**Control services with SC**

***Syntax:***

Typing SC /? on the command line shows us the following help text:

DESCRIPTION:

SC is a command line program used for communicating with the

NT Service Controller and services.

USAGE:

sc <server> [command] [service name] <option1> <option2>...

The option <server> has the form "\\ServerName"

Further help on commands can be obtained by typing: "sc [command]"

Commands:

query-----------Queries the status for a service, or

enumerates the status for types of services.

queryex---------Queries the extended status for a service, or

enumerates the status for types of services.

start-----------Starts a service.

pause-----------Sends a PAUSE control request to a service.

interrogate-----Sends an INTERROGATE control request to a service.

continue--------Sends a CONTINUE control request to a service.

stop------------Sends a STOP request to a service.

config----------Changes the configuration of a service (persistant).

description-----Changes the description of a service.

failure---------Changes the actions taken by a service upon failure.

sidtype---------Changes the service SID type of a service.

qc--------------Queries the configuration information for a service.

qdescription----Queries the description for a service.

qfailure--------Queries the actions taken by a service upon failure.

qsidtype--------Queries the service SID type of a service.

delete----------Deletes a service (from the registry).

create----------Creates a service. (adds it to the registry).

control---------Sends a control to a service.

sdshow----------Displays a service's security descriptor.

sdset-----------Sets a service's security descriptor.

showsid---------Displays the service SID string corresponding to an arbitrary name.

GetDisplayName--Gets the DisplayName for a service.

GetKeyName------Gets the ServiceKeyName for a service.

EnumDepend------Enumerates Service Dependencies.

The following commands don't require a service name:

sc <server> <command> <option>

boot------------(ok | bad) Indicates whether the last boot should

be saved as the last-known-good boot configuration

Lock------------Locks the Service Database

QueryLock-------Queries the LockStatus for the SCManager Database

EXAMPLE:

sc start MyService

Would you like to see help for the QUERY and QUERYEX commands? [ y | n ]:

Press Y if you want to display help for SC's QUERY and QUERYEX commands:

QUERY and QUERYEX OPTIONS :

If the query command is followed by a service name, the status

for that service is returned. Further options do not apply in

this case. If the query command is followed by nothing or one of

the options listed below, the services are enumerated.

type= Type of services to enumerate (driver, service, all)

(default = service)

state= State of services to enumerate (inactive, all)

(default = active)

bufsize= The size (in bytes) of the enumeration buffer

(default = 4096)

ri= The resume index number at which to begin the enumeration

(default = 0)

group= Service group to enumerate

(default = all groups)

SYNTAX EXAMPLES

sc query - Enumerates status for active services & drivers

sc query messenger - Displays status for the messenger service

sc queryex messenger - Displays extended status for the messenger service

sc query type= driver - Enumerates only active drivers

sc query type= service - Enumerates only Win32 services

sc query state= all - Enumerates all services & drivers

sc query bufsize= 50 - Enumerates with a 50 byte buffer.

sc query ri= 14 - Enumerates with resume index = 14

sc queryex group= "" - Enumerates active services not in a group

sc query type= service type= interact - Enumerates all interactive services

sc query type= driver group= NDIS - Enumerates all NDIS drivers

***Common tasks***

|  |  |  |
| --- | --- | --- |
| **Notes:** | **(1)** | *service* should always be the service's short name, not the display name. Doubleclick a service in the Services Control Panel applet (or services.msc) to find its short name, or use SC \\*computer* Query to list all services with their short name. |
|  | **(2)** | The spaces following equal signs are mandatory; if a space is removed the command will fail. |

**Start a service**

SC \\*computer* Start *service*

**Stop a service**

SC \\*computer* Stop *service*

**Restart a service**

SC \\*computer* Stop *service*

SC \\*computer* Start *service*

**Make a service start automatically**

SC \\*computer* Config *service* start= auto

**Disable a service**

SC \\*computer* Config *service* start= disabled

**Change a service's display name**

SC \\*computer* Config *service* DisplayName= "*New display name*"

**Change a service's description**

SC \\*computer* Description *service* "*This is the service's new description*"

**Change a service's recovery settings**

SC \\*computer* Failure *service* actions= restart/*60000*/restart/*60000*// reset= *120*

This command will make a service restart after 1 minute (60000 milliseconds) on the first 2 failures, and take no action at further failures (to prevent infinite failing restarts, which might consume 100% CPU capacity); the failures counter will be reset after 2 minutes (120 seconds).

|  |  |
| --- | --- |
| **Note:** | Slowly hover your mouse pointer over the options in the command line above to display tool tips for each individual value. |

**Ignore a service's failure to start at boot time**

SC \\*computer* Config *service* error= ignore