Security Software Uninstaller

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Copyright Notice
Copyright © 2005 Symantec Corporation.
All Rights Reserved.
Any technical documentation that is made available by Symantec Corporation is the copyrighted work of Symantec Corporation and is owned by Symantec Corporation. NO WARRANTY. The technical documentation is being delivered to you AS-IS, and Symantec Corporation makes no warranty as to its accuracy or use. Any use of the technical documentation or the information contained therein is at the risk of the user. Documentation may include technical or other inaccuracies or typographical errors. Symantec reserves the right to make changes without prior notice.
No part of this publication may be copied without the express written permission of Symantec Corporation, 20330 Stevens Creek Blvd., Cupertino, CA 95014.

Trademarks
Symantec, the Symantec logo, LiveUpdate, and Norton AntiVirus are U.S. registered trademarks of Symantec Corporation. Norton Internet Security, Norton Personal Firewall, Symantec AntiVirus, Symantec Client Firewall, Symantec Client Security, and Symantec Security Response are trademarks of Symantec Corporation. Other brands and product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.
Printed in the United States of America.
10 9 8 7 6 5 4 3 2 1
Technical support

As part of Symantec Security Response, the Symantec global Technical Support group maintains support centers throughout the world. The Technical Support group’s primary role is to respond to specific questions on product feature/function, installation, and configuration, as well as to author content for our Web-accessible Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering as well as Symantec Security Response to provide Alerting Services and Virus Definition Updates for virus outbreaks and security alerts.

Symantec technical support offerings include:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and Web support components that provide rapid response and up-to-the-minute information
- Upgrade insurance that delivers automatic software upgrade protection
- Content Updates for virus definitions and security signatures that ensure the highest level of protection
- Global support from Symantec Security Response experts, which is available 24 hours a day, 7 days a week worldwide in a variety of languages for those customers enrolled in the Platinum Support Program
- Advanced features, such as the Symantec Alerting Service and Technical Account Manager role, offer enhanced response and proactive security support

Please visit our Web site for current information on Support Programs. The specific features available may vary based on the level of support purchased and the specific product that you are using.

Licensing and registration

If the product that you are implementing requires registration and/or a license key, the fastest and easiest way to register your service is to access the Symantec licensing and registration site at www.symantec.com/certificate. Alternatively, you may go to www.symantec.com/techsupp/ent/enterprise.html, select the product that you wish to register, and from the Product Home Page, select the Licensing and Registration link.

Contacting Technical Support

Customers with a current support agreement may contact the Technical Support group via phone or online at www.symantec.com/techsupp. Customers with Platinum support agreements may contact Platinum Technical Support via the Platinum Web site at www-secure.symantec.com/platinum/.
When contacting the Technical Support group, please have the following:

- Product release level
- Hardware information
- Available memory, disk space, NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description
  - Error messages/log files
  - Troubleshooting performed prior to contacting Symantec
  - Recent software configuration changes and/or network changes

Customer Service

To contact Enterprise Customer Service online, go to www.symantec.com, select the appropriate Global Site for your country, then choose Service and Support. Customer Service is available to assist with the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information on product updates and upgrades
- Information on upgrade insurance and maintenance contracts
- Information on Symantec Value License Program
- Advice on Symantec's technical support options
- Nontechnical presales questions
- Missing or defective CD-ROMs or manuals
Security Software Uninstaller

About the Security Software Uninstaller ........................................................... 7
System requirements ............................................................................................ 8
Using the Security Software Uninstaller ........................................................... 8
   Using the command line parameters ......................................................... 8
   Running the Security Software Uninstaller .............................................. 9
   About performing silent uninstallation ................................................... 10
   About log files ............................................................................................... 10
About working with scripts ................................................................................ 11
   About new script files ................................................................................. 11
   Required restarts during uninstallation .................................................. 12
Script command reference ............................................................................... 12
   AddProgramParameters ............................................................................. 12
   ClearRegistry ................................................................................................ 13
   GetUninstallProgramName ....................................................................... 14
   GetUninstallProgramPath .......................................................................... 14
   GetUninstallScriptPath .............................................................................. 15
   GetUninstallShieldProgramName ............................................................ 15
   PauseForNotificationOfCompletion ......................................................... 16
   Platform ........................................................................................................ 16
   RemoveFromAutoexec ................................................................................ 17
   RestartComputer ......................................................................................... 17
   RunProgram ................................................................................................. 18
   RunRobot ...................................................................................................... 18
   RunRobotSendCancel ................................................................................ 19
   RunRobotSetRunRegistry .......................................................................... 20
   SendAffirmativeToWindow ....................................................................... 21
   SendAffirmativeToWindowAfterEnable ...................................................... 22
   SendAffirmativeToWindowWithButtonClick ............................................. 22
   SendINIEntry ................................................................................................ 23
   SendTextToEditWindow ............................................................................. 23
   SendWindowsMessage .............................................................................. 24
   SendWindowsMessageAndWait ................................................................. 25
   SetRegistry .................................................................................................... 25
   SetRegistryForUninstallSecondRun ........................................................ 26
   SetUninstallProgramName ....................................................................... 26
About the Security Software Uninstaller

The Security Software Uninstaller (AV32.exe) is a command line utility that uses script files to completely remove antivirus and firewall programs and components that are not related to Symantec antivirus protection. Because legacy products can cause conflicts or false positives during virus scans, you should completely remove these unnecessary programs and their components.

The Security Software Uninstaller includes a set of predefined scripts that are configured to remove specific antivirus and firewall programs and their components and to update the appropriate registry settings. These scripts reside in the same directory as the Security Software Uninstaller utility. In addition, you can create your own scripts for use by the Security Software Uninstaller to remove unwanted components.
System requirements

For the Security Software Uninstaller, the system requirements are as follows:

- Windows 95/98/Me
- Windows NT/2000/XP

Using the Security Software Uninstaller

When you invoke the Security Software Uninstaller, the script that is specified as a command line parameter is executed. Script files typically contain instructions for invoking the uninstallation program that is associated with the software that is being removed. The Security Software Uninstaller removes various antivirus or firewall components, and either resets or removes registry entries. Because each script is platform-specific and software-specific, the Security Software Uninstaller compares each script with the registry to determine whether it matches the platform and any existing antivirus or firewall programs that are installed. When the Security Software Uninstaller finds a match, it executes the removal instructions in the script.

The Security Software Uninstaller also lets you specify the execution of all of the script files in the directory in which the Security Software Uninstaller program is installed.

Using the command line parameters

The Security Software Uninstaller provides a set of command line parameters that allows you to customize the settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;none&gt;</td>
<td>If no parameters are specified, the Security Software Uninstaller will display a summary of the usage details.</td>
</tr>
<tr>
<td>&lt;script&gt;</td>
<td>This specifies the script file to process.</td>
</tr>
</tbody>
</table>
Running the Security Software Uninstaller

By default, the Security Software Uninstaller runs only the script that is specified when you run the tool.

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

If there are multiple products to be removed, you can either run the product multiple times specifying each script, or you can use a command line switch to instruct the Security Software Uninstaller to scan the directory in which it resides for scripts with the .aut extension and execute all of the scripts sequentially.

To run the Security Software Uninstaller with a single script

1. Open a command line window.
2. Navigate to the directory in which the Security Software Uninstaller (AV32.exe) resides.
3. At the command prompt, type the following, where filename includes the path (if the file is not in the same directory as the utility) and the file name of the script to be executed:
   
   \[AV32.exe filename.aut\]

To run the Security Software Uninstaller using all of the scripts in the installation directory

1. Open a command line window.
2  Navigate to the directory in which the Security Software Uninstaller (AV32.exe) resides.

3  At the command prompt, type the following:

    AV32.exe /all

**Note:** If you are executing all of the available scripts automatically, if the computer restarts during the uninstallation process (for example, during MSI program removal), you must restart the Security Software Uninstaller manually.

---

**About performing silent uninstallation**

The Security Software Uninstaller typically runs in silent mode, which requires no interaction with the user. Silent mode settings are determined by the specifications of each uninstallation program that the Security Software Uninstaller invokes. If the uninstallation program requires parameters, use the `AddProgramParameters` command to include the values.

See “Script command reference” on page 12.

When the Security Software Uninstaller removes legacy antivirus or firewall components from Windows 2000/XP computers, 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, if the computer restarts during the uninstallation process (for example, during MSI program removal), you must restart the Security Software Uninstaller manually.

---

**About log files**

The Security Software Uninstaller creates a log file (Status.ini) that describes whether the removal of legacy antivirus or firewall components succeeded or failed. The log file resides in the same directory as the Security Software Uninstaller.

Each time that the Security Software Uninstaller executes a script, it overwrites the log file. If you are removing multiple legacy programs, the log file records only the most recent removal attempt. In addition, the log file reports only one error each time that the Security Software Uninstaller executes a script.

The log file format is as follows:

```
[RESULT]
RETURN=1 (success) or 0 (failure)
```
The status message provides additional details primarily if the removal has failed.

About working with scripts

The Security Software Uninstaller uses a set of scripts to remove unnecessary antivirus and firewall programs that may conflict with Symantec antivirus software. The Security Software Uninstaller includes a set of predefined scripts as well as support for customized scripts that you create.

About new script files

Scripts are text files that are saved with .aut extensions. Using a text editor such as Notepad, you can create a custom script to invoke an antivirus or firewall uninstallation program, remove or edit registry entries, and remove legacy components.

Scripts have the following guidelines:

- Platform designation: Each script must include the `Platform` command as the first command in the file, which specifies the platform on which the script is to be run. See “Script command reference” on page 12.
- File extension: Each script must be saved with the .aut extension.
- Code comments: The comment character can be used only at the beginning of a line. Use a semicolon (;).
- Extra spaces or empty lines: These cannot be included.

There are two approaches to writing script files for uninstalling applications. The first sets up a list of parameters and then calls `RunRobot`, which waits for user input and sends messages to waiting dialogs. The second uses `RunProgram`, which calls the application’s uninstall program and sends responses to dialogs through script commands. If you know exactly which dialog boxes will appear, `RunProgram` is recommended because it has less timing dependencies. Otherwise, the `RunRobot` command can be used to assist with detection and responses to dialogs.

The standard set of scripts resides in the same directory as the Security Software Uninstaller. For new script files, you can either save them in the same
directory or specify the path when you use the Security Software Uninstaller to invoke your script individually.

Required restarts during uninstallation

Some uninstallation programs that the Security Software Uninstaller invokes require the computer to be restarted and the script to be restarted in order to complete the uninstallation process. If that is the case with the program that you are uninstalling, your script does not need to include the \texttt{GetUninstallProgramName} or \texttt{ClearRegistry} command to be run after the restart. The Security Software Uninstaller automatically stores values that were gathered during the first pass and retrieves them during the second pass. To complete the second run, the only requirement is to add the \texttt{RunRobot} command after the \texttt{RestartComputer} or \texttt{PauseForNotificationOfCompletion} command.

Script command reference

The Security Software Uninstaller recognizes 24 different script options.

AddProgramParameters

The \texttt{AddProgramParameters} command sets a member variable and passes parameters to the antivirus or firewall uninstallation program. This command supports only one parameter. Therefore, all parameters should be passed with only one call of this command. \texttt{RunRobot} uses the result of this command.

\textbf{Syntax}

\begin{verbatim}
addprogramparameters=parameter
\end{verbatim}

\textbf{Example}

\begin{verbatim}
addprogramparameters=/s /i
\end{verbatim}

\textbf{Parameter}

\begin{verbatim}
parameter
\end{verbatim}

The parameters that are used by the uninstallation program that is being invoked.

\textbf{Remarks}

Do not separate multiple parameters with delimiters. You should enter all parameters as they are to be used at the command line.
ClearRegistry

The **ClearRegistry** command removes a registry key or key value.

---

**Warning:** This feature automatically removes registry entries without prompting you for confirmation of these changes. There is no validation of the hKey, SubKey, or Key parameter. This command also removes all of the keys in the tree that is below the specified key.

---

**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 RegOpenKey
**SubKey**

The SubKey that is passed into RegOpenKey

**Key**

The Key that is passed into RegDeleteValue/RegDeleteKey

---

**GetUninstallProgramName**

The **GetUninstallProgramName** command 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 file name for the antivirus or firewall uninstallation 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 RegOpenKey

- **SubKey**
  
  The SubKey that is passed into RegOpenKey

- **Key**
  
  The Key that is passed into RegQueryValueEx

---

**GetUninstallProgramPath**

The **GetUninstallProgramPath** command 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 or firewall uninstallation 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, Dat
Parameters

- **hKey**: The hKey that is passed into RegOpenKey
- **SubKey**: The SubKey that is passed into RegOpenKey
- **Key**: The Key that is passed into RegQueryValueEx

### GetUninstallScriptPath

The **GetUninstallScriptPath** command 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 or firewall uninstallation 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 RegOpenKey
- **SubKey**: The SubKey that is passed into RegOpenKey
- **Key**: The Key that is passed into RegQueryValueEx

### GetUninstallShieldProgramName

The **GetUninstallShieldProgramName** command 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 file name for the antivirus or firewall uninstallation program. Use this command only if the antivirus or firewall uninstallation program internally calls the InstallShield uninstallation program (Uninst.exe). This command opens the antivirus or firewall uninstallation log script and places the `-y -a` run silent parameters in the command line for the InstallShield uninstallation program. This command is currently used for McAfee products.
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 RegOpenKey
SubKey
The SubKey that is passed into RegOpenKey
Key
The Key that is passed into RegQueryValueEx

PauseForNotificationOfCompletion
The PauseForNotificationOfCompletion command prepares for a restart and second pass on computers that cannot be automatically restarted (Windows NT servers, for example). Information that is needed on the second pass and the Run registry settings are written to a separate registry location, and then a message box lets the user know that the computer needs to be manually restarted.

Syntax
pausefornotificationofcompletion

Example
pausefornotificationofcompletion

Parameters
N/A

Remarks
This command is supported for Windows NT only.

Platform
The Platform command signifies 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

Parameter
platformname
Valid values are as follows: Win95, Win98, and WinNT.

Remarks
For Windows 2000/Me/XP, use the WinNT value.

RemoveFromAutoexec

The **RemoveFromAutoexec** command searches Autoexec.bat for the search string and deletes any lines that contain that string.

Syntax
removefromautoexec=searchstring

Example
removefromautoexec=scan.exe

Parameters
searchstring
Text to search for in the Autoexec.bat file

RestartComputer

The **RestartComputer** command writes all necessary information to the registry for the second run, if needed, and then restarts the computer.

Syntax
restartcomputer

Example
restartcomputer

Parameters
N/A
Remarks
This command is supported for Windows NT only.

RunProgram
The RunProgram command 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 shut down. 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 RegOpenKey
- SubKey
  The SubKey that is passed into RegOpenKey
- Key
  The Key that is passed into RegQueryValueEx

RunRobot
The RunRobot command automatically runs and performs all IDOK and IDYES key presses for the antivirus or firewall uninstallation program dialog boxes and then writes to the registry for the Run key. This command is used to restart the Security Software Uninstaller after the computer has been restarted by the invoked antivirus or firewall uninstallation program.

Syntax
runrobot=IdleTimeOut

Example
runrobot=300

Parameter
- IdleTimeOut [optional]
The maximum time in seconds for the Security Software Uninstaller to wait for the uninstallation program to return an idle message. If no value is supplied, the default is five minutes.

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

GetUninstallProgramName
GetUninstallShieldProgramName
or:
GetUninstallProgramPath
SetUninstallProgramName

To add parameters to this uninstallation program set, add the following command and specify the appropriate parameters:

AddProgramParameters=[parameters]

To add a script for this uninstallation program set, add the following commands:

GetUninstallScriptPath
SetUninstallScriptName

RunRobotSendCancel

The RunRobotSendCancel command automatically runs and performs all IDOK, IDYES, and IDCANCEL key presses for the antivirus or firewall uninstallation program dialog boxes, and then writes to the registry for the Run key. This command is used to restart the Security Software Uninstaller after the computer has been restarted by the invoked antivirus or firewall uninstallation program.

Syntax

runrobotsendcancel=IdleTimeOut

Example

runrobotsendcancel=300

Parameter

IdleTimeOut [optional]

The maximum time in seconds for the Security Software Uninstaller to wait for the uninstallation program to return an idle message. If no value is supplied, the default is five minutes.
Remarks
This command requires that a program path and name be previously set using
one of the following sets of commands:
GetUninstallProgramName
GetUninstallShieldProgramName
or:
GetUninstallProgramPath
SetUninstallProgramName
To add parameters to this uninstallation program set, add the following
command and specify the appropriate parameters:
AddProgramParameters=[parameters]
To add a script for this uninstallation program set, add the following commands:
GetUninstallScriptPath
SetUninstallScriptName

RunRobotSetRunRegistry
The RunRobotSetRunRegistry command automatically runs and performs all
IDOK and IDYES key presses for the antivirus or firewall uninstallation program
dialog boxes. This command also writes to the registry for the Run key so that
the Security Software Uninstaller is restarted once the computer has been
restarted by the uninstallation program that is being invoked. This command
should be used when the uninstallation program automatically restarts the
computer and then creates a dialog box that requires a keystroke after the
restart.

Syntax
runrobotsetrunregistry=IdleTimeOut

Example
runrobotsetrunregistry=300

Parameter
IdleTimeOut [optional]
The maximum time in seconds for the Security Software Uninstaller to
wait for the uninstallation program to return an idle message. If no
value is supplied, the default is five minutes.
Remarks
This command requires that a program path and name be previously set using one of the following sets of commands:

GetUninstallProgramName

or:

GetUninstallProgramPath
SetUninstallProgramName

To add parameters to this uninstallation program set, add the following command and specify the appropriate parameters:

AddProgramParameters=[parameters]

To add a script for this uninstallation program set, add the following commands:

GetUninstallScriptPath
SetUninstallScriptName

This command combines the features of RunRobot and SetRegistryForUninstallSecondRun.

SendAffirmativeToWindow
The SendAffirmativeToWindow command sends the IDOK and IDYES messages to the window whose window name has been passed in. This shuts down dialog boxes that are not controlled by RunRobot. This command is used currently for Cheyenne products.

Syntax
sendaffirmativetowindow=windowname, waittime

Example
sendaffirmativetowindow=VshieldMonitor, 60

Parameters

windowname
The text in the header of the window. This parameter is used by FindWindowEx to locate the antivirus or firewall dialog box to which to send the message. This parameter is case-sensitive.

waittime
The maximum number of seconds to search for the window name.
SendAffirmativeToWindowAfterEnable

The SendAffirmativeToWindowAfterEnable command sends the IDOK and IDYES messages to the specified window. This command repeats until the window name is found. When the window name is found, the command searches for the specified button, and then enables the button. This command can be used to shut down the Uninstall Shield dialog box. This will shut down dialog boxes that are not controlled by RunRobot. This command is currently used for FProt 95.

Syntax

sendaffirmativetowindowafterenable=windowname, buttonname, waittime

Example

sendaffirmativetowindowafterenable=Remove Programs From Your Computer, OK, 60

Parameters

windowname
- The text in the header of the window. This parameter is used by FindWindowEx to locate the antivirus or firewall dialog box to which to send the message. This parameter is case-sensitive.

buttonname
- The text on the button for which you are searching. This parameter is case-sensitive.

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

SendAffirmativeToWindowWithButtonClick

The SendAffirmativeToWindowWithButtonClick command simulates a button click in a dialog box. It sends a BN_CLICK message to the parent window of the specified button.

Syntax

SendAffirmativeToWindowWithButtonClick=WindowName, ButtonName, WaitTime

Example

SendAffirmativeToWindowWithButtonClick=Confirm File Deletion,&Yes, 60
Parameters

*WindowName*

The text in the caption of the dialog box that is displayed. This command is used to locate the dialog box.

*ButtonName*

The text on the button. This is used to locate the button on the dialog box.

*WaitTime*

The maximum number of seconds to search for the button.

Remarks

This command should be used in place of `SendAffirmativeToWindow`.

SendINIEntry

The `SendINIEntry` command allows adding or editing of key values in Windows INI files.

Syntax

`SendINIEntry=INI File Name, INI Section Name, INI Key Name, String to Write`

Example

`SetINIEntry=ofcscan.ini, INI_CLIENT_SECTION, uninstall_pwd, 70`

Parameters

*INI File Name*

INI file name

*INI Section Name*

INI section name to write to

*INI Key Name*

INI Key name to write to

*String to Write*

Value to write to the INI file

Remarks

None.

SendTextToEditWindow

The `SendTextToEditWindow` command sends a text string to an edit control on a dialog box.
Syntax
SendTextToEditWindow=WindowName, String to Write, WaitTime

Example
SendTextToEditWindow=Uninstall OfficeScan Client, 1, 60

Parameters
- **WindowName**
  - The text in the caption of the dialog box that is displayed. This is used to locate the dialog box.
- **String to Write**
  - The text to write to the edit control.
- **WaitTime**
  - The maximum number of seconds to search for the button.

Remarks
The *SendTextToEditWindow* command can be used for filling in Password dialog boxes. It assumes that the dialog box has a single edit window. If there is more than one edit window, the text is written to the first edit window found.

SendWindowsMessage

The *SendWindowsMessage* command sends a message to the window whose class has been passed in. This command is designed to shut down antivirus or firewall programs that might be running so that they can be properly uninstalled.

Syntax
sendwindowsmessage=windowname, ASCIIMessage

Example
sendwindowsmessage=VshieldMonitor, 16
Parameters

`windowname`

The text in the header of the window. This parameter is used by `FindWindowEx` to locate the antivirus or firewall dialog box to which to send the message. This parameter is case-sensitive.

`ASCII message`

The ASCII value for the message to be sent.

Remarks

Message 16 is the WM_CLOSE. This command cannot pass the wparam or the lparam.

SendWindowsMessageAndWait

The `SendWindowsMessageAndWait` command sends a 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 command shuts down windows that are not controlled by `RunRobot`.

Syntax

`sendwindowsmessageandwait=windowname, ASCII message, waittime`

Example

`sendwindowsmessageandwait=VshieldMonitor, 16, 60`

Parameters

`windowname`

The text in the header of the window. This parameter is used by `FindWindowEx` to locate the antivirus or firewall dialog box to which to send the message. This parameter is case-sensitive.

`ASCII message`

The ASCII value for the message to be sent.

`waittime`

The maximum number of seconds to search for the window name.

Remarks

Message 16 is the WM_CLOSE. This command cannot pass the wparam or the lparam.

SetRegistry

The `SetRegistry` command sets the specified registry key value.
Syntax

\texttt{setregistry=hKey, SubKey, Key, Value}

Example

\texttt{setregistry=HKEY\_CURRENT\_USER, SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows, run, E:\S32UI\debug\s32ui.exe}

Parameters

- \textit{hKey}
  - The HKey that is passed into RegCreateKey/RegOpenKey
- \textit{SubKey}
  - The SubKey that is passed into RegCreateKey/RegOpenKey
- \textit{Key}
  - The Key that is passed into RegSetValueEx
- \textit{Value}
  - The Key value that is passed into RegSetValueEx

SetRegistryForUninstallSecondRun

The \texttt{SetRegistryForUninstallSecondRun} command sets the Run registry setting so that after the RestartComputer command is performed, the Security Software Uninstaller automatically restarts.

Syntax

\texttt{setregistryforuninstallsecondrun}

Example

\texttt{setregistryforuninstallsecondrun}

Parameters

N/A

SetUninstallProgramName

The \texttt{SetUninstallProgramName} command combines the member variable set by GetUninstallProgramPath and the file name that is passed into this command. RunRobot uses the result of this command.

Syntax

\texttt{setuninstallprogramname=filename}
Example
setuninstallprogramname=uninstall.exe

Parameter
  filename
  The file that performs the uninstallation

SetUninstallScriptName

The `SetUninstallScriptName` command combines the member variable set by `GetUninstallScriptPath` and the file name that is passed into this command. `RunRobot` uses the result of this command.

Syntax
setuninstallscriptname=filename

Example
setuninstallscriptname=install.log

Parameter
  filename
  The script that is used by the uninstallation program

SetupForSecondRun

The `SetupForSecondRun` command reads the registry and prepares for a restart when a second pass is required. Information that is needed on the second pass and the Run registry settings are written.

Syntax
setupforsecondrun=runkeyname

Example
setupforsecondrun=run

Parameter
  runkeyname
  For Windows NT, the runkeyname is run. For Windows 95, this parameter is the name that is in the RunOnce key of the HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion registry key.
Remarks
■ For Windows NT, this command only needs to be used if the antivirus or firewall uninstallation program is different than the one that was run on the first pass.
■ For Windows 95, this command should be placed just before the RestartComputer or PauseForNotificationOfCompletion command for uninstallation programs that require a restart.

StopProgramForReboot
The StopProgramForReboot command prepares for a restart by the uninstallation program when a second pass is needed. Information that is needed for the second pass and the Run registry settings are written.

Syntax
stopprogramforreboot

Example
stopprogramforreboot

Parameters
N/A