MAKEMSI quickly and reliably creates MSI files in a non-programmatic way

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| [Have your say! Join the MAKEMSI discussion list or view archive! Suggest improvements. No question too simple or too complex.](http://makemsi-manual.dennisbareis.com/the_makemsi_group.htm)Top of FormBottom of Form |

|  |  |  |  |
| --- | --- | --- | --- |
| [Bottom] | [Contents] | [Prev]: MSIVAL2.EXE | [Next]: OSQL.EXE |

 | Top of FormBottom of Form[Have your say! Join the MAKEMSI discussion list or view archive! Suggest improvements. No question too simple or too complex.](http://makemsi-manual.dennisbareis.com/the_makemsi_group.htm) |

\[Tips and Tricks](http://makemsi-manual.dennisbareis.com/tips.htm)[Tools](http://makemsi-manual.dennisbareis.com/tools.htm)[MSI (Windows Installer) Platform SDK](http://makemsi-manual.dennisbareis.com/windows_installer_sdk.htm)*SC.EXE*

|  |
| --- |
| **SC.EXE** |

This is a free tool from Microsoft (comes with the version 2 "[Windows Installer SDK](http://makemsi-manual.dennisbareis.com/windows_installer_sdk.htm)").

It can be used to add service information that can't be set up in the MSI "ServiceInstall" table (such as recovery and security information).

If required it could be packaged with your product and called via a custom action script (perhaps generated with "[VbsCa](http://makemsi-manual.dennisbareis.com/vbsca.htm%22%20%5Co%20%22Defines%20the%20start%20of%20VBSCRIPT%20being%20added%20to%20the%20%5C%22Binary%5C%22%20table.)").

Note that it was written by an idiot so when it fails due to a syntax error it really won't say why and you can't trust the return code.

See the "[Testing Under the Local System Account](http://makemsi-manual.dennisbareis.com/testing_under_the_local_system_account.htm)" section for how "SC.EXE" can be used to open a command prompt running under the local system account.

|  |
| --- |
| **SC.EXE -?** |

A bit of testing shows that "\"" should be used to escape double quotes where required.

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.

 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.

 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.

 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

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 = 1024)

 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

|  |
| --- |
| **SC.EXE failure** |

DESCRIPTION:

 Changes the actions upon failure

USAGE:

 sc failure [service name] <option1> <option2>...

OPTIONS:

 reset= <Length of period of no failures (in seconds)

 after which to reset the failure count to 0 (may be INFINITE)>

 (Must be used in conjunction with actions= )

 reboot= <Message broadcast before rebooting on failure>

 command= <Command line to be run on failure>

 actions= <Failure actions and their delay time (in milliseconds),

 separated by / (forward slash) -- e.g., run/5000/reboot/800

 Valid actions are <run|restart|reboot> >

 (Must be used in conjunction with the reset= option)

|  |
| --- |
| **SC.EXE config** |

Modifies a service entry in the registry and Service Database.

SYNTAX:

sc config [service name] <option1> <option2>...

CONFIG OPTIONS:

NOTE: The option name includes the equal sign.

 type= <own|share|interact|kernel|filesys|rec|adapt|error>

 start= <boot|system|auto|demand|disabled|error>

 error= <normal|severe|critical|error|ignore>

 binPath= <BinaryPathName>

 group= <LoadOrderGroup>

 tag= <yes|no>

 depend= <Dependencies(separated by / (forward slash))>

 obj= <AccountName|ObjectName>

 DisplayName= <display name>

 password= <password>

|  |
| --- |
| **SC.EXE create** |

Creates a service entry in the registry and Service Database.

SYNTAX:

sc create [service name] [binPath= ] <option1> <option2>...

CREATE OPTIONS:

NOTE: The option name includes the equal sign.

 type= <own|share|interact|kernel|filesys|rec|error>

 (default = own)

 start= <boot|system|auto|demand|disabled|error>

 (default = demand)

 error= <normal|severe|critical|error|ignore>

 (default = normal)

 binPath= <BinaryPathName>

 group= <LoadOrderGroup>

 tag= <yes|no>

 depend= <Dependencies(separated by / (forward slash))>

 obj= <AccountName|ObjectName>

 (default = LocalSystem)

 DisplayName= <display name>

 password= <password>

|  |
| --- |
| **SC.EXE sdshow** |

DESCRIPTION:

 Displays a service's security descriptor in SDDL format

USAGE:

 sc <server> sdshow <service name>

|  |
| --- |
| **SC.EXE sdset** |

DESCRIPTION:

 Sets a service's security descriptor

USAGE:

 sc <server> sdset <service name> <SD in SDDL format>

|  |
| --- |
| **EXAMPLE** |

And yes there really are spaces after the equal sign (amazing)... Didn't it take me ages to work out how to do this!

sc.exe failure MyService reset= 240 actions= restart/5000/restart/60000/restart/600000

|  |
| --- |
| **EXAMPLE - Installation of Service** |

;----------------------------------------------------------------------------

;--- Some service constants -------------------------------------------------

;----------------------------------------------------------------------------

[#define](http://dennisbareis.com/ppwizard/hashdefine.htm) MyServiceAlias MyService

[#define](http://dennisbareis.com/ppwizard/hashdefine.htm) MyServiceDesc My Test Service (<$MSI\_MSIBASENAME>)

[#define](http://dennisbareis.com/ppwizard/hashdefine.htm) RecoveryResetInSeconds <?=6\*60\*60> ;;6 hours

[#(](http://dennisbareis.com/ppwizard/hash1line.htm) '/'

 [#define](http://dennisbareis.com/ppwizard/hashdefine.htm) RecoveryActions

 restart/5000 ;;1. Restart after 5 seconds

 restart/60000 ;;2. Restart after 1 minute

 restart/600000 ;;3+. Restart after 10 minutes

[#)](http://dennisbareis.com/ppwizard/hash1lineend.htm)

;----------------------------------------------------------------------------

;--- Add service ------------------------------------------------------------

;----------------------------------------------------------------------------

<$[DirectoryTree](http://makemsi-manual.dennisbareis.com/directorytree.htm%22%20%5Co%20%22Defines%20a%20directory%20TREE%20%28in%20MSI%20%5C%22Directory%5C%22%20table%29.%20Optionally%20makes%20a%20%5C%22CreateFolder%5C%22%20record.) Key="INSTALLDIR" Dir="[ProgramFilesFolder]\MyServiceDirectory" CHANGE="\">

<$[Component](http://makemsi-manual.dennisbareis.com/component.htm) "<$MyServiceAlias>" Create="Y" Directory\_="INSTALLDIR">

 ;--- The service EXE MUST be the keypath of the component ----------------

 <$[Files](http://makemsi-manual.dennisbareis.com/files.htm) ".\MyService.EXE" KeyFile="\*">

 ;--- Create the service (service EXE is "KeyPath" of component) ----------

 [#(](http://dennisbareis.com/ppwizard/hash1line.htm)

 <$[ServiceInstall](http://makemsi-manual.dennisbareis.com/serviceinstall.htm%22%20%5Co%20%22Creates%20a%20system%20service.)

 Name="<$MyServiceAlias>"

 DisplayName="<$MyServiceDesc>"

 Description="<$MyServiceDesc>"

 Process="Own Interactive"

 >

 [#)](http://dennisbareis.com/ppwizard/hash1lineend.htm)

 ;--- Control the service -------------------------------------------------

 <$[ServiceControl](http://makemsi-manual.dennisbareis.com/servicecontrol.htm%22%20%5Co%20%22Stops%2C%20starts%20or%20deletes%20services%20during%20install%20or%20uninstall.) Name="<$MyServiceAlias>" AtInstall="start stop" AtUninstall="stop delete">

<$[/Component](http://makemsi-manual.dennisbareis.com/component_end.htm)>

;----------------------------------------------------------------------------

;--- Need sc.exe available at install time (the CA below uses it) -----------

;----------------------------------------------------------------------------

<$[Component](http://makemsi-manual.dennisbareis.com/component.htm) "ServiceTool" Create="Y" Directory\_="INSTALLDIR">

 <$[Files](http://makemsi-manual.dennisbareis.com/files.htm) ".\sc.exe" Comment=^This tool is used to set the services recovery options.^>

<$[/Component](http://makemsi-manual.dennisbareis.com/component_end.htm)>

;----------------------------------------------------------------------------

;--- Create script to execute "sc.exe" to set recovery options --------------

;----------------------------------------------------------------------------

[#data](http://makemsi-manual.dennisbareis.com/data_for_commands.htm) "CaData"

 "INSTALLDIR" '[INSTALLDIR]'

[#data](http://makemsi-manual.dennisbareis.com/data_for_commands.htm)

<$[VbsCa](http://makemsi-manual.dennisbareis.com/vbsca.htm%22%20%5Co%20%22Defines%20the%20start%20of%20VBSCRIPT%20being%20added%20to%20the%20%5C%22Binary%5C%22%20table.) Binary="SetServiceRecovery.vbs" DATA=^CaData^>

 ;--- INSTALL -------------------------------------------------------------

 <$[VbsCaEntry](http://makemsi-manual.dennisbareis.com/vbscaentry.htm%22%20%5Co%20%22Defines%20the%20start%20of%20a%20VBSCRIPT%20custom%20action%20function%20%28entry%20point%29.) "JustBeforeStartServices">

 ;--- Call generic set recovery routine ------------------------------

 SetRecovery "<$MyServiceAlias>", <$RecoveryResetInSeconds>, "<$RecoveryActions>"

 <$[/VbsCaEntry](http://makemsi-manual.dennisbareis.com/vbscaentry_end.htm)>

 <?NewLine><?NewLine>

 '=========================================

 sub SetRecovery(ByVal ServiceAlias, ByVal RecoveryResetInSeconds, ByVal RecoveryActions)

 '=========================================

 ;--- Initialization --------------------------------------------------

 [CaDebug](http://makemsi-manual.dennisbareis.com/cadebug.htm) 1, "Set service recovery for """ & ServiceAlias & """"

 VbsCaLogInc 1

 dim ScExe : ScExe = [VbsCaCadGet](http://makemsi-manual.dennisbareis.com/vbscacadget.htm)("INSTALLDIR") & "sc.exe"

 dim ScArgs : ScArgs = "failure """ & ServiceAlias & """ reset= " & RecoveryResetInSeconds & " actions= " & RecoveryActions

 dim ScArgsQ : ScArgsQ = "qfailure """ & ServiceAlias & """"

 ;--- Set the recovery information ------------------------------------

 [CaDebug](http://makemsi-manual.dennisbareis.com/cadebug.htm) 2, "Setting up the services recovery options"

 [VbsCaRunSync](http://makemsi-manual.dennisbareis.com/vbscarunsync.htm) """" & ScExe & """ " & ScArgs, 0, "" ;;Ignore RC (can't be trusted)

 ;--- Log the current state of the recovery options -------------------

 [CaDebug](http://makemsi-manual.dennisbareis.com/cadebug.htm) 2, "Logging the services recovery options"

 [VbsCaRunSync](http://makemsi-manual.dennisbareis.com/vbscarunsync.htm) """" & ScExe & """ " & ScArgsQ, 0, ""

 ;--- Finished this service -------------------------------------------

 VbsCaLogInc -1

 end sub

<$[/VbsCa](http://makemsi-manual.dennisbareis.com/vbsca_end.htm)>

;----------------------------------------------------------------------------

;--- Call the above script during install and uninstall (schedule them) -----

;----------------------------------------------------------------------------

<$[VbsCaSetup](http://makemsi-manual.dennisbareis.com/vbscasetup.htm%22%20%5Co%20%22Sets%20up%20a%20VBSCRIPT%20based%20custom%20action.) Binary="SetServiceRecovery.vbs" Entry="JustBeforeStartServices" Seq="InstallServices-" CONDITION=^<$CONDITION\_INSTALL\_ONLY>^ DATA=^CaData^>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Microsoft awarded me an MVP (Most Valuable Professional award) in 2004, 2005, 2006, 2007, 2008 & 2009 for the Windows SDK (Windows Installer) area.http://makemsi-manual.dennisbareis.com/green-tick-9x9.gif | ***Please*** ***email me*** ***any feedback, additional information or corrections.***[See this page online (look for updates)](http://makemsi-manual.dennisbareis.com/sc_exe.htm)

|  |  |  |  |
| --- | --- | --- | --- |
| [Top] | [Contents] | [Prev]: MSIVAL2.EXE | [Next]: OSQL.EXE |

http://makemsi-manual.dennisbareis.com/madewppw.jpg***MAKEMSI© is (C)opyright Dennis Bareis 2003-2008 (All rights reserved).***Wednesday July 25 2012 at 8:13pm[Visit MAKEMSI's Home Page](http://makemsi.dennisbareis.com/)http://makemsi-manual.dennisbareis.com/red-cross-10x11.gif |  | Microsoft awarded me an MVP (Most Valuable Professional award) in 2004, 2005, 2006, 2007, 2008 & 2009 for the Windows SDK (Windows Installer) area.http://makemsi-manual.dennisbareis.com/green-tick-9x9.gif |