Addendum to Competitive Uninstaller

(Version 2.02 – 12/30/2003)

Using the Competitive Uninstaller

The Competitive Uninstaller will only uninstall a single application per run. Therefore, if you have a machine with multiple applications to remove, you will need to continue to run the Uninstaller until no applications are found. The easiest way to do this is to run the program using the /all switch which scans all script files looking for applications to uninstall.

Some anti-virus applications consist of multiple programs. In this case separate scripts have been created for each of the programs and the Competitive Uninstaller must be run multiple times until no more applications are found to remove.

Working with Script Files

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 directly calls the application's uninstall program and sends responses to dialogs through script commands. If you know exactly which dialog boxes will appear (as is true for most cases), RunProgram is more reliable and is recommended because it has less timing dependencies. Otherwise, the RunRobot command can be used to assist with detection and responses to dialogs.

Usage and switches

Starting the Competitive Uninstaller (AV32.exe) from the command line without any parameters will display usage details as follows:

```
Uninstalls applications using script files (*.aut)
See Uninstaller.pdf for instructions.

        Script          Specifies the script file to process.
        /all            Scans and processes all script files in the current directory
        /ip             Causes script file platform specifier to be ignored.
```

Where:

Script

Name of the script file to process.

/all

The /all switch causes the Uninstaller to execute each script file (*.aut) found in the current directory. The names of each of the processed scripts are echoed to the screen with a short status. For detailed information for a particular run, see the STATUS.INI file.

/ip

The /ip (**I**gnore **P**latform) switch causes the Uninstaller to ignore the platform specified in script files. This allows the user to run a script file against all platforms or run older script files against newer platforms.

Script Command Reference

SendAffirmativeToWindowWithButtonClick

This command simulates a button click in a dialog box. It will send a BN_CLICK message to the parent window of the specified button.

Syntax

SendAffirmativeToWindowWithButtonClick=*WndowName, ButtonName, WaitTime*

Example

SendAffirmativeToWindowWithButtonClick=Confirm File Deletion,&Yes, 60

Parameters

WndowName

The Window name is the text in the caption of the dialog window being displayed. This is used to locate the dialog window.

ButtonName

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

WaitTime

Specifies the maximum number of seconds to search for the button.

Remarks

This command should be used in place of SendAffirmativeToWindow.

SendINIEntry

This 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 Filename

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

This 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

WndowName

The Window name is the text in the caption of the dialog window being displayed. This is used to locate the dialog window.

String to Write

Text to write to the edit control.

WaitTime

Maximum number of seconds to search for the button.

Remarks

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