RSIScript Reference Manual
A Scripting Language for RSIGuard v4.0

Table of Contents

Revision Date: 8/15/06

Overview

RSIScript is a powerful scripting language that can be used to control RSIGuard and perform simple user-interface functions related to RSIGuard.

This document explains how to use RSIScript, and provides detailed documentation of the scripting language and its commands. It also describes how to create RSIGuard scripts to globally and semi-globally manage RSIGuard user configuration, send users messages, track RSIGuard usage. Additionally, it describes how to create macros and hotkey-triggered macros to perform useful functions in order to reduce keyboard and mouse-related strain.

Although it does not require previous programming knowledge, it does require a basic understanding of the concept of a scripting language (e.g. MSDOS batch files or Unix cshell scripts).

What Goes in an RSIScript File?

An RSIScript file should be created as you would create any other text file (e.g. with a text editor like Windows Notepad). Editors like Microsoft Word or Wordpad can be used, but when saving the file, you must specify that you want plain text format. RSIGuard defines the file extension RXS to be an RSIScript file, although TXT works fine too (the only difference is that if you open an RXS file, RSIGuard can automatically run the script).

The syntax (i.e. format) of an RSIScript file is as follows:

Each line in an RSIScript file can either: be blank; contain a command; or contain a comment (a documentation message). The following 3-line file, for example, sets the organization name. The first line is a comment because it starts with '#'. It is only there to remind you of what this script does. The third line actually sets the organization name.

```
# This script will set our organization's name within RSIGuard

set m\RSIGuard\OrganizationName ACME, Inc.
```

Command lines must begin with a command, and be followed by any options (also known as arguments) that are appropriate for the command.

Special RSIScript Files

During the course of normal operation, RSIGuard will try to run certain special scripts, if you have created them, to allow you to control and customize RSIGuard operation. These are analogous to DOS scripts like Config.sys and Autoexec.bat or Unix scripts like .login, .cshrc, etc. Special RSIGuard scripts can be created that:

- Run when a copy of RSIGuard is registered (i.e. one-time actions you want to occur on each RSIGuard computer). RSIGuard Software staff will generally create this "registration" script file that sets the default machine-based settings you request (e.g. organization name), and otherwise customizes RSIGuard to the needs of your organization. This script is generally not located on the user's computer – rather it is transmitted when the registration code is entered from the RSIGuard website.
- **Register.txt:** Run right after RSIGuard is registered. If the file Register.txt exists in the installation folder, Register.txt will be executed after registration (in addition to any script returned by use of a web-based registration code (i.e. one that starts with 'x')).
- **InstRun.txt:** Run the first time RSIGuard ever runs on a computer (uninstalling and reinstalling will NOT cause this script to run again because the first run is flagged in a registry value). This script can do initial configuration tasks that do not require an RSIGuard profile (e.g. set u\$p* is not allowed at this point). This script should be in a file InstRun.txt stored in the RSIGuard executable folder.
- **FirstRun.txt:** Run the first time a new version of RSIGuard is run for the first time. This script can do initial configuration tasks that do not require an RSIGuard profile (e.g. set u\$p* is not allowed at this point). This script should be in a file FirstRun.txt stored in the RSIGuard executable folder. This script is run immediately after InstRun.txt if this is the first run after installation. This file is often included in an MSI package for registering RSIGuard or for establishing a connection to a Remedy Interactive OES database.
- **New user script:** Run each time a new user or profile is created (i.e. actions you want to occur separately for each user and each users's various profiles). This script can be used to do things like set a default break frequency. This script is by default stored in the installation folder under the name newuser.txt, but you can change that (e.g. in the registration script with a command like "set m\RSIGuard\NewUserScript http://yourdomain.com/newusr.txt")
- **Startup scripts:** Run each time RSIGuard starts up (i.e. to perform actions you want to occur each time a user starts using RSIGuard). There are 3 different scripts that can be run on startup (all of which by default are located in the installation folder).
    1. **PreInit.txt** – This script runs before RSIGuard itself begins. It can contain only very simple commands. For example, it might contain a command like "wait 5" to wait for a network resource to become available after login before RSIGuard starts.
    2. **Init.txt** – This script runs after RSIGuard has started and can be used affect global settings.
    3. **Startup.txt** – This script runs after the RSIGuard user is online and can therefore be used to modify user settings such as break frequency. You can change the default location of this script with a command (e.g. in the registration script) like: "set m\RSIGuard\StartupFile S:\Net\RSIGuard\start.txt"

Line Preprocessing Commands

Before each command line is executed, RSIScript preprocesses the line by replacing various escape sequences with other text. This allows for some simple string manipulation and processing as well as some sophisticated logic control The following escape sequences are recognized.

aliasing

- $1-$9 are replaced by the values of the 9 general usage variables assigned via the setvar command

internet checking

- $c is replaced by 0 or 1 depending on whether or not the computer is connected to the internet

date usage

- $d the following integer value should be replaced by the date/time it represents

user identification

- $i is replaced by the users RSIGuard ID#
- $n is replaced by the name of the current user profile (e.g. Joe Smith)
- $p is replaced by the profile subkey string for the current user (see the 'set' command below) (e.g. _JOESMITH)
- $u is replaced by the username of the currently logged in Windows user (e.g. jsmith)

version checking

- $r is replaced by the RSIGuard Edition/Version (e.g. RSIGuard Stretch Edition v3.0.37)
- $v returns the Windows version (95=4.0 98=4.10 ME=4.90 NT3.51=3.51 NT4.0=4.0 2000=5.0 XP=5.1)
- $w returns 0 if Windows platform is Win95/98/ME or 1 if Windows platform is WinNT/2000/XP

variable usage

- $val(SETTING,[DEFAULT-VALUE]) – substitutes with the specified RSIGuard setting (see *set* command for details on specifying SETTING). If SETTING does not exist, DEFAULT-VALUE will be used if present.
- $sSETTING is replaced by value of SETTING (e.g. $sm\RSIGuard\DataFolder). This escape sequence is obsolete and remains for backaward compatibility – use $val() instead.
- $env(ENVVAR) – substitutes with the specified environment variable (see *setvar* command)

special characters

- $$ is replaced by the $ symbol itself
- $ch(NUM) – substitutes the character of the specified character code (e.g. $ch(65)=A)

logic tests

- $equal(OP1,OP2) – substitutes a "1" if OP1 equals OP2 or a "0" if not (i.e. 1=TRUE, 0=FALSE).
- $lessthan(OP1,OP2) – substitutes a "1" if OP1 is less than OP2 or a "0" if not
- $lessthanequal(OP1,OP2) – substitutes a "1" if OP1 is less than or equal to OP2 or a "0" if not

math operations

- $plus(OP1,OP2), $minus(OP1,OP2) – integer addition (OP1+OP2) and subtraction (OP1-OP2)
- $times(OP1,OP2), $divide(OP1,OP2) – integer multiplication (OP1*OP2) and division (OP1/OP2)

string operations

- $left(STR,NUM) – replaces with first NUM chars of STR
- $right(STR,NUM) – replaces with last NUM chars of STR

- $mid(STR,NUM1,[NUM2]) – replaces with substring starting at NUM1'th position, using NUM2 chars (or going to end of string). NUM1 is 0-based (i.e. $mid(STR,0) is entire string).

logic operations

- $or(OP1,OP2) – substitutes a "1" if either OP1 or OP2 is not "0" (i.e. if either OP is true).
- $xor(OP1,OP2) – substitutes a "1" if either OP1 or OP2 is not "0", but not both (i.e. if one OP is exclusively true).
- $and(OP1,OP2) – substitutes a "1" if OP1 and OP2 are not "0" (i.e. if both OPs are true).
- $not(OP1) – substitutes a "1" if OP1 is "0", otherwise, substitutes a "0" (i.e. toggles OP between true and false).
- $if(CONDITION,TRUESUB,FALSESUB) – substitutes FALSESUB if CONDITION is "0", otherwise substitutes TRUESUB.
- $infile(KEY,FILE) – substitutes a "1" if the string KEY appears in file FILE

multiple command bracketing

- $do(OP1,…,OPn) – Performs OP1…OPn as commands.

Here are some examples of how to use line preprocessing effectively (colors shown in examples are only to help visualize the different operands to the preprocessing commands).

Example1: ask a user with the high willpower setting if they'd like to switch to medium willpower
$if($equal($val(u\$p\BreakTimer\WillpowerLevel),0),"$do(msg text $u you might want to switch to Medium willpower,msg send)")
This uses $equal() to test if the current willpower setting is 0, which is the high-willpower setting. If so, it replaces the command line with the $do() statement. Note that quotes are needed around $do() since we don't want the $do() preprocessing step to occur unless the condition is true. Alternately, the $do statement could have called another script that uses the 'question' command to actually change the willpower setting.

Example 2: automatically register pre-selected users, and tell other users this is a trial
$if($infile($u,s:\RSIGuard\userlist.txt),"$do(register xacme1234)","$do(msg text "Tell your supervisor if you want to use RSIGuard",msg send)")
This uses $infile() to test if a user's login name appears in a text file (s:\RSIGuard\userlist.txt in this example). If the name appears in the file, the user is registered with the "register xacme1234" command. Otherwise, they are allowed to continue but told to tell their supervisor if they wish to use RSIGuard.

Example 3: automatically determine who should run RSIGuard from a username list
$if($not(infile($u,s:\RSIGuard\userlist.txt)),"$do(exit)")
This uses $infile() to test if a user's login name appears in a text file (s:\RSIGuard\userlist.txt in this example). If the name is not in the file, RSIGuard exits. This can be used to control who uses RSIGuard by scripting that all users run RSIGuard, but only those in the list will continue to run past the initial start up.

Example 4: disable ability to send Health Status Reports for people not in the Engineering department
$if($not($equal($val(u\$p\Reports\Department),Engineering)),"$do(set u\$p\GeneralSetup\AccessSendHSR 0)")
This uses $not($equal()) to test if user's Deparmtent is<u>n't</u> Engineering, & if not, disables ability to send Health Status Reports.

Script Commands Descriptions

activate – bring a specified window to the frontmost (activated) position

Syntax options:

activate WINDOWNAME

Notes:

This command allows you to bring a window/application to the foreground (useful, for example, for the 'type' command to insure where the typing ends up). The WINDOWNAME is the name that appears in the Window Title Bar. For example, to activate a Notepad Window with an untitled file, you'd use "activate Untitled – Notepad".

---

addkckey – create hotkeys

Syntax options:

addkckey KEY ID CATEGORY OPTIONS NAME

Notes:

This command allows you to create scripts that create hotkeys (e.g. for configuring a new hotkey by clicking on a script on a web page). Using this command requires assistance from RSIGuard support. Please contact us if you have a need to create hotkeys for your users. KEY is a binary mask for Ctrl/Alt/Shift/Win followed by the key's keycode (e.g. 101065 for Ctrl-Shift-A). A list of keycodes is shown in http://www.rsiguard.com/DataLoggerAnalysis.pdf in the "Extended DataLogger Tools" section. ID is an internal ID (256=Single Click, 257=Double Click, 258=Triple Click, 259=Right Click, 260=Left DragLock, 270=Right DragLock, 271=Skip Next Click, 272=Middle Click, 273=Middle Drag Lock, 512=RSIGuard-To-Front, 513=Hide RSIGuard, 514=Take A Break Now, 515=Show ForgetMeNot Now, 516=Enable/Disable AutoClick, 32768 for any other hotkey). CATEGORY is the hotkey category (0=RSIGuard Function, 1=Launch Application, 2=Open File, 2=Open Web Page, 4=Type Text, 5=Insert Text File, 6=Run RSIScript, 7=Windows Operations). OPTIONS is the "argument" to the hotkey (use "" if no argument is needed). Name is the hotkey's name.

---

addprofile – adds a new user profile

Syntax options:

addprofile PROFILENAME

Notes:

Adds a profile. Generally, this command should only be used in an automated user-setup process. It is not a recommended way to create new profiles.

---

alertset – change RSIGuard settings and notify user of change

Syntax options:

alertset SETTING number/text VALUE (set the value of the RSIGuard setting specified by SETTING to VALUE)

Notes:

This command is identical to the 'set' command described below, except that, if the current value of SETTING is not already VALUE (i.e. the setting needs to be updated), the user will receive a message letting them know that the setting is being changed.

---

bfedit – modify the list of BreakTimer filters

Syntax options:

bfedit add active/running disable/polite APPLICATION-TITLEBAR-TEXT
bfedit delete APPLICATION-TITLEBAR-TEXT
bfedit deleteall

Notes:

Commands to add and remove BreakTimer filters.
To add a filter, use a command like “bfedit add running disable PowerPoint Slide Show”. The ‘active’ option means this filter applies when the application is the active foreground application. The ‘running’ option means this filter applies if the application is running at all. The ‘disable’ option means that while the condition (active or running) is true, the BreakTimer should be disabled. The ‘polite’ option means that while the condition is true, BreakTimer should be in polite mode (where a “Break Needed” button must be clicked to start the break). The application name corresponds to the text in the title-bar of the application window.
To delete a filter, you need only include a fragment of the application name for the filter you wish to delete. For example, you could use either “bfedit delete PowerPoint Slide Show” or “bfedit delete PowerP”.
To delete the all BreakTimer filters, use “bfedit deleteall”.

---

discomfort – notate where a user is experiencing discomfort for HSRs

Syntax options:

discomfort WristHand/NeckUpperBack/Shoulder/ElbowForearm/WristHand 1/0

Notes:

Tells RSIGuard to set discomfort points associated with the specified area to be set (1) or unset (0). This command is for use by third-party applications that wish to set HSR values.

---

exit – terminate RSIGuard

Syntax options:

exit

Notes:

Exits RSIGuard.

---

fmnedit – modify the list of ForgetMeNots

Syntax options:

fmnedit add FORGETMENOTMESSAGE
fmnedit delete FORGETMENOTMESSAGEFRAGMENT
fmnedit deleteall
fmnedit addex DISPLAYFREQUENCY SOURCECHARACTER FORGETMENOTMESSAGE
fmnedit deletefromsource SOURCECHARACTER

Notes:

Commands to add and remove ForgetMeNot messages. To add a message, use a command like "fmnedit add Remember to rest". You can add line-breaks by embedding
 within the message (e.g. 'fmnedit add Remember
***This
***That'). This will also cause messages to be drawn left-justified instead of centered.
To delete a message, you need only include a fragment of the message you wish to delete. For example, you could use either "fmnedit delete Remember to rest" or "fmnedit delete Remember" or "fmnedit delete to rest" to delete the message added in the example above. To delete the entire list of ForgetMeNots, use "fmnedit deleteall".
Two additional commands are normally not used by users but external 3rd party tools. 'fmnedit addex' allows you to add a ForgetMeNot but additionally specify the frequency of display (from 1 to 100, 1 means extremely rarely, 100 means often, 50 is default value for 'fmnedit add'). It also lets you specify a source for the item. It defaults to 'U' with 'fmnedit add' (the U stands for User). But the source can be other things which allows you to use 'fmnedit deletefromsource'. 'fmnedit deletefromsource' lets you delete all strings from a particular source. 'S' is used by the system.

---

fset – remotely control RSIGuard settings and specify data type

Syntax options:

fset SETTING number/text VALUE (set the value of the RSIGuard setting specified by SETTING to VALUE)

Notes:

Exactly like the 'set' command except the number/text argument lets you specify if the data should be stored as a number or a string.

---

insertfile – insert the contents of the specified text file into the current application

Syntax options:

insertfile FILENAME

Notes:

Inserts the text stored in the file FILENAME into the application with edit focus. It does so by copying the text into the Copy buffer and then pasting it into the current application in focus.

Example:

insertfile C:\My Documents\StandardEmail.txt

---

integrate – command to integrate HR data from a database into RSIGuard reporting

Syntax options:

Integrate enc/clr DATABASE

Notes:

The 'integrate' command causes RSIGuard to search the database file for HR data for the logged in user, and to store the relevant data. The 'enc' option states that the database has been encoded. The 'clr' option states that the database is clear text. The format for the clear text database is a series of lines each with the following information:

```
Windows Login Name,Full Name,Email,EmployeeID#,Location,Department,Script
```

If fields include a comma, they must be enclosed in quotes.

---

log – control the script log file

Syntax options:

log file FILENAME (specify a file to log script information to)
log error 1/0 (specify if errors should be logged, default=1)
log info 1/0 (specify if informational msgs should be logged, default=1)
log msgs 1/0 (specify if the 'msg log' command actually writes to the log file, default=1)

Notes:

The log file gets 3 kinds of messages. Error messages are added to the log file when a command in the script file is invalid or results in an error. Info messages are added by certain commands (e.g., the 'register' command logs when a user is first registered, and the 'set' command logs your changes to a user's setup). You can create your own messages and write them to the log file (or display them on the user's screen) with the 'msg' command.

Example:

```
# define where to write the log file
log file \\NETDRIVE\RSIGuard\log.txt
# this next line specifies that error messages should be discarded and not written into the log file
log error 0
msg text This message is added to log.txt each time a user comes online in RSIGuard
msg log
msg clear
# this next line specifies that error messages should be reenabled for subsequent errors
log error 1
```

---

menuedit – modify the list of Soft Menu Items

Syntax options:

menuedit add SOURCECHARACTER MENUNUM INSERTPOSITION “MENUITEMTEXT” COMMAND
menuedit deleteall
menuedit deletefromsource SOURCECHARACTER

Notes:

Commands to add and remove soft menu items. To add an item, use “menuedit add”. The SOURCECHARACTER identifies the category of the menuitem for the purpose of deleting all items from a particular category. It is recommended to use source ‘U’ for user items. MENUNUM is 0 for Tools, 1 for Setup, 2 for Help. INSERTPOSITION tells where in the menu to put the item (1 is first, 2 is second, etc., and 0 means last item in menu). “menu deletefromsource” deletes all menu items in the specified category. “menu deleteall” deletes all soft menu items.

Example:

menuedit add G 0 0 “Visit RSIGuard Website” “open http://www.rsiguard.com”
menuedit deletefromsource G (deletes only the above menu item and others created with a source character of ‘G’

---

mouse – control mouse movement and clicking

Syntax options:

mouse click/dblclick/rtclick/ldown/lup/rdown/rup/mdown/mup
mouse relmove delx dely
mouse absmove client/screen xpos ypos

Notes:

The first usage line does mouse clicking. ‘click’ is a single left click. ‘dblclick’ is a double left click. ‘rtclick’ is a single right click. ‘ldown’ presses and holds down the left mouse button. ‘lup’ releases the left mouse button after an ‘ldown’. ‘mdown/mup’ and ‘rdown/rup’ do the same for the middle and right buttons.

‘mouse relmove’ moves the mouse the distance specified from its current position. Distance is specified in pixels (screen dots).

‘mouse absmove’ moves the mouse to an absolute position on the screen. If the third argument is ‘client’ then the coordinates are relative to the client area in the currently active window. If the third argument is ‘screen’ then the coordinates are relative to the upper left corner of the screen.

msg – create messages that can be stored in the log file or displayed on a user's screen

Syntax options:

msg text TEXT (append TEXT to the current message buffer)
msg time (append the current time & date to the message buffer)
msg send (display the current message buffer on the user's screen and then clear the message buffer (i.e. 'msg clear'))
msg log (write the current message buffer to the log file (but don't reset the message buffer (see 'msg clear'))
msg post URL (posts the built string to the specified URL)
msg clear (clear the message buffer. This only needs to be used between messages that are going only to the log file.)

Notes:

The 'msg' command allows you to format a message into a 'message buffer' for writing to the log file and/or displaying on a user's screen at startup time. You create a message by appending pieces of the message to the message buffer and then writing it to the log file ('msg log') or sending it to the user's screen ('msg send'). Sending the message to the user's screen with 'msg send' also clears the message buffer (in preparation for creating your next message). (The only reason you need to explicitly write 'msg clear' between commands that you write to the log file is that if you want to send a message to both the log file and the user's screen, you build the message buffer, then call 'msg log' then 'msg clear'. If 'msg log' also cleared the message buffer, you'd have to rebuild the message twice to send it both places.) The 'msg post' command enables advanced users to execute web POST commands (e.g. 'msg text file=log.txt&info=hello', 'msg post http://myweb.com/log.cgi' would execute 'http://myweb.com/log.cgi?file=log.txt&info=hello)

Example 1:

```
msg text Hello
msg $n
msg send
msg text The current time is
msg time
msg send
```

Example 2:

```
msg text User
msg $n
msg text logged in at
msg time
log msg
msg clear
msg text User's break-enabled state is
msg setting u\$p\BreakTimer\Enabled
log msg
msg clear
```

---

nop – performs no function. A placeholder that can be useful in certain places.

Syntax options:

nop

Notes:

Does nothing. Can be used, for example in a conditional statement, to indicate that nothing should be done in one of the conditional results.

---

open – open a file, folder, application or webpage

Syntax options:

open FILENAME
open FOLDER
open APPLICATION COMMANDOPTIONS
open WEBPAGE

Notes:

Opens/launches the specified item.

Example:

open “c:\My Documents\phonelist.txt”
open “c:\My Documents”
open notepad.exe
open notepad.exe “c:\My Documents\phonelist.txt”
open http://www.rsiguard.com

---

queryset – ask user for the value of a setting

Syntax options:

queryset SETTING number/text PROMPT (ask user for the value of SETTING)
queryset SETTING number/text PROMPT VAL1 VAL2 … VAL-N (ask user for value of SETTING from N choices)

Notes:

This command is like the 'set' command described above, except that here the user provides the value for the setting. If values are given, then instead of a freeform dialog, a combobox with the provided options is shown to the user.

Example:

queryset m\RSIGuard\DataFolder “Enter the network path for the data folder:”
queryset m\RSIGuard\DataFolder “Enter the network path for the data folder:” “S:\Group1\Data” “S:\Group2\Data”
queryset m\Custom\Value “Enter general value” “value 1” “value 2” “value 3”

---

question – ask the user a question, and execute a command based on the answer

Syntax options:

question prompt PROMPT (sets the question to be asked)
question caption CAPTION (sets the caption of the popup window)
question answer ANSWER (adds another possible answer to question)
question ask CMD1 [CMD2…] (asks the question and executes associated script)

Notes:

Allows conditional execution of a command based on a user's answer to a question. Up to 5 answers can be given, and the number of specified CMDs in the "question ask" command must correspond to the number of "question answer *" commands issued.

Example:

question prompt Which division are you in?
question answer "Laser research"
question answer "Materials processing"
question answer "Data analysis"
question ask "read c:\laser.txt" "read c:\materials.txt" "read http://www.acme.com/analysis.txt"

question prompt Visit which website?
question caption Website Decision…
question answer "RSIGuard"
question answer "Google"
question ask "open http://www.rsiguard.com" "open http://www.google.com"

question prompt Please specify your office location?
question caption Location
question answer California
question answer New York
question ask "$do(msg text The Helpdesk is at x1234,msg send)" "$do(msg text The Helpdesk is at x5678,msg send)"

---

read – execute a script file

Syntax options:

read FILESPEC (process script file at FILESPEC)

Notes:

Execute commands in a script file. Calls into script files can be nested arbitrarily deeply. FILESPEC can be either a local filename, a network filename, or an http:// based URL.

Example:

read c:\RSIGuard Scripts\doit.txt
read http://www.yourcompany.com/rsiscripts/doit.txt

---

refresh – refresh the user interface state (buttons, menus, etc.)

Syntax options:

refresh

Notes:

After some script commands (e.g. set), the user interface may need to be updated. For example, if you enable/disable a menu item or if you turn AutoClick/BreakTimer/ForgetMeNots on/off, the user-interface must be updated to reflect this. Use the refresh command after such operations so that the user interface is immediately updated.

---

register – register RSIGuard using a specified registration code

Syntax options:

register REGISTRATIONCODE (registers user if they are not currently registered)

Notes:

If users run RSIGuard by loading RSIGuard off the server every time they start up, then only the copy of RSIGuard on the server needs to be registered with a registration code (and thus you don't need to use this command). However, if each user is installing RSIGuard on their personal machine, each copy of RSIGuard must be individually registered. This command allows the registration process to be automated. If a user is not yet registered, then this command will register their copy using REGISTRATIONCODE (assuming it is a valid registration code). If the registration code starts with 'x', then internet access is required to successfully complete the registration process.

Example:

register xacme1234

---

sendconfig / senddata – transmit configuration and DataLogger data to OES database

Notes:

These commands are only for use by the OES to request data from RSIGuard.

---

sendhsr – trigger the submission of a health status report

Syntax options:

sendhsr query/noquery/nosurvey

Notes:

The 'sendhsr' command causes RSIGuard to trigger the submission of a Status Report to a previously configured location. 'sendhsr noquery' causes a DataLogger summary and previously set survey responses to be submitted (even though survey responses may be old or may never have been set). 'sendhsr nosurvey' causes a DataLogger summary to be submitted (but not survey responses) – which is equivalent to a "DataLogger Status Report" (DSR). 'sendhsr query' triggers a user survey window to appear (including a discomfort survey) and submits the survey information, and if the user permits it, also sends a DataLogger summary. 'sendhsr query' is equivalent to a "Health Status Report" (HSR). The DataLogger summary submitted in each case is composed of averages of recent DataLogger data. The length of time over which the average is computed for 'sendhsr query' is in the HKEY_CURRENT_USER\Software\RSIGuard\Settings\$p\Reports\DaysInSample registry key. The length of time over which the average is computed for 'sendhsr noquery' and 'sendhsr nosurvey' is in the HKEY_CURRENT_USER\Software\RSIGuard\Settings\$p\Reports\DSRDaysInSample registry key.

---

set – remotely control RSIGuard settings

Syntax options:

set SETTING VALUE (set the value of the RSIGuard setting specified by SETTING to VALUE)

Notes:

This command lets you control user settings remotely. You specify the setting you wish to change with “SETTING” and what you want to set it to with “VALUE”. Each setting that appears in the RSIGuard Settings Dialog (and several others) can be specified with the SETTING argument. The key is to understand how SETTING is specified. Each RSIGuard setting has a location within the registry, and SETTING must correspond to that location.

All changeable settings are stored either in HKEY_LOCAL_MACHINE\Software\RSIGuard\Settings or HKEY_CURRENT_USER\Software\RSIGuard\Settings. The SETTING identifier begins with a m\ for the HKEY_LOCAL_MACHINE settings (i.e. m for machine-based settings) and u\ for the HKEY_CURRENT_USER settings (i.e. user-based settings). The rest of the string is the rest of the registry path to the setting. For example, the organization name is stored in HKEY_LOCAL_MACHINE\Software\RSIGuard\Settings\RSIGuard\OrganizationName, so the SETTING identifier is m\RSIGuard\OrganizationName. To change it, you could use:

```
set m\RSIGuard\OrganizationName ACME Inc.
```

Because most (but not all) HKEY_CURRENT_USER settings are specific to a particular profile, you must also specify the profile you wish to change. Often in a startup file, however, you don’t know the profile – in other words, you want multiple users to use the same startup script and have it affect their current profile. You can specify the current profile with the $p substitution (see pre-processing section). For example, to enable the BreakTimer for the current user/profile, you could use the command:

```
set u\$p\BreakTimer\Enabled 1
```

Note that for boolean settings (settings than can only be true or false), you use 1 for true and 0 for false.

You could also create a command that only changes the setting for a particular profile. For example:

```
set u\_JohnSmith\BreakTimer\Enabled 1
```

The 'set' command is powerful in that it lets you arbitrarily write into a user's profile. There is no checking that the setting you are trying to change is a valid setting or that you are not changing an inappropriate setting. For example, if you misspell something in SETTING, you will create a new (probably irrelevant) setting. If you were to arbitrarily change a setting like "the number of user profiles on this PC", you would likely damage the integrity of the user profiles on the PC. Please feel free to contact RSIGuard Software directly at (831)421-0139 if you would like assistance with your particular application.

Examples:

```
# specify where all users's usage data should be stored
set m\RSIGuard\DataFolder \\NETDRIVE\RSIGuard\Data

# specify where Health Status Reports should be filed
set m\RSIGuard\ReportFilingLocation *N:\\NETDRIVE\RSIGuard\HSR

# specify your organization's name
set m\RSIGuard\OrganizationName Acme, Inc.

# This changes the minimum time between breaks for the current profile to 15 minutes
set u\$p\BreakTimer\MinInterBreakTime 15
set u\$p\BreakTimer\MinInterBreakTimeEnabled 1

# This disables access to the ForgetMeNots settings from the user interface
set u\GeneralSetup\AccessFMN 0
```

setenv – set an environment variable

Syntax options:

setenv ENVIRONMENT-VARIABLE [VALUE]

Notes:

This command sets a system environment variable. The value of environment variables can be accessed with $env()

Example:

```
setenv RSI_DEPARTMENT_CODE 3
setenv RSI_DEPARTMENT $plus($env(RSI_DEPARTMENT_CODE),1) #increment variable
```

---

setvar – creates an alias for a string to be used elsewhere in script

Syntax options:

setvar 1-9 VALUE

Notes:

Defines 1 of 9 string variables that can be substituted in subsequent commands by a '$' followed by the variable number.

Example:

```
setvar 1 \\networkdrive\rsiguard
set m\RSIGuard\DataFolder $1\data
set m\RSIGuard\StartupFile $1\scripts\startup.txt
set m\RSIGuard\NewUserScript $1\scripts\newuser.txt
```

---

type – type specified text

Syntax options:

type TEXT

Notes:

Types specified text as if you typed it from the keyboard. The following special keypresses can be created: <CTRL>, <SHIFT> or <ALT> for modified keys (e.g. to do Ctrl-Alt Y, use <CTRL<ALTy>>. For Shift, just use the capital letter, (e.g. for Ctrl-Shift-Y use <CTRLY>). You can also use <TAB>, <ENTER>, <BACKSPACE>, <UP>, <DOWN>, <LEFT>, <RIGHT>, <ESC>, <PAGEUP>, <PAGEDOWN>, <HOME>, <END>, <INSERT>, <DELETE>, <SCROLLLOCK>, <PAUSE>, <WINKEY>, <RIGHTCLICKKEY>, <BREAK>,<LESSTHAN>, <GREATERTHAN>, <F1> through <F12>.

Example:

```
type This is a test<ENTER>
// these next two lines would start an email, fill in the fields and send it (using an already running Outlook session)
open "C:\Program Files\Outlook Express\MSIMN.EXE"
type <CTRLn>info@rsiguard.com<TAB><TAB><TAB>Hi<TAB>Joe,<ENTER>Hi.<ENTER><CTRL<ENTER>>
```

---

update – change folder for DataLogger or Profile data storage and copy data to new folder

Syntax options:

update datafolder/profilefolder NEWFOLDER

Notes:

Changes where RSIGuard stores either DataLogger data or settings profile data (i.e. roaming profile). This setting could be changed with the 'set' command, but this command also copies the current data to the new location, providing a mechanism for a seamless transfer of storage location (whereas the set command would just change where data was written in the future).

Example:

update datafolder S:\NetDrive\RSIGuard\Data
update profilefolder \\svr010\NetDrive\RSIGuard\Data

---

wait – wait specified time before continuing to run script

Syntax options:

wait SECONDS

Notes:

Pauses execution of the script file for the specified number of seconds (within a range of 0.1 to 60 seconds).

Example:

wait 5 (wait 5 seconds)
wait 1.5 (wait 1 and a half seconds)

---

window – control aspects of the main RSIGuard display window

Syntax options:

window hide/show/center
window alwaysontop/strainbars/autohidebuttons/showtimes 1/0

Notes:

Affect the display of the main window. "Window Hide" and "Window Show" cause the main window to either hide in the system tray or be visible on the screen. "Window Center" forces the window to be visible and appear at the center of the screen. "Window AlwaysOnTop" sets if the display window should always be on top of other windows. "Windows StrainBar" sets if the strain indicators should be shown in the main display. "Windows AutoHideButtons" sets if the main control buttons should automatically hide when the mouse is not over the RSIGuard window. "Windows ShowTimes" sets if the work times (e.g. time since last break, time to next break, etc.) should appear in the main window.

Examples:

window hide (hide the main RSIGuard window)
window strainbars 1 (show the strainbars)