freêëMÉG·

SmartLog Login Processor

The FreeMEG Smart Logon Processor

Version 1.1 Release Documentation

What is SMARTLOG?

SmartLOG is a login processor designed to be used as the shell for a login script. It can be used in a group policy, NETLOGON, Novell Logon, Registry or simply placed in the startup section of the startmenu.

SmartLOG improves on the traditional batch file in the following areas:

- A large variable set

  Unlike a batch file that must declare an environment varible, SMARTLOG stores Variables in memory and can pass them to the registry, batch files and even as parameters on executables. Variables can be incremented and decremented, they can also be used in comparison with other Variables.

  SmartLOG can also read environment Variables and some system Variables that normal batch files cannot see.

  In addition to this, 100 custom named Variables can be defined and stored in smartlog.

- More complicated IF statements

  Comparison can be made between numbers, can be where one string contains another, can be where one file is newer than another.

  All if statements can be nested.

- Direct Registry Access.

  SmartLOG can access the registry directly, it can edit and read parts of the registry, it can store entries as Variables that can be sent to all sorts of other programs.

- Hidden Programs

  SmartLOG can run 3rd party programs hidden, so that they run as part of the login but the user cannot abort them.

SmartLOG itself will run without the user being able to close smartlog.
You can even run smartlog hidden giving total gurantee the login script will run.

- LOOPING

  SmartLOG provides looping support via the means of GOTO statements. Allowing the same code to run over and over again until a certain event is true.

- WatchDOG Lockup Protection

  SmartLOG will detect if the login script has locked up and will offer to terminate. This timer can be customized.

- Customizable Look

  SmartLOG can be customized to your enterprises desktop look and feel. Change the colour of the text and or setup your own picture in the splash screen.

Installation of SmartLOG

SmartLOG is basically installed into a location on your network or computer. The following files are required in the same directory as smartlog.exe

script.txt – The script file smartlog will use
bg.bmp – A customized Bitmap with your companys logo.

Login Script Commands

This is a list of commands that can be placed in script.txt
These are in alphabetical order.

A Quick note on Variables

If any command requires to read a varible place it with % signs around it. Example

ECHO %hello%
CALL c:\%userprofile%\User.bat

If any command requires a variable to write to simply place the varible name without the % signs around it. Example

ASSIGN myvariable "C:\Windows"
INC varibléname 200
ASSIGN hello "This is my hello message"

A Quick Note on Remarks

Any line of text can be remarked. Simply place a ; in front of the text or the command.

Example:

;this is a remarked line of text
;maybe to explain the next few lines of code.

Commands can also be remarked so that still appear in the script but won't be executed. As per the following example
;ALWAYSONTOP

Text or Variables that have spaces

If you need to put spaces in the text then you should encapsulate the entire string in " "
Example "This text contains spaces"

This is also important if you use variables that possibly contain spaces. Example:
if %fullname% equals **Fiona Cook** then use "%fullname%" instead of just %fullname%

Now for the commands:

Command: ALWAYSONTOP
Parameters: *none*
Description: If this command is present in script.txt it will force the SMARTLOG splash screen to always be on top of other windows. Including windows that are called by SMARTLOG.
Example: ALWAYSONTOP

Command: APPENDFILE
Parameters: *filename text*
Description: This command will write the contents of the text variable to a file specified by filename. If the file exists it will append it to the bottom as a new line. If the file does not exist it will create a new one.
Example: APPENDFILE "c:\test.txt" "%date% this file has been written to"

Command: ASSIGN
Parameters: variablename value
Description: This command will assign a user variable with a value. The value can be a text or numbers. SMARTLOG will decide how to treat it.
Example: ASSIGN serverpath "\\basesrv\netlogon\scripts\soe\"

Command: BGCOLOUR

Parameters: HTMLcolourcode
Description: This command will set the background of the smartlog window to the colour defined by the HTML colour code. *Note: HTML Word Colours like yellow or blue are not allowed.*
Example: BGCOLOUR #FF33FF

Command: CALL
Parameters: batchfile parameters
Description: This command will run a batch file using the CMD command interpreter. It will then wait for that command to finish before continuing with the logon script. You can send parameters including Variables to the batch file.
Example: CALL "%serverpath%\runme.bat" "%username%.ou=staff.o=nds"

Command: CALLNOWAIT
Parameters: batchfile parameters
Description: This command acts just like the **CALL** command except that it will not wait for the batch file to end before continuing with the logon script.
Example: CALL "%serverpath%\runme.bat" "%username%.ou=staff.o=nds"

Command: CLOSEKEY
Parameters: ***none***
Description: This command will close the registry key that is currently being edited in the registry.
Example: CLOSEKEY

Command: DEC
Parameters: variablename amount
Description: Decrements the variable specified by variable name by a certain whole integer amount.
Example: DEC counter 23

Command: DELETEFILE
Parameters: *filename*
Description: See ERASEFILE command.
Example: DELETEFILE "c:\test.txt"

Command: ECHO
Parameters: text
Description: Displays text on the main smartlog screen. Useful for giving user status of the scripts progress.
Example: ECHO "Updating Startmenu Icons"

Command: ERASEFILE
Parameters: *filename*
Description: This command will erase a file specified by filename. If the user has access to do this.
Example: ERASEFILE "c:\test.txt"

Command: EXECUTE
Parameters: command parameters [HIDE|SHOW]
Description: Runs a command (can be an executable, webpage or any file that has association with a program) with parameters either visible or hidden. By specifying hide or show.
Example: EXECUTE c:\windows\notepad.exe \\zfs\data\fast.txt SHOW

Command: GOTO
Parameters: labelname
Description: Causes the script to jump a location where labelname is located and continue running from their. This is good for reiteration of forks in the script. This command requires a corresponding LABEL command to work correctly.
Example: GOTO ipaddrconfig

Command: HIDEDRIVES
Parameters: driveletters
Description: Causes the drive letters listed to be hidden to the user in My Computer and Dialog Boxes. HIDEDRIVES with no parameters resets all drives to being visible. Changes may require a restart to take affect.
Example: HIDEDRIVES "ACEG"

Command: IF
Due to the complexity of this command it has its own section see below!

Command: INC
Parameters: variablename amount
Description: Increments the variable specified by variable name by a certain whole integer amount.
Example: INC counter 23

Command: LABEL
Parameters: labelname
Description: Defines a label in the script that can be jumped to using the GOTO command.
Example: LABEL startbginfo

Command: MOTD
Parameters: text
Description: Displays a Message of the day on the main SMARTLOG screen. Good for giving quick messages to the user
Example: MOTD "Internet is down until 10:00am today"

Command: MOTDCOLOUR
Parameters: HTMLColourCode
Description: Changes the font colour for the message of the day. To the colour specified by the HTML Colour Code.
Example: MOTDCOLOUR #663399

Command: NEWFILE
Parameters: *filename text*
Description: This command will write the contents of the text variable to a file specified by filename. If the file exists it will be overwritten with the contents in the text variable. If the file does not exist it will create a new one.
Example: NEWFILE "c:\test.txt" "%date% this file has been written to"

Command: READBOOLEAN
Parameters: RegistryKeyName VariableName
Description: Reads a boolean value from the registry and places it in a variable name. Useful for reading information out of the registry and putting into a batch file etc.
Make sure you have used the SETROOTKEY and SETKEY commands first.
Example: READBOOLEAN EnableProxyServer pseVar

Command: READDWORD
Parameters: RegistryKeyName VariableName
Description: Reads a dword value from the registry and places it in a variable name. Useful for reading information out of the registry and putting into a batch file etc.
Make sure you have used the SETROOTKEY and SETKEY commands first.
Example: READDWORD CommonPort cportvar

Command: READFILETOVAR
Parameters: *filename variable*
Description: Reads the contents of a file specified by filename and places that contents in the variable. Contents is limited to the amount of free memory.
Example: READFILETOVAR "c:\temp.txt" infovariable

Command: READSTRING
Parameters: RegistryKeyName VariableName
Description: Reads a string value from the registry and places it in a variable name. Useful for reading information out of the registry and putting into a batch file etc.
Make sure you have used the SETROOTKEY and SETKEY commands first.
Example: READBOOLEAN HomePage hpVar

Command: REPLACETEXT
Parameters: *text fromtext totext variable*
Description: Searches the text specified by the text parameter and replaces text specified by the fromtext parameter with text in the totext parameter then stores the lot in variable.
Example: REPLACETEXT "This is the text to change" "text to change" "text that changed" myphrase

Command: SETKEY
Parameters: RegistryPath AutoCreate
Description: Sets to the subkey located in the root key. The auto create parameter, if set to true will automatically create the key if it isn't already present in the registry.
Example: SETKEY "\Software\Microsoft\Windows\CurrentVersion" FALSE

Command: SETROOTKEY

Parameters: RootKey
Description: Will tell SmartLOG which ROOT Registry Key to use. This normally is set first before any other components of SmartLOG
Example: SETROOTKEY "HKEY_LOCAL_MACHINE"

Command: SHOWHIDDENWINDOWS
Parameters: ***none***
Description: Will force any windows that are normally hidden by SMARTLOG to show. This is normally in relation to the CALL and the CALLNOWAIT command.
Example: SHOWHIDDENWINDOWS

Command: SHOWMAINWINDOW
Parameters: ***none***
Description: Will force the main SMARTLOG window to be visible. Without this command the login script will be hidden but will still execute.
Example: SHOWMAINWINDOW

Command: SHOWMESSAGE
Parameters: text
Description: SmartLOG will show a message dialog box on the screen. This will show regardless of whether the SMARTLOG main window is visible or not. Note this will pause execution of the script until the user clicks okay.
Example: SHOWMESSAGE "Press OK to continue logon"

Command: SUPRESSERRORS
Parameters: *none*
Description: SmartLOG will not show any errors if they occur.
Example: SUPRESSERRORS

Command: TEXTCOLOUR
Parameters: HTMLColourCode
Description: Determines the colour of the font that displays in SmartLog.
Example: TEXTCOLOUR #003399

Command: WDTIMEOUT
Parameters: seconds
Description: Changes the default timeout for the watchdog timer before it determines that a lockup has occurred. Normally this is preset to 5 seconds, however if there is a slow connection or slow process this may cause SMARTLOG to think it has locked up. Increasing this number may fix this.
Example: WDTIMEOUT 45

Command: WRITEBOOL
Parameters: keyname value
Description: Writes a boolean variable to the registry. The value must be true or false. Use SETKEY and SETROOTKEY first to locate the registry value.
Example: WRITEBOOL proxyenabled TRUE

Command: WRITEDWORD
Parameters: keyname value
Description: Writes a dword variable to the registry. The value must be a number. Use SETKEY and SETROOTKEY first to locate the registry value.
Example: WRITEDWORD appearancecolour 3213312

Command: WRITESTRING
Parameters: keyname value
Description: Writes a string variable to the registry. Use SETKEY and SETROOTKEY first to locate the registry value.
Example: WRITEBOOL screensavetext "User %username% is logged on"

The IF Command

The if command provides complicated comparisons and then will run code based on whether or not the comparison is true or false.

The Basic Command:

```
IF value1 operator value2 command
```

Where

value1 and value2 can either be a static value or a variable
operator is =, <> (Not Equal), < (Less than), > (Greater Than), NEWER, CONTAINS

A simple example of an IF statement is here.

```
IF %username% = "BOGAJ" "GOTO specialdrivemap"
```

If statements can go over multiple lines by specifying a THEN and ENDIF block. Below is an example.

```
IF %username% = "BOGAJ" THEN
GOTO specialdrivemap
ENDIF
```

If statements can be nested in if statements for more complicated conditions requirements. Below is an example:

```
IF %username% = "BOGAJ" THEN
        IF %computername% = "HOMEMAIN" THEN
                GOTO specialdrivemap
        ENDIF
ENDIF
```

Note code can be tabbed to make it easier to read

Below are the different Comparison Operators and what they do:

Operator: =
Description: If values on the left and right hand side are exactly equal then the if statements is executed.
Example: IF %username% = "HARDS" THEN ...

Operator: <> or != (Not Equal)
Description: If values on the left and right hand side are different then the if statements are executed.
Example: IF %computername% <> "monda" THEN ...

Operator: <
Description: If the value on the left hand side is less than the value on the right hand side then the if statements are executed. Note this will only work with numbers.
Example: IF %numretries% < 3 THEN ...

Operator: >
Description: If the value on the left hand side is greater than the value on the right hand side then the if statements are executed. Note this will only work with numbers.
Example: IF %numretries% > 3 THEN ...

Operator: CONTAINS
Description: If the value on the left hand side is contains text that is listed on the right hand side then the if statements are executed. Note this will only work with numbers.
Example: IF %computername% CONTAINS "CR06" THEN ...

Operator: LIKE
Description: If the value on the left hand side is like the text (or is a substring of the text) that is listed on the right hand side then the if statements are executed. Note this will only work with numbers.
Example: IF "CR06" LIKE %computername% THEN ...

Operator: INLIST
Description: If the text on the left hand side is in a file given on the right hand side. The statement is true. The file must contain an exact match in the line of text
Example: IF %username% INLIST c:\names.txt THEN..

Operator: INLISTCONTAINS
Description: If the text on the left hand side is contained on one of the lines in the file (one of the lines contains text like the text on the left hand side) given on the right hand side. The statement is true.
Example: IF %username% INLISTCONTAINS c:\names.txt THEN..

Operator: INLISTLIKE
Description: If the text on the left hand side is like (or a sub string) of one of the lines in the file given on the right hand side. The statement is true.
Example: IF %username% INLISTCONTAINS c:\names.txt THEN..

Operator: NEWER
Description: If the filename on the left hand side has a modified date that is NEWER than the modified date on the right hand side then the if statement will run.
Example: IF "%srvprofile%\NTUSER.DAT" NEWER "%userprofile%\NTUSER.DAT" THEN

Operator: EXISTS
Description: If the file exists then the if statement will run
Example: IF EXISTS "C:\BOOTLOG.TXT" THEN ...

Operator: PING
Description: If the server can be pinged successfully then run the it statement otherwise don't run the if statement
Example: IF PING powerup.com.au THEN ...

Statically Defined Variables

The following variables are statically defined.

%username% - The username of the currently logged on user.
%ipaddress% - The IpAddress of the primary adapter
%hostname% - The Hostnamt or netbios name of the Windows Computer
%date% - The current date
%time% - The current time
%startdir% - Returns the path of where smartlog.exe runs from.