

Competitive Uninstaller

This document contains the following sections:

- Overview
- Supported operating systems
- Using the Competitive Uninstaller
- Working with script files
- Script command reference

Overview

The Competitive Uninstaller (AV32.exe) is a command-line utility that uses script files to completely remove antivirus programs and components that are not related to Symantec antivirus protection. Since legacy products can cause conflicts or "false positives" when performing virus scans, it is suggested that these unnecessary programs and their components be completely removed. This tool is provided with this release as an unsupported utility.

The Competitive Uninstaller includes a set of predefined scripts that are configured to remove specific antivirus programs and their components and update the appropriate registry settings. These script files are stored in the same directory as the utility. In addition, you can create your own script files that can be used by the Uninstaller to remove unwanted components.

Supported operating systems

Operating Systems supported: Windows 95/98/ME and Windows NT/2000/XP.

Using the Competitive Uninstaller

When the Competitive Uninstaller is invoked, the directory in which the AV32.exe program resides is automatically checked for script files with the .aut extension. These files typically contain instructions for invoking the uninstall program associated with the software being removed, deleting various antivirus components, and resetting or removing registry entries. Since each script is platform and software specific, each script file is compared with the registry to determine whether it matches the platform and any existing antivirus programs that are installed. When a match is found, the removal instructions contained in the script file are then executed.

The Competitive Uninstaller also includes the option to specify the execution of a single script file, eliminating the scanning of all .aut files in the Competitive Uninstaller directory.

Removing multiple antivirus programs

If more than one legacy antivirus program is found, the Competitive Uninstaller will continue to execute each applicable uninstall script. However, on Windows NT/2000/XP, platforms, if the computer reboots during the uninstall process (for example, during MSI program removal), the Competitive Uninstaller must be restarted manually.

Note: The Competitive Uninstaller supports the use of only one filename parameter. If you specify more than one script to be executed, only the first script entered as a parameter will be invoked by the utility.

Running the Competitive Uninstaller utility

By default, the Competitive Uninstaller scans the directory in which it resides for script files with the .aut extension and executes all applicable scripts sequentially.

To run the Competitive Uninstaller utility

- 1 Open a command line window.
- 2 navigate to the directory in which the Competitive Uninstaller (AV32.exe) resides.

- 3 At the command line prompt, type the following:

```
AV32.exe
```

Specifying a single script file to be executed

You can direct the Competitive Uninstaller to use a single script file by specifying the filename as a command line parameter. Specifying a file to be executed eliminates the scan of the directory in which the Competitive Uninstaller resides for additional script files.

At the command line you can specify the location of the script file to be executed, so it is not necessary to place the file in the same directory as the Competitive Uninstaller utility.

To specify a single script file to be executed

- 1 Open a command line window.
- 2 Navigate to the directory in which the Competitive Uninstaller (AV32.exe) resides.
- 3 At the command line prompt, type the following, where “filename” includes the path (if the file is not in the same directory as the utility) and filename of the script file to be executed:

```
AV32.exe filename.aut
```

Silent Uninstalls

The Competitive Uninstaller typically runs in silent mode, requiring no interaction with the user. Silent mode settings are determined by the specifications of each uninstall program invoked by the tool. If the uninstall program requires parameters, use the **AddProgramParameters** command to include the values. For more information on the **AddProgramParameters** command, see the Script command reference.

When the Competitive Uninstaller removes legacy antivirus components from Windows 2000 or XP systems, there may be additional confirmation windows that require the user to decide whether or not to remove components such as .dlls or drivers.

For Windows NT/2000/XP platforms, if the computer reboots during the uninstall process (for example, during MSI program removal), the Competitive Uninstaller must be restarted manually.

Creating a log file

The Competitive Uninstaller creates a log file (status.ini) that describes whether the removal of legacy antivirus components succeeded or failed. The log file is stored in the directory in which the Competitive Uninstaller resides.

Each time a script is executed, the log file is overwritten. So, if you are removing multiple legacy programs, only the most recent removal attempt is recorded in the log file. In addition, only one error is reported each time a script is executed.

The log file format is as follows:

```
[RESULT]
RETURN=1 (success) or 0 (failure)
STATUS= (message)
```

The status message provides additional details primarily if the removal has failed.

Working with script files

Creating a new script file

Script files are text files that are saved with the .aut extension. Using a text editor such as Notepad, you can create a custom script file to invoke an antivirus uninstall program, remove or edit registry entries, and remove legacy antivirus components.

Script files have the following guidelines:

- Platform designation: Each script created must include the Platform command as the first command in the file, specifying the platform on which the script is to be run. For more information on valid parameter values, see the Script command reference section.
- File extension: Scripts are saved using the .aut extension.
- Code comments: Use the semicolon (;). The comment character can be used only at the beginning of a line.
- Do not include extra spaces or empty lines in the script file.

The standard set of script files are included with the Competitive Uninstaller in the directory in which the program resides. For new script files, you can either save them in the same directory or specify the path when you use the tool to invoke your script individually.

Uninstall programs that require a reboot during uninstallation

Some uninstall programs that are invoked by the Competitive Uninstaller require the computer to be rebooted and the script to be restarted in order to complete the uninstall process. If that is the case with the program that you are uninstalling, note that your script does not need to include the **GetProgramName** or **ClearRegistry** commands to be run after the reboot. The values that were gathered during the first pass are automatically stored by the Competitive Uninstaller and retrieved during the second pass. To complete the second run, the only requirement is to add the **RunRobot** command after the **RestartComputer** or **PauseForNotificationOfCompletion** commands.

Script command reference

This section contains reference information for each script option recognized by the Antivirus Uninstaller. Script entries are arranged in alphabetical order.

AddProgramParameters

This command sets a member variable and is used to pass parameters to the antivirus uninstall program. Only one parameter is supported, therefore all parameters should be passed in with only one call of this command.

RunRobot uses the result of this command.

Syntax

```
addprogramparameters=parameter
```

Example

```
addprogramparameters=/s /i
```

Parameters

parameter

parameters used by the uninstall program that is being invoked.

Remarks

Do not separate multiple parameters with any delimiters. All parameters should be entered as they are to be used in the command line.

ClearRegistry

Removes a registry key or key value.

Warning: This feature automatically removes registry entries without prompting for confirmation of these changes. There is no validation of the *hKey*, *SubKey* or *Key* parameters. All keys in the tree below the specified key are also removed.

Syntax

```
clearregistry=hKey, SubKey, Key
```

Example

```
clearregistry=HKEY_CURRENT_USER, SOFTWARE\Microsoft\Windows  
NT\CurrentVersion\Windows, run
```

Parameters

hKey

The *hKey* that is passed into the *RegOpenKey*.

SubKey

The *SubKey* that is passed into the *RegOpenKey*.

Key

The *Key* that is passed into *RegDeleteValue* / *RegDeleteKey*.

GetUninstallProgramName

Reads the registry and sets a member variable to the specified key value. This command reads the registry key that contains the full path and filename for the antivirus uninstall program that is being invoked.

Syntax

```
getuninstallprogramname=hKey, SubKey, Key
```

Example

```
getuninstallprogramname=HKEY_LOCAL_MACHINE,  
SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IBM AntiVirus,  
UninstallString
```

Parameters

hKey

The hKey that is passed into the RegOpenKey.

SubKey

The SubKey that is passed into the RegOpenKey.

Key

The Key that is passed into RegQueryValueEx.

GetUninstallProgramPath

Reads the registry and sets a member variable to the specified key value. This command reads the registry key that contains the path for the antivirus uninstall program that is being invoked. This command is called prior to calling **SetUninstallProgramName** and **RunRobot**.

Syntax

```
getuninstallprogrampath=hKey, SubKey, Key
```

Example

```
getuninstallprogrampath=HKEY_LOCAL_MACHINE,  
SOFTWARE\McAfee\Scan95,Data
```

Parameters

hKey

The hKey that is passed into the RegOpenKey.

SubKey

The SubKey that is passed into the RegOpenKey.

Key

The Key that is passed into RegQueryValueEx.

GetUninstallScriptPath

Reads the registry and sets a member variable to the specified key value. This command reads a registry key that contains the path for the antivirus uninstall

script file. This command is called prior to calling **SetUninstallScriptName** or **RunRobot**.

Syntax

```
getuninstallscriptpath=hKey, SubKey, Key
```

Example

```
getuninstallscriptpath=HKEY_LOCAL_MACHINE,  
SOFTWARE\McAfee\Scan95,Dat
```

Parameters

hKey

The hKey that is passed into the RegOpenKey.

SubKey

The SubKey that is passed into the RegOpenKey.

Key

The Key that is passed into RegQueryValueEx.

GetUninstallShieldProgramName

Reads the registry and sets a member variable to the specified key value. This command reads the registry key that contains the full path and filename for the antivirus uninstall program. This command is only to be used if the antivirus uninstall program internally calls the InstallShield uninstall program (i.e.UINST.EXE). This command opens the antivirus uninstall log script file and place the "-y -a" run silent parameters in the command line for the InstallShield uninstall program. This is currently used for McAfee.

Syntax

```
getuninstallshieldprogramname=hKey, SubKey, Key
```

Example

```
getuninstallshieldprogramname=HKEY_LOCAL_MACHINE,  
SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IBM AntiVirus,  
UninstallString
```

Parameters

hKey

The hKey that is passed into the RegOpenKey.

SubKey

The SubKey that is passed into the RegOpenKey.

Key

The Key that is passed into RegQueryValueEx.

PauseForNotificationOfCompletion

This command is used to prepare for a reboot and second pass on computers that cannot be automatically rebooted (Windows NT servers, for example). Information needed on the second pass and the Run registry settings are written, then a message box is activated to let the user know that the computer needs to be manually restarted.

Syntax

```
pausefornotificationofcompletion
```

Example

```
pausefornotificationofcompletion
```

Parameters

N/A

Remarks

This command is only supported for Windows NT.

Platform

This command is used to signify the script file platform. This should be the first command in the script file. This command is required for all script files.

Syntax

```
platform=platformname
```

Example

```
platform=Win95
```

Parameters

platformname

Valid values are : Win95, Win98, WinNT

Remarks

For Windows 2000, ME, and XP, use the WinNT value.

RemoveFromAutoexec

This command searches autoexec.bat for the search string and deletes all lines that contain that string.

Syntax

```
removefromautoexec=searchstring
```

Example

```
removefromautoexec=scan.exe
```

Parameters

searchstring

Text to search for in the autoexec.bat file.

RestartComputer

Writes all necessary information to the registry for the second run, if necessary, then reboots the computer. This command is supported under Windows NT only.

Syntax

```
restartcomputer
```

Example

```
restartcomputer
```

Parameters

N/A

Remarks

This command is only supported for NT.

RunProgram

Executes the program that is read from the provided Registry keys. If there are any parameters in the registry they are used as well. This command uses the WinExec API call and does not wait for the program to complete. This command can be used when you need to run a program that **RunRobot** is unable to shutdown. This is currently used for Fprot 95.

Syntax

```
runprogram=hKey, SubKey, Key
```

Example

```
runprogram=HKEY_LOCAL_MACHINE,  
SOFTWARE\Windows\CurrentVersion\Uninstall\F-Prot 95,UninstallString
```

Parameters

hKey

The hKey that is passed into the RegOpenKey.

SubKey

The SubKey that is passed into the RegOpenKey.

Key

The Key that is passed into RegQueryValueEx.

RunRobot

Automatically runs and performs all IDOK and IDYES key presses for the antivirus uninstall program dialogs and then writes to the registry for the Run key. This command is used to restart the Competitive Uninstaller after the computer has been rebooted by the invoked antivirus uninstall program.

Syntax

```
runrobot=IdleTimeOut
```

Example

```
runrobot=300
```

Parameters

IdleTimeout [optional]

Maximum time in seconds for the Competitive Uninstaller to wait for the uninstall program to return an idle message.

If no value is supplied, the default is 5 minutes.

Remarks

This command requires that a program path and name has been previously set using one of the following sets of commands:

```
GetUninstallProgramName
```

```
GetUninstallShieldProgramName
```

or:

```
GetUninstallProgramPath
```

```
SetUninstallProgramName
```

To add parameters to the uninstall program set above, add the following command, specifying the appropriate parameters:

```
AddProgramParameters=[parameters]
```

To add a script for the uninstall program set above, add the following commands:

```
GetUninstallScriptPath
```

```
SetUninstallScriptName
```

RunRobotSendCancel

[[TECH REVIEW: THIS FUNCTION IS A DUPLICATE OF RUNROBOT. WHAT IS THE DIFFERENCE?]]

Automatically runs and performs all IDOK, IDYES and IDCANCEL key presses for the antivirus uninstall program dialogs, then writes to the registry for the Run key. This command is used to restart the Competitive Uninstaller after the computer has been rebooted by the invoked antivirus uninstall program.

Syntax

```
runrobotsendcancel=IdleTimeout
```

Example

```
runrobotsendcancel=300
```

Parameters

IdleTimeOut [optional]

Maximum time in seconds for the Competitive Uninstaller to wait for the uninstall program to return an idle message.

If no value is supplied, the default is 5 minutes.

Remarks

This command requires that a program path and name has been previously set using one of the following sets of commands:

```
GetUninstallProgramName
```

```
GetUninstallShieldProgramName
```

or:

```
GetUninstallProgramPath
```

```
SetUninstallProgramName
```

To add parameters to the uninstall program set above, add the following command, specifying the appropriate parameters:

```
AddProgramParameters=[parameters]
```

To add a script for the uninstall program set above, add the following commands:

```
GetUninstallScriptPath
```

```
SetUninstallScriptName
```

RunRobotSetRunRegistry

Automatically runs and performs all IDOK and IDYES key presses for the antivirus uninstall program dialogs, then writes to the registry for the Run key, so that the Competitive Uninstaller is restarted once the computer has been rebooted by the uninstall program that is being invoked. This command should be used when the uninstall program automatically reboots the computer and then creates a dialog box that requires a keystroke after the reboot.

Syntax

```
runrobotsetrunregistry=IdleTimeOut
```

Example

```
runrobotsetrunregistry=300
```

Parameters

IdleTimeOut [optional]

Maximum time in seconds for the Competitive Uninstaller to wait for the uninstall program to return an idle message.

If no value is supplied, the default is 5 minutes.

Remarks

This command requires that a program path and name has been previously set using one of the following sets of commands:

```
GetUninstallProgramName
```

or:

```
GetUninstallProgramPath
```

```
SetUninstallProgramName
```

To add parameters to the uninstall program set above, add the following command, specifying the appropriate parameters:

```
AddProgramParameters=[parameters]
```

To add script for the uninstall program set above, add the following commands:

```
GetUninstallScriptPath
```

```
SetUninstallScriptName
```

This command combines the features of **RunRobot** and **SetRegistryForUninstallSecondRun**.

SendAffirmativeToWindow

This command sends the IDOK and IDYES window messages to the window who's window name has been passed in. This is designed to shut down dialogs that are not controlled by **RunRobot**. This currently is used for Cheyenne.

Syntax

```
sendaffirmativetowindow=windowname, waittime
```

Example

```
sendwindowmessage=VshieldMonitor, 60
```

Parameters

windowname

The window name is the text in the header of the window. This parameter is used by FindWindowEx to locate the AntiVirus dialog to send the message to.

This parameter is case sensitive.

waittime

number of seconds to search for the window name.

SendAffirmativeToWindowAfterEnable

This command sends the IDOK and IDYES window messages to the specified window. This command will loop until the window name is found. When the window name is found, it will search for the specified button, and then enable the button. This can be used to shut down the Uninstall Shield dialog. This is designed to shut down dialogs that are not controlled by **RunRobot**. This is currently used for FProt 95.

Syntax

```
sendaffirmativetowindow=windowname, buttonname, waittime
```

Example

```
sendwindowmessage=Remove Programs From Your Computer, OK, 60
```

Parameters

windowname

The window name is the text in the header of the window. This parameter is used by FindWindowEx to locate the AntiVirus dialog to send the message to.

This parameter is case sensitive.

Buttonname

The text on the button to search and wait for the enable status before sending the message. This is case sensitive.

waittime

The maximum number of seconds to search for the window name, button, and button enable status.

SendWindowsMessage

This command sends a `windows` message to the window who's class has been passed in. This is designed to shut down AntiVirus programs that might be running so that they can be properly uninstalled.

Syntax

```
sendwindowmessage=windowname, ASCIIMessage
```

Example

```
sendwindowmessage=VshieldMonitor, 16
```

Parameters

windowname

The window name is the text in the header of the window. This parameter is used by FindWindowEx to locate the AntiVirus dialog to send the message to.

This parameter is case sensitive.

ASCIIMessage

ASCII value for the windows message to be sent

Remarks

Message 16 is the WM_CLOSE. This cannot pass the wParam or the lParam.

SendWindowsMessageAndWait

This command sends a `windows` message to the specified window. If the window is not found, it retries until the window is found or the wait time has

expired. This is designed to shut down windows that are not controlled by **RunRobot**.

Syntax

```
sendwindowmessage=windowname, ASCII message, waittime
```

Example

```
sendwindowmessage=VshieldMonitor, 16, 60
```

Parameters

windowname

The window name is the text in the header of the window. This parameter is used by FindWindowEx to locate the AntiVirus dialog to send the message to.

This parameter is case sensitive.

message

ASCII value for the windows message to be sent

waittime

number of seconds to attempt to find the window

Remarks

Message 16 is the WM_CLOSE. This cannot pass the wParam or the lParam.

SetRegistry

Set the specified registry key value.

Syntax

```
setregistry=hKey, SubKey, Key, Value
```

Example

```
setregistry=HKEY_CURRENT_USER, SOFTWARE\Microsoft\Windows  
NT\CurrentVersion\Windows, run, E:\S32UI\debug\s32ui.exe
```

Parameters

hKey

the HKey that is passed into RegCreateKey/RegOpenKey.

SubKey

The SubKey that is passed into RegCreateKey/RegOpenKey.

Key

The Key that is passed into RegSetValueEx.

Value

the Key value that is passed into RegSetValueEx.

SetRegistryForUninstallSecondRun

This command sets the Run registry setting so that after the **RebootSystem** command is performed the Competitive Uninstaller will automatically restart.

Syntax

```
setregistryforuninstallsecondrun
```

Example

```
setregistryforuninstallsecondrun
```

Parameters

N/A

SetUninstallProgramName

Combines the member variable set by **GetUninstallProgramPath**, and the filename that is passed in to this command. **RunRobot** uses the result of this command.

Syntax

```
setuninstallprogramname=filename
```

Example

```
setuninstallprogramname=uninstall.exe
```

Parameters

filename

file to perform uninstall

SetUninstallScriptName

Combines the member variable set by **GetUninstallScriptPath**, and the filename that is passed in to this command. **RunRobot** uses the result of this command.

Syntax

```
setuninstallscripname=filename
```

Example

```
setuninstallscripname=install.log
```

Parameters

filename

script file used by the uninstall program

SetupForSecondRun

This command reads the registry and prepares for a reboot when a second pass is required. Information needed on second pass and the Run registry settings are written.

Syntax

```
setupforsecondrun=runkeyname
```

Example

```
setupforsecondrun=run
```

Parameters

runkeyname

For NT the runkeyname is run.

For Win95: The runkeyname is the name that is in the RunOnce key of the HKEY_LOCAL_MACHINE

Software\Microsoft\Windows\CurrentVersion registry key.

Remarks

Windows NT: This command only needs to be used if the antivirus uninstall program is different than the one that was run on the first pass.

Windows 95: For uninstall programs that require a reboot, this command should be placed just before the **RebootComputer** or **PauseForNotificationOfCompletion** commands.

StopProgramForReboot

This command is used to prepare for a reboot by the uninstaller program when a second pass is needed. Information needed for the second pass and the Run registry settings are written.

Syntax

```
stopprogramforreboot
```

Example

```
stopprogramforreboot
```

Parameters

N/A