Getting Started

When Macro Scheduler is started it places its icon in the system tray next to the clock. To invoke the main Macro Scheduler window double click on this icon. Alternatively it is possible to right click on the icon and then select 'Settings' from the pop up menu. If you are using NT3.5x which has no system tray, Macro Scheduler is simply minimised. Just maximise to access its functions.

This is the main control centre for Macro Scheduler. From here macros can be started, recorded, deleted and edited.

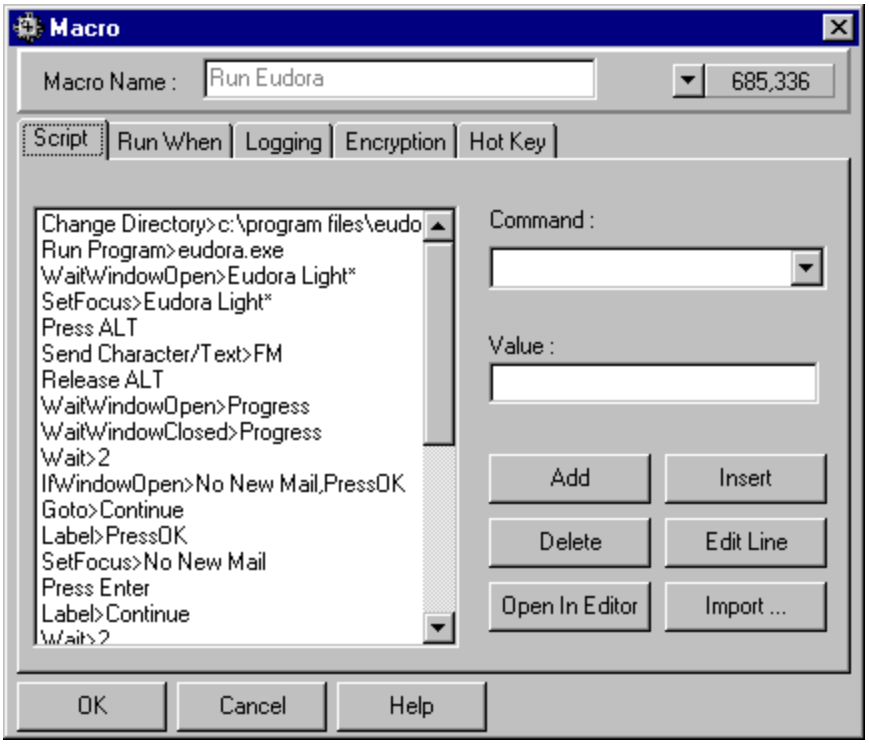
For further information see Creating Scripts and Recording Macros

Access to the functions is through the buttons or by double clicking on the appropriate macro, or right clicking on it to display a pop up menu.

Pressing OK when using Windows95 or NT4.0 hides the Macro Scheduler window and it continues to run in the system tray. On NT3.5x pressing OK will minimise Macro Scheduler.

Creating Scripts

To create a new script click 'New' on the main Macro Scheduler window. To Edit an existing script, select it from the list by clicking on it and then press 'Edit'. Having done this you will be presented with the following window:



In this example an existing script was selected to be edited. If 'New' had been pressed then this form would appear blank.

There are two ways of building scripts from this window. One way is to use the drop down list of commands and the other is to use the built in editor. The latter method is recommended only for more experienced users and is great for maintaining long scripts. See Using The Editor for more information.

To build your script using the drop down list, simply select the desired script command, enter a value if one is required and then click 'Add' to add it to the end of the script. To insert a line, highlight the line you want to insert above and click 'Insert' after choosing the right command. You can delete a line from the script by selecting it and clicking on 'Delete'.

To help you find the required command, you can sort the command list by right clicking on the drop down box and selected 'Sorted'.

If you choose a command that requires a parameter value, enter one in the box marked 'Value'. If you forget Macro Scheduler will tell you what you need to enter. For detailed help on a command, select it from the list and then press F1.

If you need to edit a line quickly, highlight the line and click 'Edit Line'. The value will then appear in the value box and the 'Edit Line' button will now say 'Update'. After editing the value click on 'Update' and the script will be updated.

The button marked 'Open In Editor...' opens the script in the editor. The Import button allows you to load

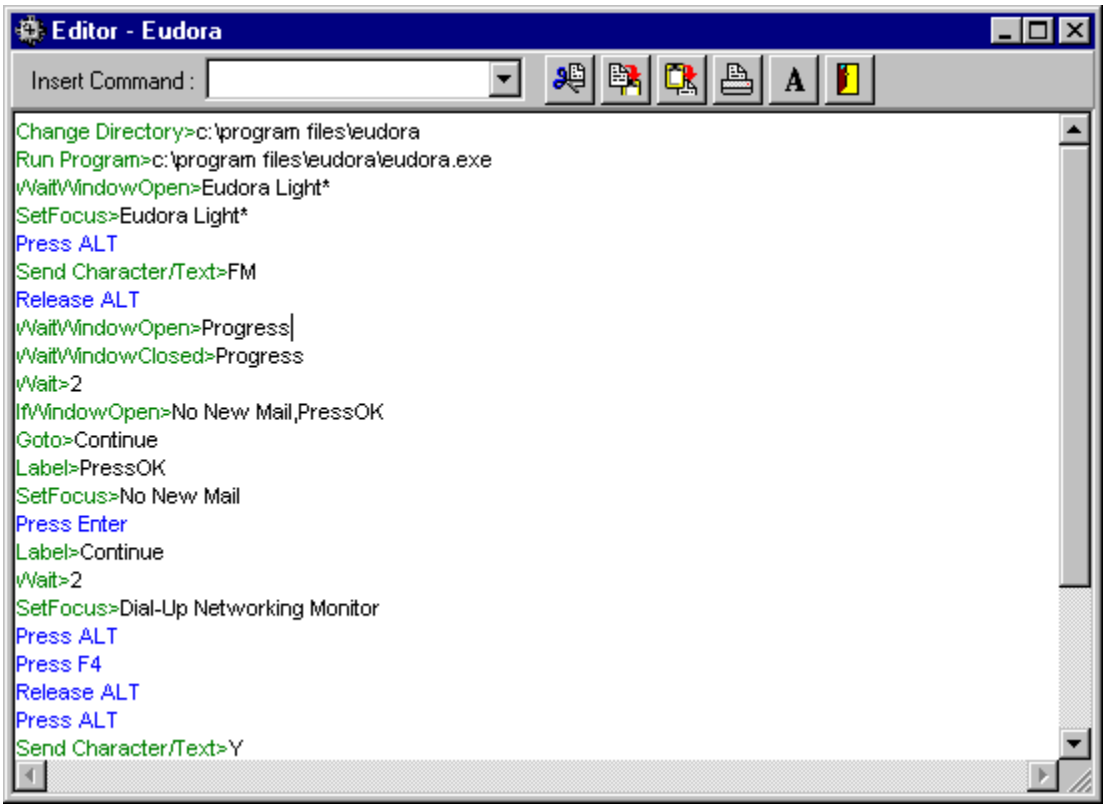
in a script that has already been created. This is useful if you have a number of Macro Scheduler installations and want to make use of a script created on a different PC for example.

The numbers at the top right of this window show your mouse cursor position. Use these to determine the correct parameters when using the MouseMove command, much easier and quicker than guessing !! The button to the left of the numbers will reveal a drop down menu. On here you can toggle between absolute coordinates and relative coordinates. When set to relative, the numbers show you the relative coordinates to the window the cursor moves over. The other option allows you to attach a small tag to the mouse cursor that shows the coordinates and follows the cursor around. This means that if the script window becomes concealed by another application you can still see the cursor position.

Once you have created your script press 'OK' to save it and return to the main window. Press 'Cancel' to leave without saving the changes.

Using The Editor

While the method mentioned in Creating Scripts is very easy to use and avoids any errors occurring, it can become a bit laborious when creating very long and complicated scripts, especially if you need to perform cut and paste operations with large or repeated chunks of code. This is where the editor comes in.



To open your script in the editor simply edit the script in the usual way and then click on the 'Open In Editor...' button on the script settings window.

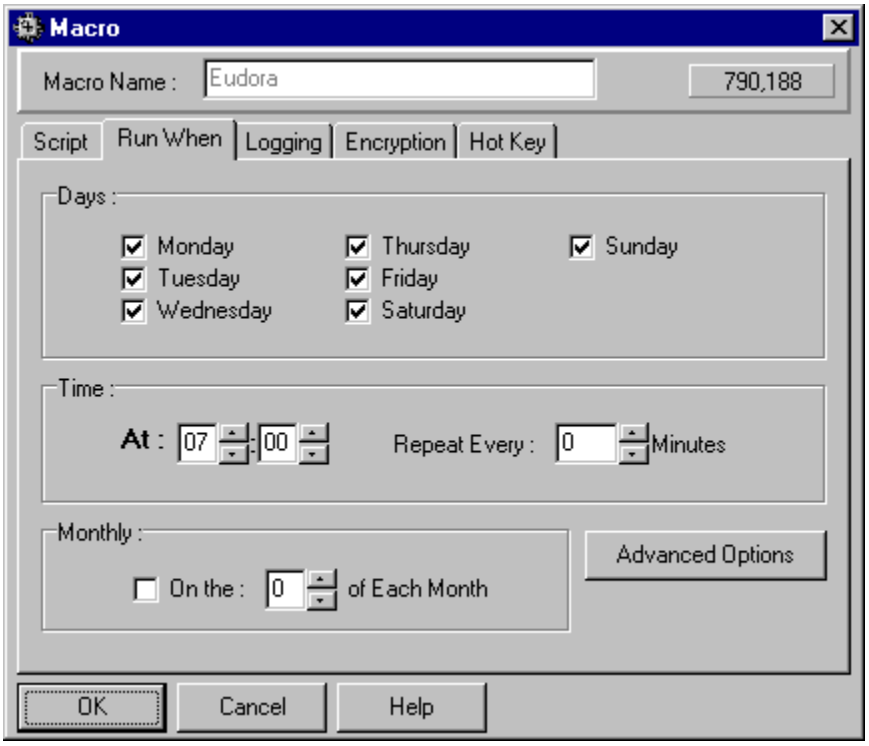
As a guide there is also a drop down command list on the panel which will write out your chosen command in the editor. This is useful for avoiding spelling mistakes and getting the case wrong. The usual editing functions are available from the speed buttons to cut, copy and paste. These operations can also be achieved using the standard windows shortcut keys (CTRL-X,CTRL-C,CTRL-V). The script can be printed by clicking on the printer button. If you're not sure what a button does just hover the mouse cursor over it to reveal the tool tip. When you have finished editing, close the editor by pressing on the door button or using the standard close button on the title bar. All changes made will now appear in the script box on the script settings window.

To change the default font name and size use the font button represented with a capital A.

Scheduling Scripts

Once you have created your macro you will probably want to execute it. Macros can be run at any time from the main window, from Windows shortcuts, from the command line or to a specified schedule.

To set up a schedule, select the appropriate script and choose to edit it to invoke the macro settings screen. Then select the tab marked 'Run When' to display the following options.



Mark off the days on which you want the macro to run and enter a time. The time must be entered in 24 hour notation. If you want the macro to be repeated enter an appropriate value in the 'repeat every' box. If you don't want it repeated simply leave this set to zero. The repeat every box allows you to specify upto 1440 minutes (a full day).

To make the macro run on a monthly basis check the monthly box and enter the day of the month on which you want the macro invoked.

If you choose to run a macro on the 5th of the month and also check the Friday box, the macro will run every Friday AND on the 5th of the month regardless of what day the 5th is.

In the 'Advanced' scheduling settings you can specify what should happen when Macro Scheduler restarts. This can be useful if you have Macro Scheduler start every time you reboot your computer. For instance, you can set Macro Scheduler to continue to repeat when the computer restarts. You can also determine what should happen if a schedule was missed while the computer was turned off.

If you have created a schedule for a macro, but wish to temporarily stop scheduling without changing the schedule details, you can do so from the pop up menu of the script list on the main Macro Scheduler window.

A Note About Screensavers

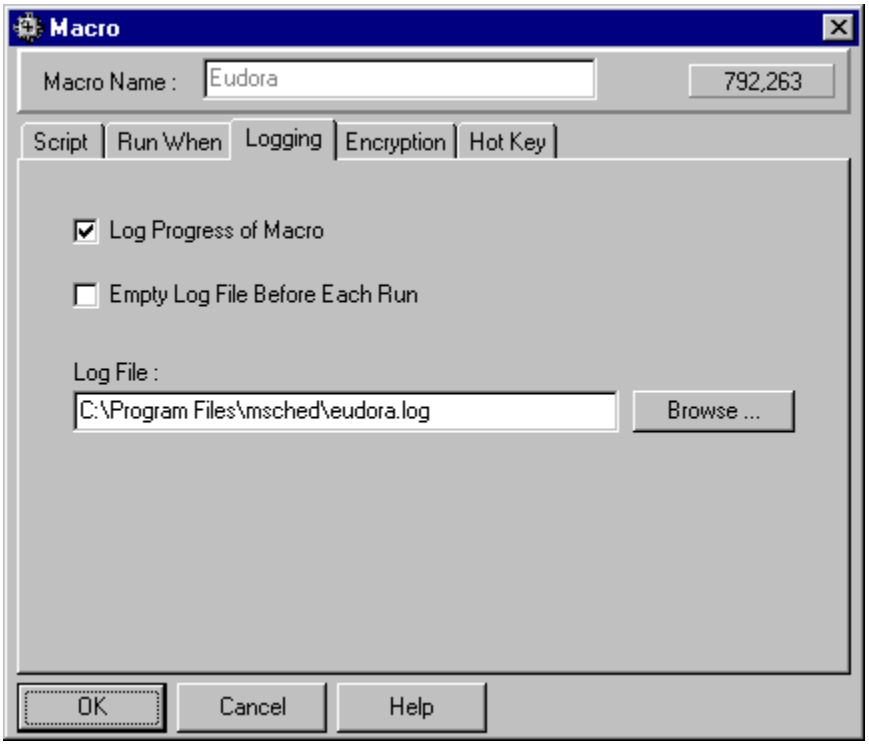
Screensavers usually stop successful detection of other windows. Consequently if a script that needs to setfocus or wait for windows to appear is run while a screensaver is active it may not work correctly.

To get round this Macro Scheduler temporarily disables screensaving just before it runs a script and re-enables screensaving when the script completes. It also attempts to determine if a screensaver is currently active and if so closes it down. However, there are many different implementations of screensavers which operate in different ways, making their detection and close down a rather unreliable process. To try to ensure that Macro Scheduler is successful in closing an active screensaver it briefly moves the mouse back and forth before running the script.

This should work in most cases. However, if you find that Macro Scheduler is unable to stop your screensaver, then the most reliable method of making sure that a scheduled macro runs properly is to simply disable screensaving altogether.

Logging

To set up a log file for a script select the tab marked 'Logging' on the macro settings window.



To enable logging check 'Log Progress of Macro'.

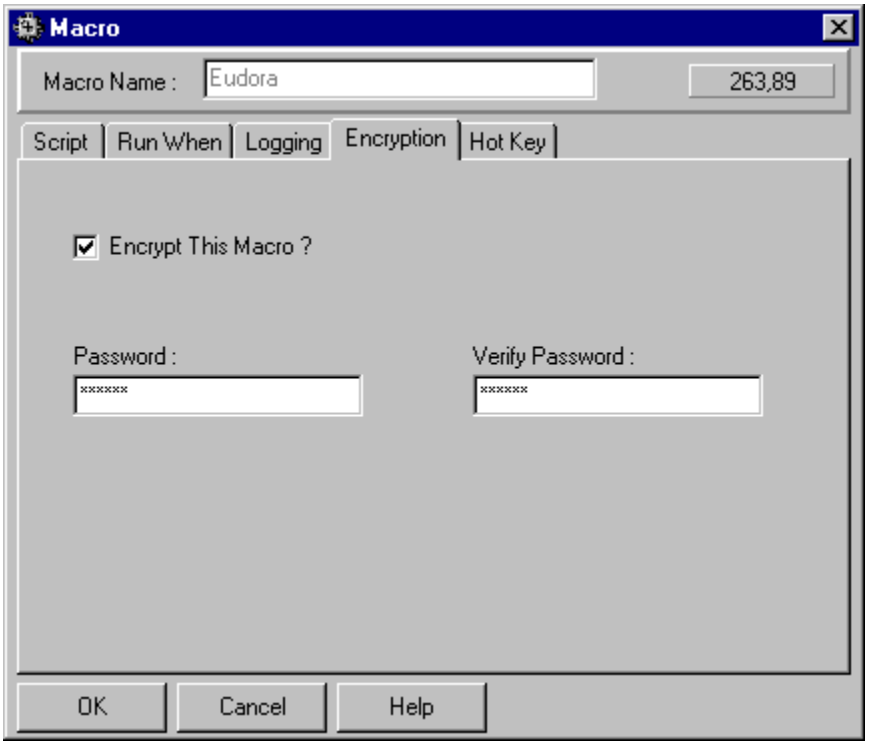
You can have Macro Scheduler purge the log file before each run by ticking the second box.

Enter a file name for the log file or select an existing one by using the browse button. If you like you can use one file for more than one macro.

Encryption

The majority of us probably won't ever have to use the encryption facility. However, if you need to use Macro Scheduler to automate a process which involves sending passwords to other applications or to send other sensitive information, then you would want to ensure that only the right people can edit the script and see the secrets.

Macro Scheduler allows you to set a password for a script which must then be used to edit it. The script file itself is scrambled so that if it is edited in any way it wouldn't make any sense.



Simply tick the box and provide a password.

The password must be entered twice to ensure it is entered correctly.

Next time you try to edit the macro you will be asked for the password.

To disable encryption on a macro that has been encrypted, edit it and then un check this box.

Recording Macros

To record a macro press the 'Record' button on the main window. You will be prompted for a name for the macro and recording will commence when Start is pressed.

By default CTRL-ALT-S will terminate the recording. You can select an alternative key from the drop down list box if required.

New in version 4.2 was the ability to have the recorded macro translated into an ordinary script which can then be edited in the usual way. It is possible to choose whether or not to translate the macro when recording is finished. If you choose not to have it translated at this stage you can do it later by attempting to edit the macro. If you select to have the macro translated you can also decide whether or not to remove adjacent mouse moves (see below).

Carry out the tasks you want to be captured and finally press CTRL-ALT-S (or chosen key) or click on 'Stop' to end the recording.

The new macro will appear in the macro list and can be executed by clicking the 'Run Now' button.

To schedule a recorded macro you need to create a new script and use the Macro> command to call the macro from the script. This script can then be scheduled in the usual way. See Script Commands. Alternatively if you translated the macro to a script you will be able to schedule it as usual.

Recorded macros consist of very low level system commands. Each key press and mouse click is treated separately. Consequently when you edit a translated macro you will find that each Send Character/Text> command has only one letter after it which represents the key that was pressed. Furthermore, where you might choose to use a LClick if you were writing a script manually, the program will use one LDown followed by one LUp command, possibly with a wait statement between the two. There will be many wait statements. This is necessary for the macro to reflect as accurately as possible what took place during the record. The only time you will find that the program has simplified things is when interpreting mouse moves.

If you chose to remove adjacent mouse moves, the only mouse move given will always be the last one before a different command. This speeds up playback and keeps the script file much shorter (Moving your mouse slowly across the screen can create hundreds of mouse move messages, but in a script only the last one is needed). However, there might be cases when it is preferable to leave all mousemoves intact, which is why the option is there to allow you not to remove adjacent mousemove commands if required.

Macros that are not translated may be played back more precisely than those translated, since each recorded Windows message can be replayed just as it was sent.

Playing Scripts and Macros

To play a macro or script without scheduling it use the 'Run Now' button on the main window. When running a regular script the 'Run Now' button will become a 'Stop' button. The script can then be cancelled during its execution.

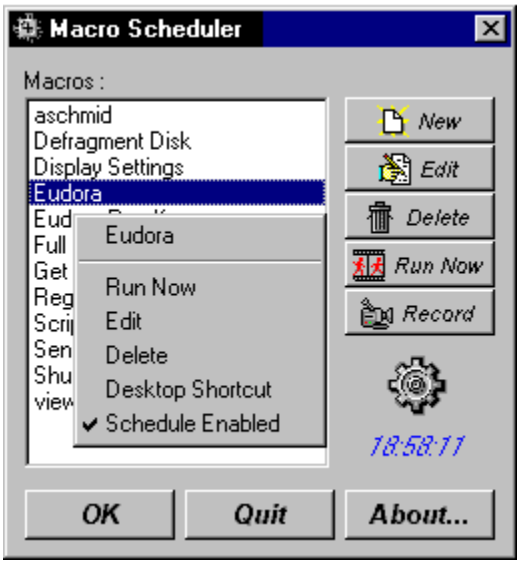
The menu that pops up from the icon in the task bar also has an option called 'Break'. This works like the 'Stop' button and allows you to cancel the execution of a script. This option is available even when a script is executed automatically by the scheduler.

You can run recorded macros within ordinary scripts by using the Macro> command. See Script Commands.

Macros can also be assigned to desktop shortcuts or run from the command line. See Creating Desktop Shortcuts and Command Line Option.

Creating Desktop Shortcuts

You can tell Macro Scheduler to create a shortcut for a macro by selecting the appropriate macro from the main window and then clicking the right mouse button to display a pop up menu.



Select the last option and a shortcut will be placed on your desktop. To run the macro you then only need to double click your desktop icon. Once it is on your desktop you can, if you prefer, move it elsewhere in the usual way using explorer etc.

Command Line Option

It is possible to run macros from the command line using the following syntax:

msched macroname

e.g. to run the Defragment Disk example script you would type:

msched Defragment Disk

This is useful for creating shortcuts and running Macro Scheduler scripts from other programs or from macros created in other applications such as Word or Excel.

However, if you want to create a shortcut, you can get Macro Scheduler do it for you. Click here for details.

When running a Macro from the command line in this way, you can also pass parameter values into the script :

msched Example Script /filename=testfile.txt /path=c:\outpath\

The above example runs a script called 'Example Script' and passes two variable values in filename and path. These variables can be used in the usual way within the script, e.g. :

```
Change Directory>path
ifFileExists>filename,ok
Goto>end
Label>ok
Message>File Exists !!
Label>end
```

License Agreement

Registration

Why Register ?

The License Agreement specifies that you can use Macro Scheduler freely for evaluation purposes only for 30 days. After that time you must register it.

As well as legalising your copy of Macro Scheduler, registering will bring you the following benefits :

- The reminder notice will no longer be displayed when starting macros.
- Technical Support.
- Free updates.
- Access to the Macro Scheduler email discussion list.

How much does it cost ?

The cost of registering one copy of Macro Scheduler is :

40 USD

Are there any multi-user licenses ?

Yes, the following table outlines the current licensing options available and their costs :

License	Price (USD)
5	150
10	260
25	600
50	1000
100	1400
500	4000
1000+	6000

How do I register ?

Registration can be carried out on-line at www.atlantic-coast.com

To register point your browser at :

http://www.atlantic-coast.com/cgi-bin/sellonline/m012cart.htm

After completing the registration form on line you will be emailed a username and a registration key. From the about box click on 'Register' and enter your username and registration key. The limitations will then be lifted and you can have as many macros as you like and much bigger scripts.

To pay by credit card you must contact SoftShop.

Alternatively (for non credit card orders) you can contact MJT Net Ltd by post, phone, fax or email as follows :

MJT Net Ltd
19 Dorchester Court
New Hartley
Tyne & Wear
NE25 0SS
UK

Phone : +44 (0) 976 691 276
Fax : +44 (0) 191 237 2998
Email : sales@mjtnet.demon.co.uk

We are happy to accept purchase orders by fax or post and send an invoice.

Support

If you are a registered user you can get support by emailing us at :

msched-sup@mjtnet.demon.co.uk

Please also send bug reports, comments and suggestions to this address.

For hints and tips have a look at the Scripts & Tips page at :

http://www.mjtnet.com/arch/scripts.hts

Keep an eye on the MJTNET Software web site for new product announcements and information:

http://www.mjtnet.com/

Script Commands

This page is included for historic reasons. For detailed information on each command please see the command reference.

The following table shows a list of commands which require a parameter or value to be entered :

Command	Description	Value Required
Run Program	Runs a program/command	Command line to run
Execute File	Executes a file using its association e.g. Specifying a .doc file will invoke Word & load the document.	Full path to file
Macro	Runs another macro or script	name of macro/script to run
Change Directory	Changes directory	Directory to change to
Wait	Waits given seconds	Seconds to wait
Send Character/Text	Sends given text	text to send
WaitWindowOpen	Waits for given window	title text of window*
WaitWindowClosed	Waits for window to close	title text of window*
Setfocus	Sets focus to specified window	title text of window*
Label	Inserts a label in the script for branching control to	A label name
Goto	Branch to the specified label	A label name
IfWindowOpen	Moves to specified label if given window is present	*Window title and label in the format: window title,label
Message	Displays a message	Message text to display
DeleteFile	Deletes a file	The file to delete
CopyFile	Copies a file	Source File and Destination in format: Source,Destination
MoveFile	Moves a file	Source File and Destination in format: Source,Destination
EditIniFile	Edits contents of an ini file	Filename,Section,Entry,New Value
ReadIniFile	Reads a parameter from an ini file into a variable	Filename,Section,Entry,Variable
MouseMove	Moves Mouse Cursor	Coordinates to move to : Xpos,Ypos e.g. MouseMove>550,254
MouseMoveRel	Moves Mouse Cursor Relative to active window	Coordinates to move to : Xpos,Ypos e.g. MouseMove>550,254
IfFileExists	Moves to specified label if given file exists	Filename and label : filename,label
IfFileChanged	Moves to specified label if given file date is in range of days specified.	Filename,[<,>,=]days old,label e.g. to see if test.txt is less than 30 days old: test.txt,<30,end
Let	Declares a variable and assigns it a value It is possible to use numeric and string values	Variable and Value : MyName=Freddy

		LoopCounter=1
Add	Adds a value to another The first parameter must be a variable	Add>A,5 is interpreted as A=A+5 Add>B,A is niterpreted as B=B+A
Sub	Subtracts a value from another The first parameter must be a variable	Sub>A,5 is interpreted as A=A-5 Sub>A,B is interpreted as A=A-B
ConCat	Concatinates two string values The first string must be a variable name	ConCat>A,B A now has B appended to it
If	Compares two values and branches to given label if result is true	If>5=5,SomeLabel If>B=A,SomeLabel If>MyName=Freddy,SomeLabel
CreateDir	Creates a directory	Path and new directory name
Input	Prompts for input	Variable and prompt : surname,Please Enter Your Surname
GetDate	Retrieves the current date in system short date format defined in regional settings	Variable to store date in
GetTime	Retrieves the current time in system time format defined in regional settings	Variable to store time in
MidStr	Retrieves a substring of another string	<string>,<start>,<length>,<result> e.g. MidStr>John Smith,1,4,FirstName FirstName would become 'John' string,start and length can be passed in variables.
Position	Returns the start position of one string in another.	<substring>,<string>,<start>,<result> e.g. Position>John,John Smith,1,pos pos would equal 1. All parameters can be passed in variables
GetClipBoard	Retrieves textual contents of clipboard	Variable to store text in.
PutClipBoard	Puts text onto clipboard.	Value to add to clipboard.
GetWindowPos	Retrieves upper left coordinates of specified window.	<window name>,<Xpos>,<Ypos>
FindWindowWithText	Locates window containing given text	<text_to_find>,<setfocus?>,<result variable>
Day	Returns the current day number	Variable to store result in
Month	Returns the current Month number	Variable to store result in
Year	Returns the current Year number	Variable to store result in
GetActiveWindow	Returns title and X,Y coords of active window	<title>,<Xpos>,<Ypos>
CloseWindow	Closes specified window/application	*Window Title

*Window title text no longer has to be an exact match. If the last character specified is an asterisk (*) Macro Scheduler will first attempt to setfocus, or find, the first window whose title matches the text entered exactly. If it cannot make an exact match it then looks at all windows and sets focus to the first one it finds whose title contains the entered text. This solves the problem with applications such as Word or Netscape which change their titles depending on the document loaded. It is best to try to provide an exact (including case) window title to ensure the correct window is found, as many applications have multiple invisible windows with similar names. Specifying text without a trailing asterisk will force Macro Scheduler to only look for an exact match.

The Wait command is quite important. Imagine starting your word processor. It takes a few seconds (sometimes longer!) between starting the program and it being ready for your input. The Wait command

can be used to make sure that Macro Scheduler does not start sending the next command before the receiving program is ready. The time you should specify really depends on your computer and what is running at the time. The only way of knowing exactly what to enter is by trial and error. Go through the motions manually before building the script and note roughly how long everything takes.

Of course, you could use the WaitWindowOpen command to wait for your program to appear before the script continues.

The list below contains the special key press and mouse commands which do not need a value.

Notice that for some keys there is a Release command. This is to cater for situations where these keys are used with other keys, particularly when using 'Hot Keys'.

For example to exit a program you would press ALT and F together to activate the File menu, followed by the X key to select the Exit option. To simulate this in a script you would Press ALT, then send the text FX and finally Release ALT. This would appear in the script window as :

Press ALT
Send Character/Text>FX
Release ALT

Be careful to make sure that if one of these keys is pressed you always release it again, otherwise you may find that everything you type after running the script is being interpreted as though you were holding one of these keys down while you type !! Pressing it again usually clears this anyway !!

Press Backspace
Press Tab
Press Enter
Press Esc
Press F1
Press F2
Press F3
Press F4
Press F5
Press F6
Press F7
Press F8
Press F9
Press F10
Press F11
Press F12
Press Home
Press End
Press Up
Press Down
Press Left
Press Right
Press Page Up
Press Page Down
Press Ins
Press Del
Press Shift
Release Shift
Press CTRL
Release CTRL
Press ALT

Release ALT
Press ALTGR
Release ALTGR
Press CAPS (has the effect of toggling caps lock.)
Press Num Lock
Press Scroll Lock
Press NP0 (0 on Number Pad)
Press NP1 (etc)
Press NP2
Press NP3
Press NP4
Press NP5
Press NP6
Press NP7
Press NP8
Press NP9
Press NP Add
Press NP Subtract
Press NP Multiply
Press NP Divide
Press NP Decimal
CapsOn (switches caps lock on. Has no effect if already on.)
CapsOff (switches caps lock off. Has no effect if already off.)
NumOn (switches Num lock on. Has no effect if already on.)
NumOff (switches Num lock off. Has no effect if already off.)
ScrollOn (switches Scroll lock on. Has no effect if already on.)
ScrollOff (switches Scroll lock off. Has no effect if already off.)
LClick
RClick
LDblClick
RDblClick
LUp
LDown
RUp
RDown

Hot Keys

Each script can be assigned a hot key to allow the script to be launched from a keyboard shortcut.

To assign a hot key select the tab marked 'HotKey' from the script settings window and select the appropriate keys from the drop down lists.

Press OK to save the settings. No matter what program you are working with, as long as Macro Scheduler is running, the chosen key combination will now launch your macro.

Add

Add>Value,Number

Adds Number to Value.

Interpreted as Value = Value + Number

Value must be a variable containing either a numeric or date value. Number can be either a literal number or a variable containing a numeric value.

For date values this function will add the number of days, represented in Number to the given date value.

Example

Let>Counter=5
Add>Counter,2

i.e. Counter=Counter+2

In this example the numeric variable, Counter, is given a new value of 7.

CapsOff

Switches caps lock off. If Caps lock is already off, no action is taken. If Caps lock is on, it is switched off.

CapsOn

Switches caps lock on. If Caps lock is already on, no action is taken. If Caps lock is off, it is switched on.

Change Directory

Change Directory>path

Changes the current directory to the directory specified in path. path can be a literal string or a variable.

Example

Change Directory>c:\program files\my directory\

ConCat

ConCat>string1,string2

Concatinates string1 with string2. String1 must be a variable containing a string. String2 can be a literal string or a variable. The result is that string1 has string2 appended to it.

Example

```
Let>path=c:\temp\
ConCat>path,myfile.txt
```

In this example path becomes 'c:\temp\myfile.txt'

CopyFile

CopyFile>sourcefile,destinationfile

Copies the file (or files), sourcefile, to destinationfile

sourcefile, and destinationfile may be variables. Wildcards can be used.

Example

CopyFile>c:\temp\myfile.txt,c:\my documents\myfile.old

Or with variables:

```
Let>filename=c:\temp\myfile.txt
Let>newfilename=c:\temp\myfile.new
CopyFile>filename,newfilename
```

CreateDir

CreateDir>directory

Creates a new directory. The directory may be a full path.

Example

CreateDir>c:\my documents\test

or

Let>dir=c:\my documents\test
CreateDir>dir

EditIniFile

EditIniFile>inifile,section,entry,newvalue

Edits a section entry in an ini file. inifile can be a full path.

All parameters can be variables containing strings.

Example

EditIniFile>c:\program files\myini.ini,settings,user,fred

ExecuteFile

ExecuteFile>file_to_execute

Executes a file using the application associated with the given file's filetype.

file_to_execute can include a full path.

Example

ExecuteFile>report.doc

or

Let>filename=c:\my documents\accounts.xls
ExecuteFile>filename

GetDate

GetDate>result

Returns the current date in the specified variable. The format of the date depends on the regional settings of the system.

Example

```
GetDate>date
Let>msg=The Date Is :
ConCat>msg,date
Message>msg
```

GetTime

GetTime>result

Returns the current time in the specified variable. The format of the time depends on the regional settings of the system.

Example

```
GetTime>time
Let>msg=The Time Is :
ConCat>msg,time
Message>msg
```

Goto

Goto>Label_Name

Causes execution to continue at the specified label, missing any commands in between. If the label does not exist an error message will be displayed.

Goto in conjunction with Label, can be used to create infinite loops. Use the If.. commands to cause conditional branching. To break out of infinite loops press Stop, or choose the Break option from the taskbar pop up menu.

Example

```
Label>Start
..
..
Goto>SecondBit
..
..
Label>SecondBit
..
```

If

If>expression,label_name

Causes execution to continue at the specified label, if expression is true, missing any commands in between. If the label does not exist an error message will be displayed.

The following operators can be used in the conditional statement :

=	Equals
>	Greater than
<	Less than

Values in the expression can be numeric or string values, or variables containing such values.

Example

```
Label>Start
..
..
..
If>a<b,Start
```

IfFileChanged

IfFileChanged>filename,range,label_name

Causes execution to jump to the specified label if the given file's date is in the range of days specified.

Example

To see if test.txt is less than 30 days old :

```
IfFileChanged>test.txt,<30,end
..
..
Label>end
```

IfFileExists

IfFileExists>filename,label_name

Branches to the specified label if the file specified in filename exists.

Example

```
IfFileExists>myfile.txt,end
..
..
Label>end
```

IfWindowOpen

IfWindowOpen>window_title,label_name

Checks to see if the specified window is open. If so, it causes the script to continue from the specified label without running any other lines of code in between.

The window_title may contain the * symbol at the end to indicate a wildcard.

If the last character of the window title specified is an asterisk (*), Macro Scheduler will first attempt to find, the first window whose title matches the text entered exactly. If it cannot make an exact match it then looks at all windows and sets focus to the first one it finds whose title contains the entered text. This solves the problem with applications such as Word or Netscape which change their titles depending on the document loaded. It is best to try to provide an exact (including case) window title to ensure the correct window is found, as many applications have multiple invisible windows with similar names. Specifying text without a trailing asterisk will force Macro Scheduler to only look for an exact match.

Example

```
IfWindowOpen>Notepad - [Untitled],donotepad
..
..
Label>donotepad
```

or, with a wildcard :

```
IfWindowOpen>notepad*,donotepad
..
..
Label>donotepad
```

Input

Input>variable,prompt

Displays a dialog box to request information from the user. The dialog box displays the prompt specifies in prompt and accepts input into variable.

Example

Input>name,Please enter your name ...

Label

Label>Label_Name

Marks a point in the script to allow execution to be passed to that point by the Goto, and If.. commands.

Goto in conjunction with Label, can be used to create infinite loops. Use the If.. commands to cause conditional branching. To break out of infinite loops press Stop, or choose the Break option from the taskbar pop up menu.

Example

```
Label>Start
..
..
Goto>SecondBit
..
..
Label>SecondBit
..
```

LClick

Simulates a left button mouse click at the current point on the screen.

The following commands will produce the same result :

LDown
LUp

LDblClick

Simulates a left mouse button double click at the current point on the screen.

The following will achieve the same result :

LClick
LClick

or

LDown
LUp
LDown
LUp

LDown

Simulates a press of the left mouse button. This is like pressing the mouse button down but not releasing it. It is half of a click.

Issuing this command and then using MouseMove would implement dragging. Use LUp to complete the operation.

Let

Let>variable_name=value

Let is used to assign a value to a variable.

Where parameters are passed to commands, variables can also be passed. Variables can also be embedded within a parameter by enclosing the variable name within % symbols.

Examples

Let>name=freddy
Let>a=5

Let>path=c:\Program Files\
Run Program>%path%myapp.exe

LUp

Releases the left mouse button. It is the latter half of a click.

See LDown.

Macro

Macro>macro_name [/variable=value|variable [/variable=value|variable] ...]

Executes another macro. The specified macro must exist in the current installation of Macro Scheduler. It must be a valid macro that appears in the macro list on Macro Scheduler's main form.

To pass values to the macro specify each one after a / character. The variable name given should exist in the script to be run. The value to assign to that variable is specified after the = character.

Examples

Macro>Defragment Disk

Macro>MyMoveFile /source=c:\temp\myfile.bat /destination=c:\temp\myfile.bak

Message

Message>message_text

Displays a message box containing the text specified in message_text. In order that execution of the script can continue, message boxes are not modal. This means they can be used to display information even if the script is not being run interactively.

Example

Message>Hello World!

or with variables ..

Let>mymsg=Hello World!
Message>mymsg

MidStr

MidStr>string,start,length,result

Returns a substring of specified length from a given position in a string. result is a variable in which to store the returned string. Any parameter can be a variable containing the appropriate values.

Example

In the following example, the variable somevalue becomes equal to 'Happy' :

```
MidStr>Happy Birthday,1,5,somevalue
Message>somevalue
```

MouseMove

MouseMove>X,Y

Moves the mouse cursor to screen position X,Y. 0,0 is the upper left hand corner of the screen. The maximum limits are determined by your screen resolution settings. Variables containing the coordinates can be used in the command.

To help determine a particular point on the screen, the macro window has a cursor monitor which updates as you move the cursor. See Creating Scripts.

Example

If position 504,252 is within the area taken up by a button, the following script would cause that button to be clicked :

```
MouseMove>504,252
LClick
```

MoveFile

MoveFile>sourcefile,destinationfile

Moves the file (or files), sourcefile, to the file (or files) destinationfile.

sourcefile, and destination_path may be variables. Wildcards can be used.

Example

MoveFile>c:\temp\myfile.txt,c:\temp\myfile.bak

Or with variables:

```
Let>filename=c:\temp\myfile.txt
Let>newfilename=c:\temp\myfile.bak
MoveFile>filename,newfilename
```

NumOff

Switches Num lock off. If Num lock is already off, no action is taken. If Num lock is on, it is switched off.

NumOn

Switches Num lock on. If Num lock is already on, no action is taken. If Num lock is off, it is switched on.

Position

Position>substring,string,start,result

Returns the starting position of a substring in a string. The search commences at the position specified in start. If found the starting position of the substring is returned in the result variable. If no match is found this value will be zero.

Example

In this example, StartPos will contain the value 4 :

Position>Smith,Mr Smith,1,StartPos

Press ...

All commands starting with Press, facilitate the sending of non-character keys. The following is a complete list of all commands, with explanations where necessary :

Press Backspace	
Press Tab	
Press Enter	
Press Esc	
Press F1	
Press F2	
Press F3	
Press F4	
Press F5	
Press F6	
Press F7	
Press F8	
Press F9	
Press F10	
Press F11	
Press F12	
Press Home	
Press End	
Press Up	
Press Down	
Press Left	
Press Right	
Press Page Up	
Press Page Down	
Press Ins	
Press Del	
Press Shift	
Release Shift	
Press CTRL	
Release CTRL	
Press ALT	
Release ALT	
Press ALTGR	
Release ALTGR	
Press CAPS	has the effect of toggling caps lock.
Press Num Lock	
Press Scroll Lock	
Press NP0	0 on Number Pad
Press NP1	etc
Press NP2	
Press NP3	
Press NP4	
Press NP5	
Press NP6	
Press NP7	
Press NP8	
Press NP9	
Press NP Add	Num pad operator keys
Press NP Subtract	etc
Press NP Multiply	

Press NP Divide
Press NP Decimal
Press LWinKey }
Press RWinKey } Windows 95 Keys
Press MenuKey }

RClick

Simulates a right button mouse click at the current point on the screen.

The following commands will produce the same result :

RDown
RUp

RDblClick

Simulates a right mouse button double click at the current point on the screen.

The following will achieve the same result :

RClick
RClick

or

RDown
RLUp
RDown
RUp

RDown

Simulates a press of the right mouse button. This is like pressing the mouse button down but not releasing it. It is half of a click.

ReadIniFile

ReadIniFile>inifile,section,entry,result

Reads a value from an entry in an ini file and places the result in a variable.

All parameters can be variables containing strings.

Example

This example reads a username from an ini file and displays it in a message :

```
ReadIniFile>c:\program files\myini.ini,settings,user,username
Let>msg=The Username is
ConCat>msg,username
Message>msg
```

Release ...

Some keys that can be pressed must also be released. This facilitates holding down a key while another is pressed, such as with the ALT key for instance.

For example, to exit a program you would press ALT and F together to activate the File menu, followed by the X key to select the Exit option. To simulate this in a script you would Press ALT, then send the text FX and finally Release ALT. This would appear in the script window as :

Press ALT
Send Character/Text>FX
Release ALT

The following release key commands exist :

Release ALT
Release ALTGR
Release CTRL
Release Shift

Remark

Remark>Some Comment

The remark statement is ignored by the interpreter. It exists simply to allow comments to be placed in the code. In fact, any text that is not a recognised command can be used for this purpose.

Run Program

Run Program>path

Executes a specified file. Files than can be executed are .exe, .bat, and .com files.

Example

To open Notepad :

Run Program>notepad.exe

A path may be specified if necessary :

Run Program>c:\my programs\eudora\eudora.exe

RUp

Releases the right mouse button. It is the latter half of a click.

See RDown.

ScrollOff

Switches Scroll lock off. If Scroll lock is already off, no action is taken. If Scroll lock is on, it is switched off.

ScrollOn

Switches Scroll lock on. If Scroll lock is already on, no action is taken. If Scroll lock is off, it is switched on.

Send Character/Text

Send Character/Text>text_to_send

This command sends the specified text to the window that currently has the focus. See SetFocus.

Examples

Send Character/Text>Here Is Some Text ...

The following example simulates pressing ALT-FX, a standard key combination for closing a program :

Press ALT
Send Character/Text>fx
Release ALT

Variables can be used :

Let>SomeText=Hello World
Send Character/Text>SomeText

SetFocus

SetFocus>window_title

Sets focus to the specified window. The window_title may contain the * symbol at the end to indicate a wildcard.

If the last character of the window title specified is an asterisk (*), Macro Scheduler will attempt to setfocus to the first window whose title matches the text entered exactly. If it cannot make an exact match it then looks at all windows and sets focus to the first one it finds whose title contains the entered text. This solves the problem with applications such as Word or Netscape which change their titles depending on the document loaded. It is best to try to provide an exact (including case) window title to ensure the correct window is found, as many applications have multiple invisible windows with similar names. Specifying text without a trailing asterisk will force Macro Scheduler to only look for an exact match.

Example

SetFocus>notepad*

Sub

Sub>Value,Number

Subtracts a number from a value.

Interpreted as Value = Value - Number

Value must be a variable containing a numeric or date value. Number can be either a literal number or a variable containing a numeric value.

For date values this function will subtract the number of days, represented in Number from the given date value.

Example

Let>Counter=5
Sub>Counter,2

i.e. Counter=Counter-2

In this example the numeric variable, Counter, is given a new value of 3.

Wait

Wait>seconds_to_wait

This command makes Macro Scheduler pause for the specified number of seconds.

Example

To wait 5 seconds :

Wait>5

This will work too :

Let>WaitTime=5
Wait>WaitTime

WaitWindowClosed

WaitWindowClosed>window_title

Waits for a specified window to close. Execution of the script will not continue until the window with the specified title text is no longer present. The window title may contain the * symbol at the end to indicate a wildcard.

If the last character of the window title specified is an asterisk (*), Macro Scheduler will first attempt to setfocus, or find, the first window whose title matches the text entered exactly. If it cannot make an exact match it then looks at all windows and sets focus to the first one it finds whose title contains the entered text. This solves the problem with applications such as Word or Netscape which change their titles depending on the document loaded. It is best to try to provide an exact (including case) window title to ensure the correct window is found, as many applications have multiple invisible windows with similar names. Specifying text without a trailing asterisk will force Macro Scheduler to only look for an exact match.

The system variable WW_TIMEOUT can be used to set the number of seconds after which this command should timeout. If set to zero (the default) the timeout will not occur and the command will continue indefinitely. If WW_TIMEOUT is used, WW_RESULT will indicate whether or not the command ended succesfuly. If it timed out WW_RESULT will be set to FALSE. If the window it was waiting for appeared within the timeout setting, the WW_RESULT value will be set to TRUE.

Examples

```
WaitWindowClosed>Progress
```

```
Let>WW_TIMEOUT=10
WaitWindowClosed>Progress
If>WW_RESULT=FALSE,Endit
..
..
Label>Endit
```

WaitWindowOpen

WaitWindowOpen>window_title

Waits for a specified window to open/appear. Execution of the script will not continue until a window with the specified title text appears. The window title may contain the * symbol at the end to indicate a wildcard.

If the last character of the window title specified is an asterisk (*), Macro Scheduler will first attempt to setfocus, or find, the first window whose title matches the text entered exactly. If it cannot make an exact match it then looks at all windows and sets focus to the first one it finds whose title contains the entered text. This solves the problem with applications such as Word or Netscape which change their titles depending on the document loaded. It is best to try to provide an exact (including case) window title to ensure the correct window is found, as many applications have multiple invisible windows with similar names. Specifying text without a trailing asterisk will force Macro Scheduler to only look for an exact match.

The system variable WW_TIMEOUT can be used to set the number of seconds after which this command should timeout. If set to zero (the default) the timeout will not occur and the command will continue indefinitely. If WW_TIMEOUT is used, WW_RESULT will indicate whether or not the command ended succesfuly. If it timed out WW_RESULT will be set to FALSE. If the window it was waiting for appeared within the timeout setting, the WW_RESULT value will be set to TRUE.

Examples

```
Run Program>c:\program files\msoffice\winword.exe
WaitWindowOpen>microsoft word*


Let>WW_TIMEOUT=30
WaitWindowOpen>microsoft word*
If>WW_RESULT=FALSE,Endit
..
..
Label>Endit
Message>Error starting Word!
```

MouseMoveRel

MouseMoveRel>X,Y

Moves the mouse cursor to the position X,Y relative to the upper left corner of the window currently in focus. 0,0 will be the upper left hand corner of the active window. Variables containing the coordinates can be used in the command.

The advantage of this command over the MouseMove command, is that this will not fail when the window changes its position or resizes, or if the screen resolution is changed.

To help determine a particular point on the screen, the macro window has a cursor monitor which updates as you move the cursor. See Creating Scripts. To determine a point relative to a specific window, try moving that window so that its upper left corner is in the upper left corner of the screen (position 0,0). Maximising the app would achieve the same result. Then you can use the values displayed in the cursor position monitor of Macro Scheduler.

Example

If position 40,50 is a point relative to the current window, on a button, the following script would cause that button to be clicked :

```
MouseMove>40,50
LClick
```

DeleteFile

DeleteFile>filename

Deletes the file/files in filename.

filename can be a variable. Wildcards can be used.

Example

DeleteFile>c:\temp*.*

GetClipBoard

GetClipBoard>result_variable

Retrieves the contents of the clipboard as text and places it in the specified variable.

Example

GetClipBoard>WhatsInTheClipBoard

PutClipBoard

PutClipBoard>SomeValue

Places the specified text onto the clipboard. A variable may be used, or a literal value.

Example

PutClipBoard>Hello Wold !!

GetWindowPos

GetWindowPos>window_title,X,Y

Locates the window specified in window_title and retrieves its upper left screen coordinates. X and Y are variables in which to store the coordinates. window_title may be a variable or literal.

The window_title may contain the * symbol at the end to indicate a wildcard.

If the last character of the window title specified is an asterisk (*), Macro Scheduler will attempt to setfocus to the first window whose title matches the text entered exactly. If it cannot make an exact match it then looks at all windows and sets focus to the first one it finds whose title contains the entered text. This solves the problem with applications such as Word or Netscape which change their titles depending on the document loaded. It is best to try to provide an exact (including case) window title to ensure the correct window is found, as many applications have multiple invisible windows with similar names. Specifying text without a trailing asterisk will force Macro Scheduler to only look for an exact match.

Example

GetWindowPos>My Computer,X,Y

The following example achieves the same result as the MouseMoveRel command, moving to the point 10,10 relative to Notepad :

```
GetWindowPos>notepad*,npX,npY
Add>npX,10
Add>npY,10
MouseMove>npX,npY
```

FindWindowWithText

FindWindowWithText>text_to_find,setfocus_flag,result_variable,copy_edits_flag

This function attempts to locate a window containing somewhere within it the text specified in text_to_find. setfocus_flag can be set to 1 or 0. If set to 1, the function will activate the window it finds containing the specified text. If a window is found, result_variable will contain the found window's title text. If no window is found it will contain "NOT FOUND".

The copy_edits_flag can be set to 1 or 0. When set to 1, Macro Scheduler will attempt to temporarily copy text in any edit boxes it finds into the clipboard and see if text_to_find is contained within that text. The clipboard will be reset afterwards.

NB. Not all text can be detected successfully. Text on buttons, labels belonging to other controls, window titles, and pre-set text in edit boxes can usually be detected successfully. It may not be possible to find stand-alone label text or text entered by the user. Also, text in list boxes and combo boxes etc cannot be detected. To find text entered by the user it may be necessary to set the copy_edits_flag on.

This function is useful when an application has two windows with the same name, and allows the correct one to be located and focused.

Example

FindWindowWithText>Continue,1,windowname,0

Day

Day>result

Returns the current day number of the month in the specified variable.

Example

```
Day>the_day
Month>the_month
Year>the_year
Message>The date is : %the_day% - %the_month% - %the_year%
```

Year

Year>result

Returns the current year in the specified variable.

Example

```
Day>the_day
Month>the_month
Year>the_year
Message>The date is : %the_day% - %the_month% - %the_year%
```

Month

Month>result

Returns the current month number in the specified variable.

Example

```
Day>the_day
Month>the_month
Year>the_year
Message>The date is : %the_day% - %the_month% - %the_year%
```

CloseWindow

CloseWindow>window_title

Closes the specified window. The window_title may contain the * symbol at the end to indicate a wildcard.

If the specified window is the main window of an application, then that application will begin to close down. Any processing that the application does on exit will be carried out as usual.

If the last character of the window title specified is an asterisk (*), Macro Scheduler will attempt to setfocus to the first window whose title matches the text entered exactly. If it cannot make an exact match it then looks at all windows and sets focus to the first one it finds whose title contains the entered text. This solves the problem with applications such as Word or Netscape which change their titles depending on the document loaded. It is best to try to provide an exact (including case) window title to ensure the correct window is found, as many applications have multiple invisible windows with similar names. Specifying text without a trailing asterisk will force Macro Scheduler to only look for an exact match.

Example

CloseWindow>notepad*

GetActiveWindow

GetActiveWindow>window_title,X,Y

Retrieves information about the current active window. The window title, and top left coordinates of the active window are stored in window_title, X and Y.