挖了。

**9.2  查看已经保存的wifi网络（不包括密码）**

（1）查看所有接口上已经保存的WIFI网络

执行命令【netsh wlan show profiles】，查看当前系统已经保存的网络

C:\WINDOWS\system32>netsh wlan show profile

接口 WLAN 上的配置文件:

组策略配置文件(只读)

---------------------------------

<无>

用户配置文件

-------------

所有用户配置文件 : 醉生梦死

所有用户配置文件 : @PHICOMM\_06\_5G

所有用户配置文件 : @PHICOMM\_32

接口 WLAN 2 上的配置文件:

组策略配置文件(只读)

---------------------------------

<无>

用户配置文件

-------------

所有用户配置文件 : @PHICOMM\_32

所有用户配置文件 : @PHICOMM\_06\_5G

所有用户配置文件 : 醉生梦死

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

（2）查看指定接口上的已保存的WIFI网络

执行命令【netsh wlan show profile interface="WLAN 2"】



**9.3 查看指定wifi的密码**

假如我们要显示WLAN接口下保存的「@PHICOMM\_32」的密码，可以执行命令【netsh wlan show profile name="@PHICOMM\_32" key=clear】

C:\WINDOWS\system32>netsh wlan show profile name="@PHICOMM\_32" key=clear

接口 WLAN 上的配置文件 @PHICOMM\_32:

=======================================================================

已应用: 所有用户配置文件

配置文件信息

-------------------

版本 : 1

类型 : 无线局域网

名称 : @PHICOMM\_32

控制选项 :

连接模式 : 自动连接

网络广播 : 只在网络广播时连接

AutoSwitch : 请勿切换到其他网络

MAC 随机化: 禁用

连接设置

---------------------

SSID 数目 : 1

SSID 名称 :“@PHICOMM\_32”

网络类型 : 结构

无线电类型 : [ 任何无线电类型 ]

供应商扩展名 : 不存在

安全设置

-----------------

身份验证 : WPA2 - 个人

密码 : CCMP

身份验证 : WPA2 - 个人

密码 : GCMP

安全密钥 : 存在

关键内容 : as69865069847

费用设置

-------------

费用 : 无限制

阻塞 : 否

接近数据限制 : 否

过量数据限制 : 否

漫游 : 否

费用来源 : 默认

接口 WLAN 2 上的配置文件 @PHICOMM\_32:

=======================================================================

已应用: 所有用户配置文件

配置文件信息

-------------------

版本 : 1

类型 : 无线局域网

名称 : @PHICOMM\_32

控制选项 :

连接模式 : 自动连接

网络广播 : 只在网络广播时连接

AutoSwitch : 请勿切换到其他网络

MAC 随机化: 禁用

连接设置

---------------------

SSID 数目 : 1

SSID 名称 :“@PHICOMM\_32”

网络类型 : 结构

无线电类型 : [ 任何无线电类型 ]

供应商扩展名 : 不存在

安全设置

-----------------

身份验证 : WPA2 - 个人

密码 : CCMP

身份验证 : WPA2 - 个人

密码 : GCMP

安全密钥 : 存在

费用设置

-------------

费用 : 无限制

阻塞 : 否

接近数据限制 : 否

过量数据限制 : 否

漫游 : 否

费用来源 : 默认

# 十、netsh命令设置端口

## [windows命令行下用netsh实现端口转发(端口映射)](https://www.cnblogs.com/xiaozi/p/9453944.html)

## **window有端口转发功能：任何传入到本地端口的TCP连接（IPv4或IPv6）都可以被重定向到另一个本地端口，或远程计算机上的端口，并且系统不需要有一个专门用于侦听该端口的服务。**

微软Windows的netsh是一个命令行脚本实用工具。使用netsh工具 ，可以查看或更改本地计算机或远程计算机的网络配置。不仅可以在本地计算机上运行这些命令，而且可以在网络上的远程计算机上运行。

可以手动运行Netsh命令，或创建批处理文件或脚本实现过程的自动化。netsh提供了脚本功能，让您在批处理模式下针对指定的计算机，运行一组命令。利用netsh ，可以将配置脚本保存为文本文件，便于存档或用于配置其他的计算机。

netsh命令已经推出很长时间，在Windows 2000/XP/2003中均带有Netsh命令。Windows Server 2008只是对netsh的参数项做了些扩展。

**配置方法：**

假定需要通过192.168.204.150的**3389端口转发到3340端口**，则需要在192.168.204.150主机的命令行输入如下语句：

netsh interface portproxy add v4tov4 listenport=3340 listenaddress=192.168.204.150 connectport=3389 connectaddress=192.168.204.150

可以查看存在的转发

netsh interface portproxy show all

删除指定规则

netsh interface portproxy delete v4tov4 listenport=3340 listenaddress=192.168.204.150

使用netstat确保3340端口当前处于被侦听状态：

netstat -ano | findstr :3340



禁用系统防火墙

netsh firewall set opmode disable

启用防火墙

netsh firewall set opmode enable

|  |
| --- |
|  |

# 参考文档3

netsh全称是network shell，它包括有线与无线的网络设置，是非常方便的工具 ，而且可以避免调用api的复杂编程，因为api里有一大把复杂的结构，C++资源Com调用等。

连接一个wifi，往往会要求输入密码和设置是否自动连接，下次就会用保存的密码自动连接了，这就需要为每个wifi热点保存配置文件，就是profile。

**1、显示本机保存的profiles**

列出配置文件

netsh wlan show profiles

配置文件是以wifi的ssid命名的，如执行命令后没有显示，说明没有保存有profile文件，连接WiFi成功后，系统会保存该profile。

**2、连接某个配置文件**

用  netsh wlan connect name=profileName

连接其中一个profile，profileName   一定是本机已保存的wifi配置文件，而不能是一个没有连接过的，没有保存配置文件的wifi名。就是profileName    只能是netsh wlan show profiles命令查询得到的profiles。

netsh wlan  connect  name=VIPWIFI  
已成功完成连接请求。

netsh wlan  connect  name=Yunmedia7  
没有分配给指定接口的配置文件“Yunmedia7”。

**3、断开连接**

netsh wlan disconnect

**4、删除一个配置文件**

netsh wlan  delete profile name="VIPWIFI"  
已从接口“WLAN”中删除配置文件“VIPWIFI”。

profileName   可以打引号也可以不打引号。

**5、导入配置文件**

netsh wlan add profile filename="wifi.xml"  
注意：这里面参数"wifi.xml"是xml文件名，默认路径为当前目录，添加成功后提示：已将配置文件 VIPWIFI添加到接口 WLAN。

**6、查看配置文件的具体信息**

netsh wlan show  profile  name ="VIPWIFI"

接口 WLAN 上的配置文件 VIPWIFI:  
=======================================================================

已应用: 所有用户配置文件

配置文件信息  
-------------------  
    版本                   : 1  
    类型                   : 无线局域网  
    名称                   : VIPWIFI  
    控制选项               :  
        连接模式           : 手动连接  
        网络广播           : 只在网络广播时连接  
        AutoSwitch         : 请勿切换到其他网络  
        MAC 随机化: 禁用

连接设置  
---------------------  
    SSID 数目              : 1  
    SSID 名称              :“VIPWIFI”  
    网络类型               : 结构  
    无线电类型             : [ 任何无线电类型 ]  
    供应商扩展名           : 不存在

安全设置  
-----------------  
    身份验证         : WPA2 - 企业  
    密码                 : CCMP  
    身份验证         : WPA2 - 企业  
    密码                 : GCMP  
    安全密钥               : 不存在  
    802.1X                 : Enabled  
    EAP 类型               : Microsoft: 智能卡或其他证书  
    802.1X auth 凭据       : 计算机或用户凭据  
    缓存用户信息           : 是

费用设置  
-------------  
    费用                : 无限制  
    阻塞                : 否  
    接近数据限制        : 否  
    过量数据限制        : 否  
    漫游                : 否  
    费用来源            : 默认

查看wifi密码的命令：

netsh wlan show  profile  name ="VIPWIFI"  key=clear

**7、设置手动连接和自动连接**

Netsh WLAN set profileparameter name="Profile\_Name" connectionmode=manual  
Netsh WLAN set profileparameter name=" Profile\_Name" connectionmode=auto

**8、导出配置文件**

将一个特定的无线网络配置文件，导出到指定位置：

Netsh WLAN export profile name=“profileName”  key=clear  folder=“filePath”

export命令为计算机上存储的每个无线网络配置文件创建一个XML文件。请注意，我们使用的命令有key=clear参数，它将触发在XML文件中，包含每个网络的网络安全密钥，因此请确保将这些记录保存在安全的地方，因为使用记事本可以很容易地查看或编辑这些记录。

然后，可以使用导出的XML配置文件，使用以下命令将每个无线配置文件设置导入到所有接口和用户：

Netsh WLAN add profile filename=“文件\_路径.XML"

或者，如果要将配置文件导入特定的无线接口且仅导入当前用户，则可以使用以下命令：

Netsh WLAN add profile filename=“路径\_With\_文件名.xml“  Interface=”Wireless Network Connection“  用户=当前

netsh  wlan  export  profile name=VIPWIFI   folder=D:\8021xxml

接口配置文件“VIPWIFI”已成功保存在文件“D:\8021xxml\WLAN-VIPWIFI.xml”中。

**查看列出网卡信息接口**

**无线网卡信息**

netsh wlan  show interfaces

系统上有 1 个接口:

    名称                   : WLAN  
    描述                   : Realtek RTL8822BE 802.11ac PCIe Adapter  
    GUID                   : 8dbaff05-5923-4726-b406-1631dd0e2742  
    物理地址               : c0:b5:d7:37:a8:19  
    状态                   : 已连接  
    SSID                   : VIPKID  
    BSSID                  : 94:28:2e:53:0a:02  
    网络类型               : 结构  
    无线电类型             : 802.11ac  
    身份验证               : WPA2 - 企业  
    密码                   : CCMP  
    连接模式               : 配置文件  
    信道                   : 48  
    接收速率(Mbps)         : 360  
    传输速率 (Mbps)        : 360  
    信号                   : 100%  
    配置文件               : VIPWIFI

    承载网络状态  : 不可用

**有线网卡信息**

netsh lan  show interfaces

系统有 4 个接口:

    名称             : VMware Network Adapter VMnet1  
    描述             : VMware Virtual Ethernet Adapter for VMnet1  
    GUID             : 140dc0a9-7ac2-4e11-a6f5-9ae896389ea1  
    物理地址         : 00-50-56-C0-00-01  
    状态             : 已连接。网络不支持身份验证。

    名称             : VMware Network Adapter VMnet8  
    描述             : VMware Virtual Ethernet Adapter for VMnet8  
    GUID             : 4938e088-38b9-4246-aa9d-a90697f0723e  
    物理地址         : 00-50-56-C0-00-08  
    状态             : 已连接。网络不支持身份验证。

    名称             : 以太网 3  
    描述             : TAP-Windows Adapter V9  
    GUID             : 36021622-e28d-4f36-9c81-6785999498fa  
    物理地址         : 00-FF-36-02-16-22  
    状态             : 网络电缆被拔出

    名称             : 以太网  
    描述             : Realtek PCIe GbE Family Controller  
    GUID             : 65efcfa3-9c76-4d10-9bf1-9a1dc9d3b7ed  
    物理地址         : 80-E8-2C-31-0E-4C  
    状态             : 网络电缆被拔出

**开启无线接口**

netsh interface set interface "Interface Name" enabled



# 扩展1：

## 1>使用netsh命令备份网络配置

### netsh dump > e:/netsetting\_bak.txt

## 2>使用netsh命令导入网络配置

### netsh –f e:/netsetting\_bak.txt

# 扩展2：

## ipconfig的一些特殊用法：

|  |
| --- |
|  |

# 扩展3：

## tracert命令的用法

|  |
| --- |
|  |

### 应用举例：tracert 192.168.100.48

|  |
| --- |
|  |

# 扩展4：

## 如何使用netsh advfirewall firewall而不是netsh firewall控制Windows Server 2008的防火墙行为

### 如何使用"netsh advfirewall firewall"上下文，而不是"netsh firewall"上下文控制Windows Vista 和 Windows Server 2008 中的防火墙行为

适用于： Windows Server 2008 EnterpriseWindows Server 2008 DatacenterWindows Server 2008 Standard [详细](https://support.microsoft.com/)

重要提示：本文章是 Microsoft 软件自动翻译的结果，而非专业译者翻译的结果。 Microsoft 提供专业人员翻译的文章、由自动翻译生成的文章以及来自 Microsoft 社区的文章翻译，因此你能够以自己的语言阅读所有知识库文章。 需要注意的是，由自动翻译生成的文章（包括 Microsoft 社区审阅的文章）可能包含词汇、句法或语法错误。 对于因不正确的内容翻译或使用不正确的内容翻译而造成的不准确或错误或任何损害，Microsoft 概不负责。

查看原始英文文章：[947709](https://support.microsoft.com/en-us/help/947709/how-to-use-the-netsh-advfirewall-firewall-context-instead-of-the-netsh)

**简介**

Windows Vista 和 Windows Server 2008 中，由防火墙的 netsh命令行上下文可用。此上下文提供了用于控制 Windows 防火墙行为,是由netsh 防火墙上下文在早期版本的 Windows 操作系统中提供的功能。这里还提供了更精确地控制的防火墙规则的功能。这些规则包括以下的每个配置文件设置：

* 域
* 私有
* 公共

防火墙 netsh命令行上下文可能会在未来版本的 Windows 操作系统中不建议使用。我们建议您使用netsh 由防火墙上下文以控制防火墙行为。注意：不建议在 Windows Vista 中使用 netsh 防火墙命令行。本文介绍如何使用netsh 由防火墙上下文，而不是控制 Windows Vista 和 Windows Server 2008 中的 Windows 防火墙行为的netsh 防火墙上下文。

**更多信息**

重要：如果您是成员的管理员组，并且您从具有提升权限的命令提示符运行命令的计算机上启用用户帐户控制。若要使用提升的权限启动命令提示符下，查找图标或开始菜单条目用来启动命令提示符会话，用鼠标右键单击它，然后单击**以管理员身份运行**。下表提供了常用命令的一些示例。可以使用这些示例来帮助您从较早的netsh 防火墙上下文迁移到新的netsh 由防火墙上下文。此外，提供了可用于获取详细的内联帮助由命令。

**示例 1： 启用程序**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall add allowedprogram C:\MyApp\MyApp.exe "My Application" ENABLE | netsh advfirewall firewall add rule name="My Application" dir=in action=allow program="C:\MyApp\MyApp.exe" enable=yes |
| netsh firewall add allowedprogram program=C:\MyApp\MyApp.exe name="My Application" mode=ENABLE scope=CUSTOM addresses=157.60.0.1,172.16.0.0/16,LocalSubnet profile=Domain | netsh advfirewall firewall add rule name="My Application" dir=in action=allow program="C:\MyApp\MyApp.exe" enable=yes remoteip=157.60.0.1,172.16.0.0/16,LocalSubnet profile=domain |
| netsh firewall add allowedprogram program=C:\MyApp\MyApp.exe name="My Application" mode=ENABLE scope=CUSTOM addresses=157.60.0.1,172.16.0.0/16,LocalSubnet profile=ALL | Run the following commands:  netsh advfirewall firewall add rule name="My Application" dir=in action=allow program="C:\MyApp\MyApp.exe" enable=yes remoteip=157.60.0.1,172.16.0.0/16,LocalSubnet profile=domain  netsh advfirewall firewall add rule name="My Application" dir=in action=allow program="C:\MyApp\MyApp.exe" enable=yes remoteip=157.60.0.1,172.16.0.0/16,LocalSubnet profile=private |

有关如何添加防火墙规则的详细信息，运行以下命令：

netsh advfirewall firewall add rule ?

**示例 2： 启用端口**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall add portopening TCP 80 "Open Port 80" | netsh advfirewall firewall add rule name="Open Port 80" dir=in action=allow protocol=TCP localport=80 |

有关如何添加防火墙规则的详细信息，运行以下命令：

netsh advfirewall firewall add rule ?

**示例 3： 删除启用的程序或端口**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall delete allowedprogram C:\MyApp\MyApp.exe | netsh advfirewall firewall delete rule name=**rule name** program="C:\MyApp\MyApp.exe" |
| delete portopening protocol=UDP port=500 | netsh advfirewall firewall delete rule name=**rule name** protocol=udp localport=500 |

有关如何删除防火墙规则，运行以下命令：

netsh advfirewall firewall delete rule ?

**示例 4： 配置 ICMP 设置**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall set icmpsetting 8 | netsh advfirewall firewall add rule name="ICMP Allow incoming V4 echo request" protocol=icmpv4:8,any dir=in action=allow |
| netsh firewall set icmpsetting type=ALL mode=enable | netsh advfirewall firewall add rule name="All ICMP V4" protocol=icmpv4:any,any dir=in action=allow |
| netsh firewall set icmpsetting 13 disable all | netsh advfirewall firewall add rule name="Block Type 13 ICMP V4" protocol=icmpv4:13,any dir=in action=block |

有关如何配置 ICMP 设置的详细信息，运行以下命令：

netsh advfirewall firewall add rule ?

**示例 5： 将日志记录设置**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall set logging %systemroot%\system32\LogFiles\Firewall\pfirewall.log 4096 ENABLE ENABLE | Run the following commands:  netsh advfirewall set currentprofile logging filename %systemroot%\system32\LogFiles\Firewall\pfirewall.log  netsh advfirewall set currentprofile logging maxfilesize 4096 netsh advfirewall set currentprofile logging droppedconnections enable  netsh advfirewall set currentprofile logging allowedconnections enable |

有关更多信息，请运行以下命令：

netsh advfirewall set currentprofile ?

如果要将特定配置文件的日志记录设置，请使用以下选项之一，而不是"currentprofile"选项：

* Domainprofile
* Privateprofile
* Publicprofile

**示例 6： 启用 Windows 防火墙**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall set opmode ENABLE | netsh advfirewall set currentprofile state on |
| netsh firewall set opmode mode=ENABLE exceptions=enable | Run the following commands:  Netsh advfirewall set currentprofile state on   netsh advfirewall set currentprofile firewallpolicy blockinboundalways,allowoutbound |
| netsh firewall set opmode mode=enable exceptions=disable profile=domain | Run the following commands:  Netsh advfirewall set domainprofile state on   netsh advfirewall set domainprofile firewallpolicy blockinbound,allowoutbound |
| netsh firewall set opmode mode=enable profile=ALL | Run the following commands:  netsh advfirewall set domainprofile state on   netsh advfirewall set privateprofile state on |

有关更多信息，请运行以下命令：

netsh advfirewall set currentprofile ?

如果要设置特定的配置文件的防火墙状态，请使用以下选项之一，而不是"currentprofile"选项：

* Domainprofile
* Privateprofile
* Publicprofile

**示例 7： 还原默认策略设置**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall reset | netsh advfirewall reset |

有关更多信息，请运行以下命令：

netsh advfirewall reset ?

**例如 8： 启用特定服务**

|  |  |
| --- | --- |
| **旧命令** | **新命令** |
| netsh firewall set service FileAndPrint | netsh advfirewall firewall set rule group="File and Printer Sharing" new enable=Yes |
| netsh firewall set service RemoteDesktop enable | netsh advfirewall firewall set rule group="remote desktop" new enable=Yes |
| netsh firewall set service RemoteDesktop enable profile=ALL | Run the following commands:  netsh advfirewall firewall set rule group="remote desktop" new enable=Yes profile=domain  netsh advfirewall firewall set rule group="remote desktop" new enable=Yes profile=private |