Macro Express Pro
Manual of Instruction

Table of Contents

Introduction

Welcome to Macro Express Pro

Macro Express is the premier macro utility on the market. Use Macro Express to record, edit and play back mouse and keyboard macros. Its powerful tools and robust features will noticeably boost productivity.

What is a Macro?

A macro is a set of written or recorded commands that can be played back at will to perform a specific task. These tasks can be as simple as inserting a name and address into a word processor to something more complex such as launching a program, copying data from it, activating another program, pasting the data into it and repeating the steps several times.

Tasks performed by macros are typically repetitive in nature allowing significant savings in time by executing the macro instead of manually repeating the commands.

Creating a Macro

There are several methods available to create macros with Macro Express. The simplest is to capture (record) keystrokes and mouse movements. Another method is to use the Quick Wizards to build a macro through a series of questions and prompts. Or build more complex macros with the easy to use Script Editor or Direct Editor. The tutorial provides examples of creating macros using each of these methods.

Playing Back a Macro

Macro Express provides a number of methods for executing macros. These include the use of Hot Keys, Shortkeys, Scheduled Macros, Window Activation, Mouse Clicks, Popup/Floating Menus and others described below.

1. A Hot Key is a set of keys on the keyboard that are pressed to initiate the macro. This could include pressing CTRL+ALT+R or CTRL+F12 or any number of combinations.

2. A Shortkey is a small string of characters. When a matching combination is typed on the keyboard, it is erased and the assigned macro executes. For example, type /add to insert an address into another application.

3. A Scheduled Macro executes on a given time schedule.

4. A Window Activated macro runs when a specific window gains focus (appears) on the computer screen.

5. Start a macro by clicking on a specific area of the screen with the left, right or middle Mouse button.

6. A Popup Menu or Floating Menu is basically a list of defined macros. Once the list is displayed, click on the desired macro or press an associated number or letter to execute the macro.

7. Activate a macro through a Batch File or Command Line.

8. Execute macros by using Windows API calls or Windows Explorer.

9. Run a macro from the Macro Explorer by using the Run Macro Now option.

10. Start a macro when a particular word or phrase is copied to the Clipboard.

11. Invoke a macro when the information in a specified File or Folder changes.

12. Activate macros when a Process starts or stops running or when the TCP/IP stack becomes available or unavailable.

13. Trigger macros from a System Event, such as a Windows message, screen resolution or color change.

Macro Scopes

Macro scopes may be used to place restrictions on macros. Assign macros to play back in all Windows programs, in one specific window or program, or in every program except the one selected. The tutorial section also provides a few examples of how to do this.

Quick Start Guide

Macro Express Pro is a utility to help you automate repetitive computer tasks. In the process it will save you time and money.

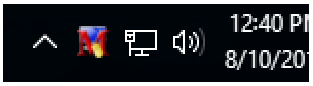
This guide covers a few basic concepts to help you better understand Macro Express.

Starting Macro Express

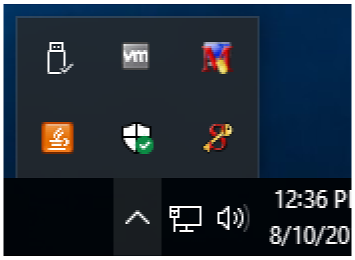
The default installation of Macro Express places a shortcut in the Windows Startup folder. This action causes Macro Express to load every time Windows starts. To open the Macro Express window do one of the following:

1. Press the system hot key of Ctrl + Alt + Shift + W

2. Left click on the Macro Express icon in the Windows notification area.



3. If the icon is hidden, left click on the triangle button to show the hidden icons. Left click on the Macro Express icon.



Player and Editor

Macro Express is comprised of two main components, the Macro Express Player and the Macro Express Editor.

Macro Express Player

The Macro Express Player is what executes macros and it must be running in order for macros to play back. When you run Macro Express, you are really running the Player.

By default, the Macro Express Player shows up as an icon in the notification area. From this icon you can invoke the Macro Express Editor or shut down the Macro Express Player.

You can optionally set the Macro Express Player to be completely hidden so that no icon shows in the notification area. With this option set, you would need to use the "System Macros" to invoke the editor or terminate the program.

Macro Express Editor

The Macro Express Editor is the tool used to create and edit macros. The Editor cannot be run by itself. The Player needs to be running in order to open the Editor.

Macros

A macro is a script containing commands that automate computer tasks. Each command performs one step, such as moving the mouse, launching a program or entering keystrokes.

There are several methods available for creating macros. No programming language or special skills are required or necessary.

1. Over two dozen Quick Wizards guide you through a series of questions and build the macro for you.
2. Record keystrokes and mouse movements with the easy to use Capture process.
3. Utilize the Script Editor or Direct Editor to write more powerful macros.

Macro Activation

Macro Express provides a number of options for executing (running) a macro.

1. Press a hotkey such as Ctrl + R or Alt + F11. Choose from over 840 combinations.
2. Type a shortkey - a short letter combination from 2 to 32 characters.
3. Run the macro on a schedule - hourly, daily, weekly, monthly or practically any other schedule.
4. Run automatically when a window opens or closes.
5. Click on a specific area of a screen or window.
6. Use popup or floating menus
7. Run a macro when a specific directory has been modified
8. Initiate a macro based on the contents of the clipboard
9. Plus others ...

Files

Macros shown in the Macro Express Pro - Explorer window are saved to a .mex macro file. The Explorer window displays the name of the .mex file in use and its location.

When the program is first installed, two .mex files are created - macex.mex and samples.mex.

macex.mex

On initial installation, this file does not contain any macros. It is suggested that you start with this file to write and save new macros.

samples.mex

The samples.mex file contains a number of sample macros that you may use for reference. It is recommended that you do not use this file as the default to create and store macros.

Additional .mex macro files may also be created for use. Open the Macro Express Pro - Explorer window and click on File and New File to create a new file. Or click on File and Open File to load in a file that you previously created and subsequently closed.

Converting from prior versions of Macro Express

Some preferences are not transferred from prior versions of Macro Express. You will need to configure Macro Express Pro by clicking Options and Preferences in the Macro Express Editor. Read the help topic "Upgrading from v3.x" in the Macro Express help for more information.

Tutorials

Written Tutorials and Video tutorials walk you through the process of creating and running macros.

Context Sensitive Help

Macro Express contains extensive help. Access the appropriate help topic by clicking the Help menu or by pressing F1 in any open Macro Express window.

Program Navigation

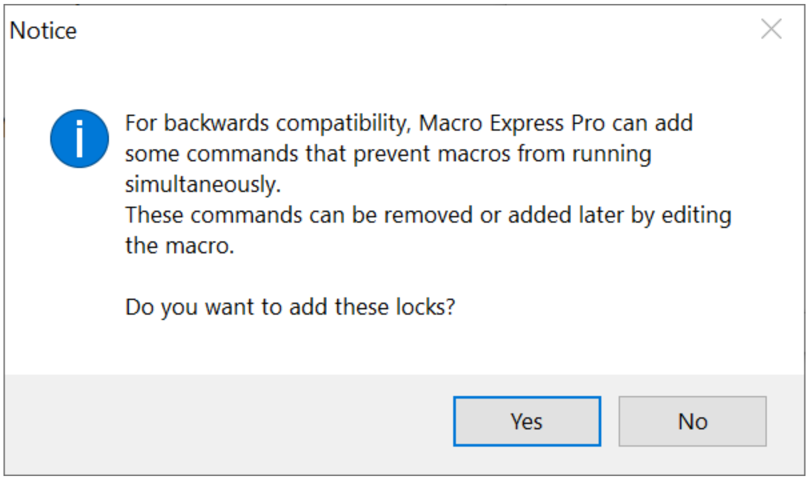
Macro Express Pro consists of several components which may be accessed by clicking on the icons or accessing through the menu commands. Each component is briefly discussed below.

	Macro Explorer	By default, the program opens to the Macro Express Pro - Explorer. This displays the list of macros in the file currently open. When saving macros in the editors or other options, the program returns to the Macro Explorer.
	New File	Create a new macro (.mex) file.
	Open File	Open an existing macro (.mex) file. Or click on the triangle button to view a list of recently used macro files and select one to open.
	New Macro	Click on the New Macro button to start the process of building a macro in the Script Editor or Direct Editor. The Script Editor is the default editor. If desired, click on the Direct Editor icon while in the Script Editor to switch to the Direct Editor.
	Run Now	Click this button to run the macro that is highlighted in the Macro Express Pro - Explorer window.
	Edit Macro	Highlight a macro from the list of already created macros. Click on the Edit Macro button to open the Script Editor and display the macro. Or double click on the macro in the list to open it in the Script Editor.
	Capture	Starts the process for recording or capturing a macro. Keystrokes and mouse clicks/movements are recorded and saved.
	Quick Wizards	Opens the Quick Wizards dialog. There are 25+ different wizards to provide step by step instructions for creating various macro types.
	Recycle Bin	Macro Express has its own recycle bin. Any macros deleted from the Macro Explorer are sent to the Recycle Bin. If a macro is deleted by mistake it can be restored to its original file. Or the macros in the recycle bin may be removed permanently.

Converting from v3 or v4 to Macro Express Pro 6

By default, Macro Express Pro 6 installs to a different folder than Macro Express 3 or Macro Express Pro v4.

1. Macro Express Pro allows more than one macro to run simultaneously. When importing or opening a v3 macro file in Macro Express Pro, you will see the message displayed below. Answer Yes to place a File Lock command in each of the macros being converted. This prevents the macro from playing while another macro is already running.


Notice

For backwards compatibility, Macro Express Pro can add some commands that prevent macros from running simultaneously.
These commands can be removed or added later by editing the macro.

Do you want to add these locks?

2. Macro Express 3 files converted to the Macro Express Pro file format cannot be converted back to the older format. Keep backup copies of the version 3 macro files if you ever intend to revert back to Macro Express 3 use.

When converting the file to the Macro Express Pro format, a backup copy of the file in Macro Express 3 format is created. The backup is saved to the same location as the original file.

3. Macros written to automate Macro Express may not convert properly because the Macro Express Pro window titles have changed from v3. The accelerator keys have also changed.

4. Macros written in Macro Express Pro v4 will convert seamlessly to Macro Express Pro 6.

Installing Macro Express

Installing Macro Express

The steps for installing Macro Express Pro are fairly standard.

1. First download the program from www.macros.com/download.htm.
2. Save the downloaded file to a folder on your hard drive.
3. Locate the downloaded file and double click the .exe file to start the installation process.
4. Follow the on screen instructions to complete the install.

If not familiar with Macro Express, please read through the Quick Start Guide and Tutorial sections of the Help file.

Licensing Macro Express Pro

If you have purchased the license for Macro Express you will also receive a license code. From the Macro Explorer click on Help > License Macro Express to activate the License Information window. Enter your name and code exactly as they appear on the license instructions you received when you purchased Macro Express. Press OK to save the license information.

See the Advanced License Information topic for consideration of Administrator and non-Administrator login rights and Macro Express licensing.

Advanced Installation

Quiet Install

A quiet (or silent) install means Macro Express Pro will be installed without the user having to answer any questions during the installation process. This is helpful for installing Macro Express Pro from within another setup program or installing from a server on a network.

The /quiet command line parameter tells the Macro Express Pro installer to do a quiet Install. This option must include the slash and must be in all lowercase. The Macro Express Pro Setup program must be run with elevated privileges (as administrator) when the /quiet parameter is used.

Macro Express Pro will not be automatically launched when the /quiet option is used during installation.

Installation options

These command line options allow you to adjust how Macro Express Pro is installed. You can choose things such as where to install the program, whether Macro Express Pro should be put in the Startup folder so it is automatically launched when Windows starts, or whether the Quick Start Guide should be displayed after Macro Express Pro is installed.

Each option consists of a property name and value separated by an = sign. Property names must be in UPPERCASE, no spaces are allowed on either side of the = sign, and property values that are literal strings must be enclosed in quotation marks.

INSTALLPATH=c:\myprograms\Macro Express Pro
Used to override the default installation folder.

STARTUP=N
Tells the Macro Express Pro installer to not create a shortcut in the Windows Startup folder that launches Macro Express Pro when Windows starts up.

QUICKSTART=N
Do not display the Quick Start Guide when the program is launched for the first time.

LICENSENAME="Your License Name"
LICENSECODE="AB3E6-45F98-1200Z-34CC2-DF65R"
Use these properties to enter the Macro Express Pro license information. If the license is not entered here it may be entered when Macro Express Pro runs.

- ✓ **Note:** This license information is for demonstration purposes only. Use the License Information received upon purchase of Macro Express Pro. Both the License Name and License Code are required.

CONFIGFILEPATH="c:\InstallFiles\Macro Express Pro.wcfg"
This allows setting the default preferences at install time.

To do this, install Macro Express Pro on a computer and set the preferences to desired settings. Follow by clicking Tools > Export Program Configuration to create a file such as Macro Express Pro.wcfg containing the preferences.

NOEDITOR=Y
Installs the Macro Express Pro player but does not install the editors used to create or modify macros.

RUNQUIET=Y
This property installs Macro Express Pro and sets the most common preferences for a group of networked computers. These options apply to the user installing the program and any new users:

- The Splash Screen is disabled
- The Editor is not loaded when the program starts
- The Quick Start Guide is not displayed the first time the program runs
- Backups are disabled
- The First Time setup wizard is disabled
- Macro Halted messages are suppressed
- The samples.mex macro is not copied to the default macro files folder
- The display of periodic messages during 30-day trial is disabled
- Check for updates is set to Manually

Examples

The Macro Express Pro Setup program must be run with elevated privileges (as administrator) when the /quiet parameter is used.

Example 1

Installs Macro Express Pro in the c:\Programs Files (x86)\Macro Express Pro 6 folder with default preferences. After the installation is complete, the program will be licensed to John Smith.

```
MacroExpressPro6Setup.exe /quiet LICENSENAME=" John Smith"
LICENSECODE="AB3E6-45F98-1200Z-34CC2-DF65R"
```

Example 2

Installs Macro Express Pro in the d:\Programs\Macro Express Pro 6 folder. Macro Express Pro will not be launched when Windows is started. The Quick Start Guide will not be displayed when Macro Express Pro is first launched. After the installation is complete, the program will be licensed to John Smith.

```
MacroExpressPro6Setup.exe /quiet INSTALLPATH="d:\Programs\Macro Express Pro 6"
LICENSENAME=" John Smith" LICENSECODE="AB3E6-45F98-1200Z-34CC2-DF65R" STARTUP=N
QUICKSTART=N
```

Example 3

Installs Macro Express Pro in the c:\Programs Files (x86)\Macro Express Pro 6 folder with the most common preferences for a group of networked computers. After the installation is complete, the program will be licensed to John Smith.

```
MacroExpressPro6Setup.exe /quiet RUNQUIET=Y LICENSENAME=" John Smith"
LICENSECODE="AB3E6-45F98-1200Z-34CC2-DF65R"
```

Example 4

Installs Macro Express Pro in the c:\Program Files (x86)\Macro Express Pro 6 folder and sets the preferences to those contained in the MacroExpressPro.wcfg file. After the installation is complete, the program will be licensed to John Smith.

```
MacroExpressPro6Setup.exe /quiet LICENSENAME=" John Smith"
LICENSECODE="AB3E6-45F98-1200Z-34CC2-DF65R"
CONFIGFILEPATH="E:\MacroExpressProInstall\MacroExpressPro.wcfg"
```

✓ **Note:** This license information is for demonstration purposes only. Use the License Information received upon purchase of Macro Express Pro. Both the License Name and License Code are required.

Advanced License Information

Windows can be configured to allow different users to log onto a single computer. Some versions of Windows require a user to log in. Normally only one user is logged in at a time. One or more login names have access to the entire computer and are said to have administrative access rights. Other login names do not have administrative access rights. These login names have restrictions about what they can change on the computer.

If the Name and Code are entered into the License Information dialog when logged in as an administrator, Macro Express Pro is licensed for all users. If, however, the License Information is entered when logged in as a non-administrator, Macro Express Pro is only licensed for that user login.

Uninstalling Macro Express

To remove Macro Express from the computer follow the steps below:

1. Click on File > Terminate Macro Express Editor & Player to close the program.

2. From the taskbar, click the Start menu button to open the Start menu.
3. Open the Windows Control Panel.
4. Based on the version of Windows you are running, select the option from the control panel used to uninstall applications.
5. Highlight Macro Express Pro in the list of installed programs.
6. Again, depending on your version of Windows click on the option to remove the program.
7. Follow the on-screen prompts to remove Macro Express from your computer.

**There is not an uninstaller for the portable version. Use Windows/File Explorer to delete the files.

Tutorials

Tutorial - Introduction

The tutorials in this section demonstrate how to create various macro types and how to use many of the program features.

Creating Macros

There are several ways to create macros with Macro Express.

1. Capture (record) your keystrokes and mouse movements.

These two tutorials show different options for recording a macro.

Capture a Macro using the Capture Icon

Capture a Macro - System Hot Key

2. The Quick Wizard guides you through a series of questions and builds the macro.

Step through the Quick Wizard tutorial to create a reminder message.

Create Macro with Quick Wizard

3. Write the macro in the Direct Editor

Learn more about the Direct Editor with the HotKey tutorial and how it can be used to write macros.

Creating a Hot Key Macro

4. Scripting Editor

Use the Scripting Editor to create simple to complex macros. Each command is displayed on a separate line in the script for easy viewing and modification. These tutorials demonstrate the use of the Scripting Editor.

Creating a Shortkey Macro

Creating a Scheduled Macro

Using Window Controls

Copy and Paste between Two Applications

ASCII Text File Process

Using Text String Variables

Using Integer Variables

Other Tutorials

This tutorial shows how to build a popup menu. Create a menu of macros to display on screen and just select the one you want to run.

Create a Popup Menu

Determine where macros will run using the macro scope feature. This tutorial shows how to have macros only run in specific programs or windows, run in all but one or several windows, or run in all programs.

Macro Scope

The videos for each of these tutorials may be viewed online at www.macros.com.

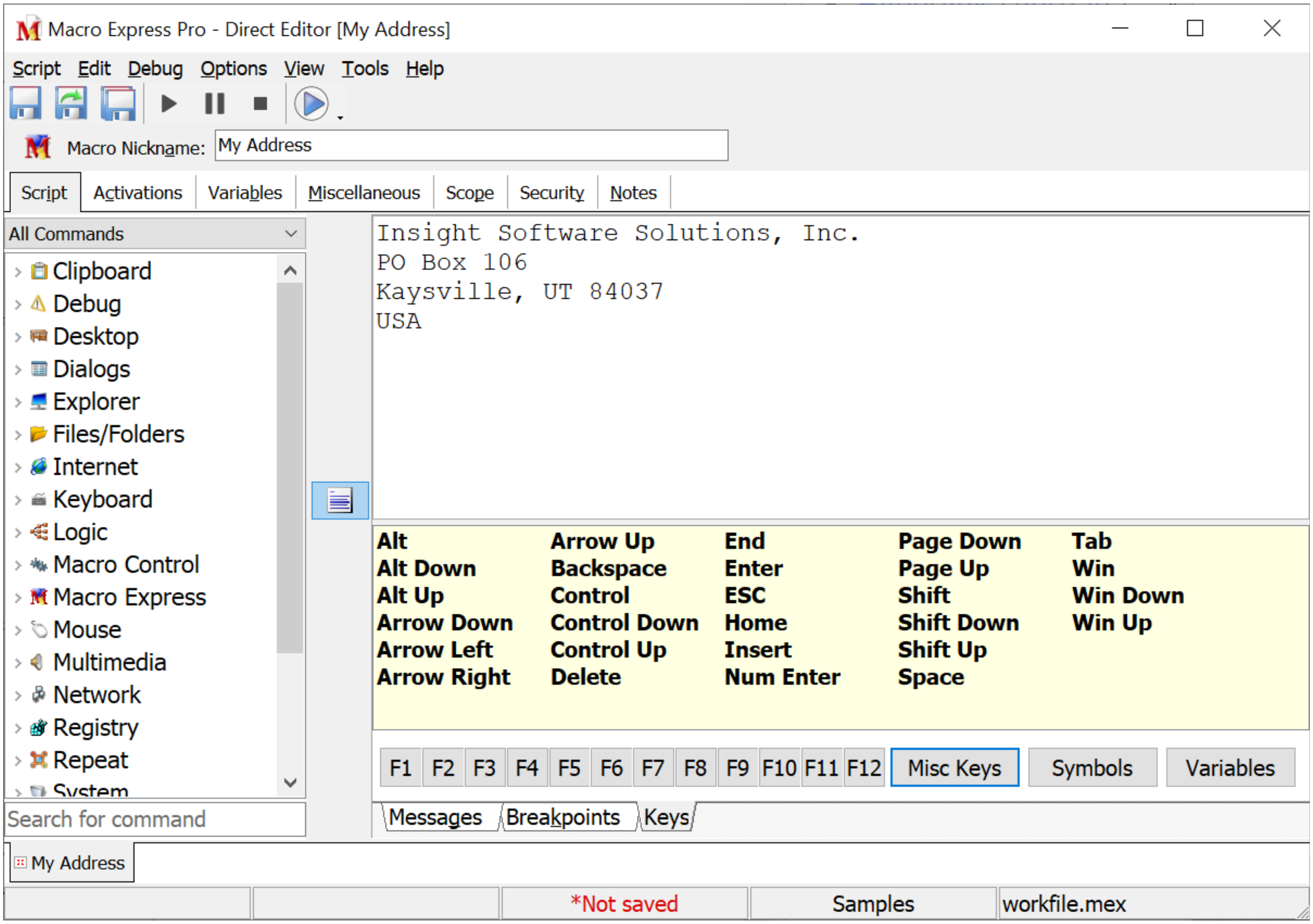
Creating a Hot Key Macro

In this exercise we will create a macro assigned to a hot key. When complete, pressing the hot key will run the macro.

Setting up the Macro and Assigning the Hot Key

1. From the Macro Express Pro - Explorer menu click on Macro > New Macro to open the Create New Macro window.
2. Hot Key should already be selected as the Activation.
3. Click on the HotKey edit box to place the cursor in this field.
4. Hold down the CTRL key, the SHIFT key and the T key at the same time and release them.
5. CTRL+SHIFT+T should appear in the edit box. This is the hot key we will define for the macro.
6. Enter "My Address" in the Nickname field.
7. Click OK to open the Script Editor.
8. Next, click on View > Direct Editor to open the Direct Editor, which we will use to create the macro.

Other methods to create macros are via the Quick Wizards or capturing (recording) a macro. These options are reviewed in other sections.



Macro Nickname: My Address

```
Insight Software Solutions, Inc.
PO Box 106
Kaysville, UT 84037
USA
```

*Not saved | Samples | workfile.mex

The Direct Editor looks like the screen shot above.

Creating the Macro

1. In the edit field, type in your address.
2. Click on the Scope Tab and make sure Global is selected. Global is the default option.
3. Click on the Save and Close button to save the macro and close the Editor.
4. Open the Macro Explorer window to see the newly defined macro, My Address, added to the list of available macros.

Playing Back the Macro

1. First, let's run the notepad program.

2. Once notepad is running, hold down the CTRL key, the SHIFT key and the letter T. Then release all the keys.
3. Your address will type out into the notepad program.

Creating a Shortkey Macro

This topic shows how to create a macro assigned to a Shortkey. When completed the macro will play back your email address.

Defining Shortkeys Default Settings

First, we need to define the default settings that will apply to all Shortkeys that you may later create.

1. From the Macro Express Pro - Explorer menu click on Options > Preferences to load the Preferences window.
2. Click on Activations and then the Shortkeys tab to display the Shortkeys options.
3. Select the Use Prefix Keys option and enter z in the Prefix Keys entry box.
4. Click the OK button to return to the Macro Explorer.

✓ **Note:** This setting requires that z be pressed before typing any Shortkey macro or it will not run.

Setting up the Macro and Assigning the Shortkey

1. From the Macro Express Explorer menu click on the New Macro icon to open the Create New Macro window.
2. Select Shortkey from the Activation list.
3. Type "email" (without the quotes) into the Original Shortkey entry space.
4. In the Nickname box, enter My Email Address and click OK to open the Script Editor.

Creating the Macro

1. In the Script Editor click on the Scope tab and make sure Global is selected from the drop down menu.
2. Now click on the Script tab to start writing the macro.
3. Scroll down the list of commands and click on the Keyboard command to

expand this listing.

4. Double click on the Text Type command found in this group to activate the Text Type window.
5. Type your email address in the Keystrokes field.
6. If you want to have the macro paste the text into your application via the Windows clipboard, select the Use Clipboard to Paste Text option from the drop down menu. Otherwise the text is inserted into your application using the Simulate Keystrokes default option, which types out the text letter by letter.
7. Press the OK button to save the entry and return to the Script Editor.

✓ **Note:** A new entry should now appear under the section called Macro Script. This entry should read Text Type: followed by the email address that you entered. If you made a mistake entering your email address, you can double click on this entry and edit the address in the Text Type window.

8. From the Script Editor, click on the Save and Close button to save the macro and close the Script Editor.
9. Open the Macro Explorer to see the newly defined macro, "My Email Address", added to the list of available macros.

Playing Back the Macro

1. To test the macro, run the notepad program.
2. Once notepad is running, type the following: zemail
3. Watch as your email address is typed into the notepad program.

Creating a Scheduled Macro

This page demonstrates creating a Scheduled Macro, one that plays back on a schedule. To start, click on the New Macro icon from the Macro Express Pro - Explorer window to open the Create New Macro window.

Setting the Schedule

1. Select Schedule from the Activation list to display the scheduling options.
2. For this example select the "When a Specified Time has Elapsed" from the Run the Macro drop down menu.
3. In the Schedule tab, select 0 hours, 0 minutes and 15 seconds.
4. Click on the Range tab.
5. Select the Play Macro Indefinitely option from the drop down list.
6. Leave the "Start calculating the next runtime at" fields alone. Check to make sure the time listed is not later than the time on the computer clock. Make sure that the date displays today's date.
7. Leave the "Run the macro between these hours of the day" option not checked.
8. Leave every day of the week selected.
9. Enter "Display the Time" as the macro nickname.
10. Click the OK button to open the Script Editor.

Based on the schedule just set, the macro will run every 15 seconds.

Setting the Program Scope

We'll create this macro to only play back in the Notepad window and only if it is already running.

1. Click on the Scope tab and select Window/Program Specific.
2. Click on the Add Window button to open the Select Window dialog.
3. Click on the Visible tab and enter "notepad" (without the quotes) in the

Window Title edit field.

4. Select Partial Match from the drop down list.
5. Leave the option "A window with this title must be on top" unchecked.
6. Click OK to save the Scope information.

Building the Macro

1. Click on the Script tab to start writing the macro.
2. To begin, expand the Windows/Programs category of commands.
3. Double click on the Window Activate command.
4. Enter "notepad" (without the quotes) for the Window Title and click on the OK button.
5. Select Partial Match from the drop down list and click OK to save the information and insert into the macro script.
6. Expand the Variables category of commands.
7. Double click on Date/Time to open the Date/Time window.
8. In the Date/Time tab, make sure the "Set it to the current date/time" option is selected.
9. Click on the Operation tab.
10. Select the "Type the Date/Time stamp out" option.
11. Enter "h:mm:ss" (without the quotes) in the Choose a Date/Time Format edit box.
12. Click OK to save the changes.

✓ **Note:** There should be two lines of script displayed in the Script Editor. Double click on either of these entries to make changes.

13. From the Script Editor, click on the Save and Close button.
14. Open the Macro Explorer window to see your newly defined macro, Display the Time, added to the list of available macros.

Playing back the Macro

As long as Macro Express is running, a timed macro should be ready to execute. However, in this example the timed macro will only execute if Notepad is running.

1. To test the macro, run the Notepad program.
2. Once Notepad is running, just wait.
3. The time is typed into Notepad every 15 seconds.

Disabling the Macro

After verifying that the macro works, either delete or disable this macro since you won't want this Scheduled Macro to continue executing every 15 seconds indefinitely! To delete the macro, highlight the macro in the Macro Explorer list and choose "Macros > Remove Macro" from the menu.

To keep this example around, but not to play back every 15 seconds, right mouse click on the macro. Left click on Disable to disable the macro. The macro is now displayed in the Explorer window with italicized red font and possibly a line drawn through it, depending on Preference settings. This quickly alerts that the macro is disabled and will not play back until it is enabled.

Capture a Macro using the Capture Icon

This example shows how to record a macro. We'll capture the macro in the Windows Paint program.

Once Paint is running, we want to **switch** directly to Macro Express. If running Macro Express in the notification area, left click on the Macro Express icon in the notification area; otherwise click on Macro Express in the taskbar.

Defining the Macro

1. From the Macro Express Pro - Explorer window click on the Capture icon on the sidebar to open the Create New Macro window.
2. HotKey should be the default Activation.
3. Click on the HotKey edit box to place the cursor in this field.
4. Next, hold down the SHIFT key, the CTRL key and the U key all at the same time and release them. This is the hot key assigned to the macro we record.
5. Make sure the box next to Wait for the Hot Key to be Released Before Activation is checked.
6. Enter My Paint Drawing as the nickname and click OK.

Defining the Capture Options

The Capture a Macro window opens so that the default capture options may be modified before capturing the macro. The default settings are defined from the Macro Explorer window in Options > Preferences > General > Capture. Let's adjust the capture settings for this macro capture.

Set Scope

The first thing to do is to set the scope so that the macro only starts in the MS Paint program.

1. Click on the Set Scope button to open the Set Scope window.
2. Change the scope to Window/Program Specific if not already selected.
3. Click on the Add Program button to open the Select a Program Window.

4. Click on the Running Processes tab and highlight mspaint.exe.
5. Make sure that the "This program must have a window on top" option is checked.
6. Click OK and click OK again to save and return to the Capture a Macro window.

Other Capture Options

1. Check the Capture the Keystrokes option.
2. Check the Capture Mouse option and select the Relative to Window option from the drop down menu.
3. Make sure the Capture and Restore the Window Size option is checked.
4. Only Capture the Mouse Clicks should not be checked.
5. From the drop down menu select Capture the Macro Speed at Actual Speed.
6. Choose "untitled - Paint" from the drop down list of the Begin Capture in Window section.

Start Recording

Now we are ready to record the mouse movements, clicks and keystrokes.

1. Click on the Capture button. This closes the Capture a Macro window and opens the Paint window.
2. At this point the macro is now recording. We'll just draw a simple face in Paint.
3. Click on a color.
4. Draw a circle and place two eyes, a nose and mouth within the circle.
5. Terminate the capture by pressing the hot key we assigned to the capture - CTRL+SHIFT+U.
6. A message appears indicating that the capture is complete.

Play back the Macro

1. Open the Paint window. If it is not already the active window, click on the Paint entry in the task bar.
2. We need to start with a clean screen in Paint. So click on File > New in Paint and follow the prompts to have a fresh screen.
3. Press the hot key CTRL+SHIFT+U to start the macro playback.
4. The face just drawn will be redrawn complete with any hesitations made while capturing!

Capture a Macro - System Hot Key

Macro Express has eight hot keys automatically assigned as System Macros. One of these System Macros is called Start / Stop Capture and by default is defined as CTRL+ALT+SHIFT+X. Pressing CTRL+ALT+SHIFT+X starts the recording of a macro. Pressing it a second time stops the recording. The system macro hot key combination may be changed to another hot key combination not presently being used in Macro Express.

Capture Settings

Before recording any macros using the System Hot Key we first need to define the default capture settings.

1. From the Macro Express Pro - Explorer click on Options > Preferences > General and select the Capture tab to display the Capture default settings.

✓ **Note:** These are the settings that are used when the System Start / Stop Capture Hot Key function is invoked. These are also the default settings that appear when clicking on the Capture icon to start a macro recording.

2. Leave the scope set to global.
3. Check the Capture Keystrokes option.
4. Check the Capture Mouse option and select the Relative to Window option from the drop down menu.
5. Make sure the Capture the Window Size option is checked.
6. We also want to just capture the mouse clicks. We don't need to capture every movement of the mouse. If we were drawing or dragging with the mouse, then we would want to capture all mouse movements.
7. Turn off the "Capture at a specific speed" option to capture the macro without any delays. Be careful when doing this. In some applications the macro may play back too fast and not work correctly.
8. Check the Display a Prompt Before Capturing box to display a message indicating the capture is starting.
9. Click OK to save the settings.

Capturing the Macro

An important thing to remember when recording a macro is to make sure that the program to start recording in is active and on top. Our example recording uses the WordPad application.

When recording, try to use keystrokes as much as possible, rather than the mouse. But there are certain functions that do not have keystroke equivalents and require the mouse. For this example we’ll record a simple macro to change the font color. This can be done with just keystrokes.

1. First run the Windows WordPad program.
2. Once focus is on WordPad, press CTRL+ALT+SHIFT+X.
3. A notice may appear warning about the use of the ALT key. Click OK.
4. Next appears a notice that the macro is about to start recording. Click the OK button to start recording.

The macro is now recording. Let's do something fairly simple such as selecting a new font color.

5. Press Alt + h then type C2 to open the text color choices.
6. Arrow down several times to choose a color and press Enter.
7. Terminate the macro capture by pressing the same system macro, CTRL+ALT+SHIFT+X.

Macro Information

The "Create New Macro" window opens for assigning a macro activation.

1. Select HotKey as the Initial Activation if not already the default.
2. Place the cursor in the hotkey edit window and press CTRL+SHIFT+S.
3. Release the keys to display CTRL+SHIFT+S in the edit box and use as the macro activation hot key.
4. Type WordPad Color Change in the nickname field and press OK.

Play back the macro

1. Open a WordPad window with a little bit of text in it.
2. Highlight the text in the WordPad window.
3. Press the hot key CTRL+SHIFT+S to run the macro.
4. The text color is changed to what we chose in the macro.

Create Macro with Quick Wizard

In this tutorial we'll demonstrate how to create a macro using the Quick Wizard. The Wizards will guide us step by step through the creation process. For this example we will create a reminder message to display on screen at the time we designate.

1. From the Macro Express Pro - Explorer window click on the Quick Wizards icon to start the wizard.
2. Select the first choice shown - Create a Reminder, etc. option and click on the Next button.
3. Select the Create a Reminder option and click on the Next button.
4. The next screen just points out a few possible uses for the reminders. Click Next to continue.
5. In the Edit window enter the text. For this example, enter "This is a reminder to purchase Macro Express at www.macros.com/orders.htm." Click on the Next button to continue.
6. Select the Sound Beep notification option to also play a beep when the macro runs.
7. Check the box next to Delete this macro after displaying the reminder. Once the macro has run, the macro is removed from the list of macros in the Macro Explorer and sent to the Macro Recycle Bin.
8. Click Next to set up the schedule for the reminder message.
9. From the drop down list, change the Macro Run option to Once.
10. Change the Run At time to 5 minutes ahead of the current time.
11. Make sure the date is set to today's date and click Next.
12. Insert "My Reminder" in the Macro Nickname edit field (without the quotes) and click Finish to close the Wizard.

Open the Macro Express Explorer window to see the new macro in the list of created macros. Wait the 5 minutes specified and the reminder message will pop up on the screen.

Close the message and the macro is removed from the macro list in the Explorer window and sent to the Macro Recycle Bin.

Create a Popup Menu

This example demonstrates how to create a popup menu. Before creating a popup menu, you must already have some macros defined.

- ✓ **Note:** If you have not done so, first create both a Hot Key macro and a Shortkey macro as described in the Creating a Hot Key and Shortkey tutorials. This example is based on the assumption that you have already created these macros.

Initial setup of the menu

1. From the Macro Express Pro - Explorer menu click on Macro > New Macro to open the Create New Macro window.
2. Make sure that the Hot Key activation is selected and place the cursor in the hot key edit box.
3. Hold down the ALT key, the SHIFT key and the Z key all at the same time and release them. ALT+SHIFT+Z should appear in the edit box. This is the hot key we will use for the popup macro.
4. Enter My Menu as the macro nickname.
5. Check the box next to build as a Popup macro and click OK to open the Menu Builder.

The Menu Builder

The Menu Builder contains two columns. The left hand column lists all macros available for use in the menu. The right hand column displays the macros that we assign to the menu.

Adding Macros to the Menu

For this example we will only include two macros in the menu. But in practice you can add many more.

a. Highlight the My Address macro (the one we created earlier) and then click on the green Insert ➡ button.

b. Next highlight the My Email Address macro (created earlier) and click on the green Insert button.

These two macros should now be displayed in the right hand column.

Select Menu Style and Positioning

a. Before saving the menu, click on the Miscellaneous tab and make sure the Menu is Enabled button is checked.

b. Select Nickname and Scope from the drop down menu as the Menu Style.

c. Enter 500 in both the X and Y menu positioning fields.

d. Check the box next to Place an icon in the notification area option.

Select an Icon to Display in Notification Area

The next step is to select an icon to use with the menu. Right above the Menu tab and to the left of the Menu Nickname is an icon. Hovering the mouse cursor over the icon displays the text - Change Macro Icon.

a. Click on this icon to open the Select an Icon window.

b. Click on the Browse (...) button to open a view of the Macro Express Pro folder.

c. Double click on the Icons folder to display the folder contents.

d. You will see a number of .dll files containing icon sets - .ico files.

e. Double click on the IconsLG.dll file to open the Select an Icon window.

f. Click on the blue push pin icon, item number 1, and click OK.

g. The icon in the Build a Menu window has been updated to reflect the change made.

h. This new icon will be displayed in the notification area.

i. Click on Save and Close to save the menu.

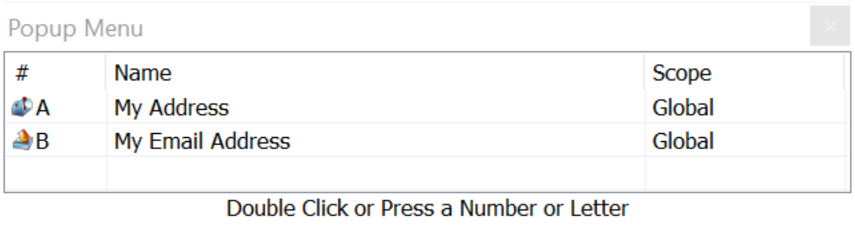
j. Return to the Macro Explorer window to see the newly defined popup macro, ALT+SHIFT+Z, added to the list of available macros.

Testing the Popup Menu

There are two methods for activating the menu we created. One is to press the hot key we assigned or to left mouse click on the icon displaying in the notification area.

Hot Key Activation

Let's test out the popup menu and run the macros in Notepad. Once notepad is running, hold down the ALT+SHIFT+Z keys.



Popup Menu

#	Name	Scope
A	My Address	Global
B	My Email Address	Global

Double Click or Press a Number or Letter

A menu like the one above should appear, listing the two macros previously defined. Either press the corresponding letter or number next to the macro or double click on the macro in the menu to run the selected macro in notepad.

Notification Area Activation

Another way to invoke the popup menu is to left click on the blue push pin icon in the notification area. This icon should now be showing in the notification area, because of the settings defined previously.

Menu Styles

The popup menu may be displayed in four different forms, including Nickname and Scope, which is what we are currently using. Now take a quick look at the other options for displaying the menu.

First Part of Macro

a. Double click on the Popup Menu macro from the list of macros in the Macro Explorer to open the Menu Builder window.

b. Click on the Miscellaneous tab.

c. Select the First Part of Macro option from the Menu Style drop down list.

d. Click on Menu > Save Menu to save the change.

e. Left click on the "blue push pin" icon in the notification area.

f. The menu opens displaying the first part of the contents of each macro.

Icon

a. Close the popup menu and return to the Menu Builder.

b. Click on the Miscellaneous tab and select the Icons Only option from the Menu Style drop down list.

c. Next, choose the Horizontal option and enter 10 for Maximum Icons per Row.

d. Click on Menu > Save Menu to save the change.

e. Now click on the "blue push pin" icon in the notification area.

f. The menu displays a list of icons.

g. Click on one of the icons in the menu to run the macro.

Windows Default

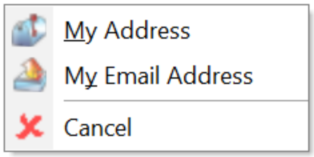
a. Return to the Menu Builder and click on the Miscellaneous tab.

b. Select the Windows Default option from the Menu Style drop down list.

c. Click on Menu > Save Menu to save the change.

d. Now click on the "blue push pin" icon in the notification area.

e. The menu displays as shown below.



f. Click on the macro you want in order to execute the macro.

g. Windows also assigns underscores to each of the macros, where possible.

h. So in the example above, pressing the M key activates the My Address macro.

Tutorial - Macro Scopes

Macro Scope

The macro scope indicates under what conditions a macro should play back. Four categories are available. They are Global, Global Except, Window Specific and Program Specific.

A. Global macros play back in all Windows programs.

B. Global Except macros run in all Windows programs or windows except the application or window specified.

C. Window Specific macros run only a window that has a matching Window Title as specified by the macro.

D. Program Specific macros only play back in a selected program.

Let's look at some real examples of how this all works. To illustrate the process, let's run two different programs for testing the macro and scope. Run Windows Notepad and Windows WordPad.

Global Macro

We have actually already created a couple of Global macros. If you have not created these macros, please refer to the Hot Key section for instructions on creating the CTRL+SHIFT+T macro. A global macro plays back in all Windows programs.

To illustrate, bring Notepad to the foreground (click on it in the taskbar). Now press CTRL+SHIFT+T to see the address typed into the program. Now bring WordPad to the foreground (click on it in the taskbar). Press CTRL+SHIFT+T to see the address typed into WordPad.

Program Specific Macro

We will first change the CTRL+SHIFT+T macro to become a Program Specific macro.

1. Minimize Notepad and WordPad.
2. With Macro Express Pro - Explorer the active window, highlight the CTRL+SHIFT+T macro (nickname of "My Address") from the Macro Explorer

and click on the Edit Macro button to open the Script Editor.

3. Click on the Scope tab from the Script Editor.
4. From the Current Scope drop down menu, select Window/Program Specific.
5. Click on the Add Program button to open the Select a Program window.

✓ **Note:** The Select a Program window has two tabs, Start Menu and Running Processes. The Start Menu tab allows you to look through programs in your Start Menu to locate the .exe file. Or you can click on the Browse button to find the file anywhere on your computer.

6. For our example, click on the Running Processes tab to list all programs currently running on the computer.
7. Make sure that the box next to "This program must have a window on top" is checked.
8. Scroll through the list of programs running and find notepad.exe.
9. Highlight it in the list and click on the OK button.
10. To see what effect the changes have on the macro and bring Notepad to the foreground (click on it in the taskbar). Press CTRL+SHIFT+T to see the address typed into the program.
11. Now bring WordPad to the foreground (click on it in the taskbar). Press CTRL+SHIFT+T. Nothing happens because the macro is set to only work in Notepad.

Global Except

The next step is to change the CTRL+SHIFT+T macro to become a Global Except macro.

1. First minimize Notepad and WordPad.
2. With Macro Express Pro - Explorer the active window, highlight the CTRL+SHIFT+T (nickname of "my address") macro from the Macro Explorer and click on the Edit Macro button to open the Script Editor.
3. Click on the Scope tab.
4. From the Current Scope drop down menu, select Global Except.

We can now choose to have the macro work everywhere except a specific window or program that we select. For this tutorial we will use the program option.

5. Notepad.exe should be displayed as the program that will not allow the macro to work.
   a. If not, highlight any entries and click on the Remove button to delete them.
   b. With no programs or windows selected click on the Add Programs button.
   c. From the Select a Program window click on the Running Processes tab to list all programs currently running on the computer.
   d. Make sure that the box next to "This program must have a window on top" is checked.
   e. Scroll through the list of programs running and find notepad.exe. Highlight it in the list and click on the OK button.
6. To see what effect the changes have on the macro, bring Notepad to the foreground (click on it in the taskbar) and press CTRL+SHIFT+T. Nothing happens because the macro is defined to not work with Notepad.
7. Now bring Wordpad to the foreground (click on it in the taskbar). Press CTRL+SHIFT+T to see the address typed into the program.
8. Activate another program that accepts text input such as an email program and press CTRL+SHIFT+T. The address types into the program.

Window Specific Macro

The last step is to change the CTRL+SHIFT+T macro to become a Window Specific macro. The macro only plays if a window with a specific title is open or on top. (A Window Title is the text that appears in the top caption bar of most windows).

1. First minimize WordPad and Notepad.
2. With Macro Express Pro - Explorer the active window, highlight the CTRL+SHIFT+T macro from the Macro Explorer and click on the Edit Macro button to open the Script Editor.
3. Click on the Scope tab from the Script Editor.
4. From the Current Scope drop down menu, select Windows/Program Specific.

5. Highlight Notepad.exe listed in the edit field.
6. Click on the Remove button to delete this entry.
7. Click on the Add Window button to open the Select Window dialog.

✓ **Note:** The Select Window dialog contains two tabs, Visible and Hidden. You may select a window that is visible or a hidden program window that is never displayed. For this tutorial we will use a Visible window.

8. Highlight Untitled - Notepad.
9. Check the box next to "A window with this title must be on top".
10. Click OK to save.
11. To see what effect these changes have on the macro, bring Notepad to the front of the screen (click on it in the taskbar). Press CTRL+SHIFT+T to see the address typed into the program.
12. Now bring WordPad to the front of the screen (click on it in the taskbar). Press CTRL+SHIFT+T. Nothing happens because CTRL+SHIFT+T is set to only work in an Untitled - Notepad window.

Copy and Paste Between Two Applications

Below is a simple example of how to set up a copy and paste routine between two applications. Use this as a guideline for creating routines between other applications.

You might be tempted to try and use the ALT+TAB key to switch between programs. This is actually not always reliable and in fact will not even work via Macro Express on some computers due to the low level nature of this key command. The solution is to use the Windows Activate command.

Setting up the Macro

Choose a HotKey and macro Scope.

1. From the Macro Express Pro - Explorer menu click on Macro > New Macro to open a Create New Macro window.
2. The Initial Activation should default to HotKey.
3. Click on the HotKey edit box to place the cursor in this field.
4. Hold down the CTRL key, the SHIFT key and the P key all at the same time and release them.
5. CTRL+SHIFT+P should appear in the edit box.
   This is the hot key used to activate the macro.
6. Enter Copy and Paste in the Nickname field and click OK to open the Script Editor.
7. Click on the Scope tab and make sure that the macro scope is set to global.
8. Then click on the Script tab to start building the macro.

Building the Macro in the Script Editor

The object of the macro is to repetitively copy one line at a time over to WordPad.

1. This first command instructs the macro to bring notepad to the front and be ready to receive keystrokes.

   a. Scroll through the list of command categories and click on

Windows/Programs to expand the list.

b. Double click on the Window Activate command.

c. In the Activate Window field, enter "notepad" (without the quotes) as the Window title.

d. Make sure the Partial Match option is selected. Click on the OK button to save the entry.

2. The next step in this process is to highlight a single line of text byusing Shift+End.

   a. To do this scroll through the list of command categories and click on Keyboard to expand the category.

   b. Double click on Text Type to open the Text Type window.

   c. Click on Shift and then End in the yellow area of this window to insert <SHIFT><END> into the Keystrokes edit field.

   d. Click OK to save the changes.

3. The next step is to copy the highlighted text to the Windows Clipboard.

   a. Scroll through the list of command categories and click on Clipboard to expand the category.

   b. Double click on the Clipboard Copy command to insert into the script.

4. Before pasting the text into WordPad, we will first move the cursor to the next line in notepad so that the macro is ready to copy the next line of text.

   a. To do this, scroll through the list of command categories and click on Keyboard to expand the category.

   b. Then double click on Text Type to open the Text Type window.

   c. Click on Home and then Arrow Down in the yellow area of this window to insert <HOME><ARROW DOWN> into the Keystrokes edit field.

d. Click OK to save the changes.

5. This step activates WordPad and allows text entry into this window.

   a. To activate WordPad, click on the Windows/Programs category of commands to expand the list.

   b. Double click on the Window Activate command.

   c. In the Activate Window field, enter "wordpad" (without the quotes) as the Window title.

   d. Make sure the Partial Match option is selected.

   e. Click on the OK button to save the entry.

6. The next step of the macro is to paste the text from the clipboard into WordPad.

   a. Scroll through the list of command categories and click on Clipboard to expand the category.

   b. Double click on the Clipboard Paste command to insert into the script.

7. Now place an <ENTER> in the macro to move the cursor to a new line in WordPad.

   a. Scroll through the list of command categories and click on Keyboard to expand the category.

   b. Double click on Text Type to open the Text Type window.

   c. Click on Enter in the list of options in the yellow area of this window to insert <ENTER> into the Keystrokes edit field.

   d. Click OK to save the changes.

8. Two final issues need to be addressed. The first is to place the macro in a loop so that it copies multiple lines. In this example we will set the macro to copy the first 5 lines in notepad over to WordPad.

   a. Highlight the first line of the macro script - Activate Window: "notepad".
   b. Scroll through the list of command categories and click on Repeat to expand the category.
   c. Single click on the Repeat Start option and then click on the Insert Command Before Selected arrow - the top green arrow of the three insert arrows to open the Repeat Options window.
   d. Make sure the Repeat Start option is selected and enter 5 in the Number of Times to Repeat entry field.
   e. Click OK to save the change and insert the Repeat command as the first line of the macro.

9. This step closes the repeat loop.

   a. Single click on the End Repeat command
   b. Click on the Add Command to End of Script button - the bottom of the three Insert buttons to insert an End Repeat command as the last command of the macro script.

10. The last thing is to address a possible timing issue. It's possible for Macro Express to send the Clipboard Paste command before WordPad has fully activated. So what we need to do is insert a short delay right after activating either window.

    a. Add a Delay command to the script.
       1) Scroll through the list of command categories and click on Timing to

expand the list.

2) Then double click on Delay to open the Set Delay window.

3) Select the Delay in Seconds option and enter .5 (one half second).

4) Click OK to save the change.

b. Make a duplicate of the Delay command

1) Find the delay command we just created in the script and highlight it.

2) Then click on the Duplicate button to the right of the script to make a copy of this command.

You should now have two Delay commands listed in the script.

c. Positioning the Delay commands in the script.

1) Highlight one of the Delay commands and click on the Up or Down arrow on the right side of the Macro Script.

2) Move the delay command so that it is positioned below the Activate Window: "notepad" command and above the Text Type:<SHIFT><END> command.

3) Highlight the other Delay command and move it so that it is positioned below the Activate Window: "wordpad" command and above the Clipboard Paste command.

11. Click on File > Save to save the macro.

Macro Script

The macro script should look like what is displayed below. If the macro does not look like this, arrange the commands so that they are in the following order.

```
Repeat Start (Repeat 5 times)
  Window Activate: notepad
  Delay: 0.5 seconds
  Text Type (Simulate Keystrokes): <SHIFT><END>
  Clipboard Copy
  Text Type (Simulate Keystrokes): <HOME><ARROW
DOWN>
  Window Activate: wordpad
  Delay: 0.5 seconds
  Clipboard Paste
  Text Type (Simulate Keystrokes): <ENTER>
End Repeat
```

Testing the Macro

1. To test the macro run the notepad and WordPad applications.
2. Enter five lines of text into notepad or copy the first five lines of the macro script displayed above and paste them into notepad.
3. Make sure you don't have any blank lines between the lines of text.
4. Next, place the mouse cursor at the beginning of the first line of text in notepad.
5. Then press CTRL + Shift + P to start the macro and watch it switch between both programs, copying each line from notepad and pasting into WordPad.

Using Text String Variables

Below is an example of using Text String Variables. There are hundreds, even thousands of uses for Text String Variables. This example provides an idea of how they can be used and manipulated.

This macro, when completed, fixes typing transposition errors such as "adn" or "teh" instead of "and" or "the". Just place the cursor between the two transposed letters and press the assigned hot key to correct the typing error.

Setting up the Macro

1. From the Macro Express Pro - Explorer menu click on Macro > New Macro to open the Create New Macro window.
2. Hot Key should already be selected as the Activation.
3. Click on the HotKey edit box to place the cursor in this field.
4. Hold down the F4 key and release it.
5. F4 should appear in the edit box. This is the hot key we will define for the macro.
6. Enter "Fix Transposed Letters" (without the quotes) in the Nickname field.
7. Click OK to open the Script Editor.
8. In the Script Editor click on the Scope tab and make sure that the macro scope is set to global.
9. Click on the Script tab to start building the macro.

Building the Macro in the Script Editor

1. Add a Comment line to the script as a reminder on how to use the macro.

   a. Scroll through the list of command categories and click on Macro Control to expand the list.
   b. Double click on the Comment command.

c. Type "Place cursor between the transposed letters" (without the quotes) in the Comments Text edit field.
d. Click OK to save the entry.

The text is displayed in the Script Editor in bold blue font.

2. The next step is to highlight the two letters that are reversed using the Text Type command.

a. Scroll through the list of command categories and click on Keyboard to expand the category.
b. Double click on Text Type to open the Text Type window.
c. Click on Arrow Left from the list of options in the yellow area of this window.
d. Then click on Shift Down, Arrow Right, Arrow Right and Shift Up.

The entire entry should look like the following: <ARROW LEFT><SHIFTD><ARROW RIGHT><ARROW RIGHT><SHIFTU>.

e. Click OK to save and return to the Script Editor.

3. This next step cuts the highlighted text and copies it to the clipboard.

a. Scroll through the list of command categories and click on Clipboard to expand the category.
b. Double click on Clipboard Cut to insert the clipboard cut command as the next item in the script.

4. Now save the clipboard contents to a text string variable.

a. Scroll through the list of command categories and click on Variables to expand the category.

b. Double click on Variable Set String to open the Variable Set String window.

c. Select the Set Value from Clipboard option from the drop down menu.

d. Enter %T[1]% in the Destination Variable field and click OK to save.

5. Save the first letter to a new variable - T[2].

a. Scroll through the list of command categories and click on Variables to expand the category.

b. Double click on Variable Modify String to open the Variable Modify String window.

c. Select Copy Part of Text from the drop down list to copy the first letter of the variable T[1] and save it to variable T[2].

d. Enter %T[2]% in the Text Variable field and enter %T[1]% the Variable to copy from field.

e. Enter the number "1" (without the quotations) in the Starting Position field.

f. In the Number of Characters to Copy field enter the number "1" (without the quotations).

g. Click OK to save.

6. Save the second letter to a separate variable - T[3].

a. Scroll through the list of command categories and click on Variables to expand the category.

b. Double click on Variable Modify String to open the Variable Modify String window.

c. Select the Copy Part of Text option from the drop down list.

d. Enter %T[3]% in the Text Variable field and enter %T[1]% in the

Variable to copy from field.

e. Enter the number "2" (without the quotations) in the Starting Position field.

f. In the Number of Characters to Copy field enter the number "1" (without the quotations).

g. Click OK to save.

7. This last step plays back the corrected text.

   a. Scroll through the list of command categories and click on Keyboard to expand the category.

   b. Double click on Text Type to open the Text Type window.

   c. In the Keystrokes edit field, type the following: %T[3]%%T[2]%.

   This types out the second letter of the transposed text first, reversing the order and correcting the typing error.

   d. Click OK to save and return to the Script Editor.

Macro Script

The macro script should look like what is displayed below. If the macro doesn't look like this, arrange the commands so that they are in the following order.

```
// Place cursor between transposed letters.
Text Type (Simulate Keystrokes): <ARROW LEFT><SHIFTD><ARROW
RIGHT><ARROW RIGHT><SHIFTU>
Clipboard Cut
Variable Set String %T[1]% from the clipboard contents
Variable Modify String %T[2]% to a substring in %T[1]% starting at 1 and 1
characters long
Variable Modify String %T[3]% to a substring in %T[1]% starting at 2 and 1
characters long
Text Type (Simulate Keystrokes): %T[3]%%T[2]%
```

Click on Script > Save Macro to save the macro just created.

Testing the macro

To test the macro, run the notepad program. (Depending on your version of Windows, click on the Windows Start button, enter notepad in the Search field and press enter. Or right click on the Windows Start button, select Run, enter notepad and click OK.)

Type the following text into notepad - Tihs is a tset. (Include the typing errors.)

Insert the cursor between the "i" and "h" in the word "Tihs". Press the F4 key to play back the macro. The letters are reversed to correct the spelling to "This". Try the same thing with the misspelled word "tset".

- ✓ **Note:** If assigning original names to variables, rather than using T[1], D[1], etc., then the following characters are not allowed in the variable name: [ ] < > , . - + = \ | / * %

Advanced - Using Integer Variables

The object of the completed macro is to generate three random numbers from 1 to 100 and display the results. This macro requires using Integer Variables as well as Arrays.

Setting up the Macro

1. From the Macro Express Pro - Explorer menu click on Macro > New Macro to open the Create New Macro window.
2. Hot Key should already be selected as the Activation.
3. Click on the HotKey edit box to place the cursor in this field.
4. Hold down the CTRL key, the SHIFT key and the R key all at the same time and release them.
5. CTRL+SHIFT+R should appear in the edit box. This is the hot key we will define for the macro.
6. Enter "Generate Random Numbers" (without the quotes) in the Nickname field.
7. Click OK to open the Script Editor.
8. In the Script Editor click on the Scope tab and make sure that the macro scope is set to global.
9. Click on the Script tab to start building the macro.

Building the Macro in the Script Editor

1. Scroll through the list of commands and expand the Variables category. Double click on the Variable Set Integer command to open the window of the same name.
2. From the drop down list select the Set to a Random Value option.

Creating the variable and array

a. Place the cursor in the Destination Variable field to display the Variables button to the right of the field.

b. Click on the Variables button to open the Insert Variable window.

c. Click on the Add button to open the Variable Properties window.

d. Enter "Random" in the Variable Name field.

e. From the drop down Variable Type list select Integer Variables.

f. Check the box next to Create as an Array.

g. Enter 3 (three) in the Elements field and click OK to return to the Insert Variable window.

h. Expand the list of Integer variables (click on +) to see the Random variable created.

i. Highlight the Random variable to see Random[ ] displayed in the Variable Name field.

j. Insert the number 1 (one) between the brackets so it displays as Random[1].

k. Click OK to return to the Variable Set Integer command.

3. The Destination Variable field should now display %Random[1]% as the variable.
4. Enter the number 1 (one) in the Minimum Value field.
5. Enter the number 100 (one hundred) in the Maximum Value field.
6. Click OK to save the command and place it in the Script Editor.

Generating the Random Numbers

7. In the Script Editor highlight the Variable Set Integer command shown in the script and click on the Duplicate button twice (on the right hand side of the script) so that you should now see three identical lines in the script.
8. Double click on the second line in the script to open the Variable Set Integer command.
9. Edit the Variable name by replacing the 1 (one) in brackets with a 2 (two). It

should now display as %Random[2]%.

10. Click OK to save and return to the Script Editor.

11. Double click on the third line in the script to open the Variable Set Integer command.

12. Edit the Variable name by replacing the 1 (one) in brackets with a 3 (three). It should now display as %Random[3]%.

13. Click OK to save and return to the Script Editor.

Have the macro display the results

These next steps set up the display of the three random numbers in a text box. After completing lines 14 through 27, the Text Box Display command window should look like the image below.

14. Scroll through the list of commands and expand the Dialogs category. Double click on the Text Box Display command to open the window of the same name.

15. Insert "Display random numbers" in the Header field.

16. Place the cursor in the Text Box field and the Variables button will appear to the right of the field.

17. Click on the Variables button and expand the Integer Variables type.

18. Select the Random variable to place Random[ ] in the Variable Name field.

19. Insert 1 (one) between the brackets so that it reads as Random[1] and click OK.

20. This places %Random[1]% in the Text Box field.

21. Copy the text %Random[1]% to the clipboard. Highlight the text and press Ctrl + C to copy to the clipboard.

22. Move the cursor to the second line (Right Arrow and the Enter key).

23. Press Ctrl + V to paste the text into the second line.

24. Move the cursor to the third line and press Ctrl + V to paste again.

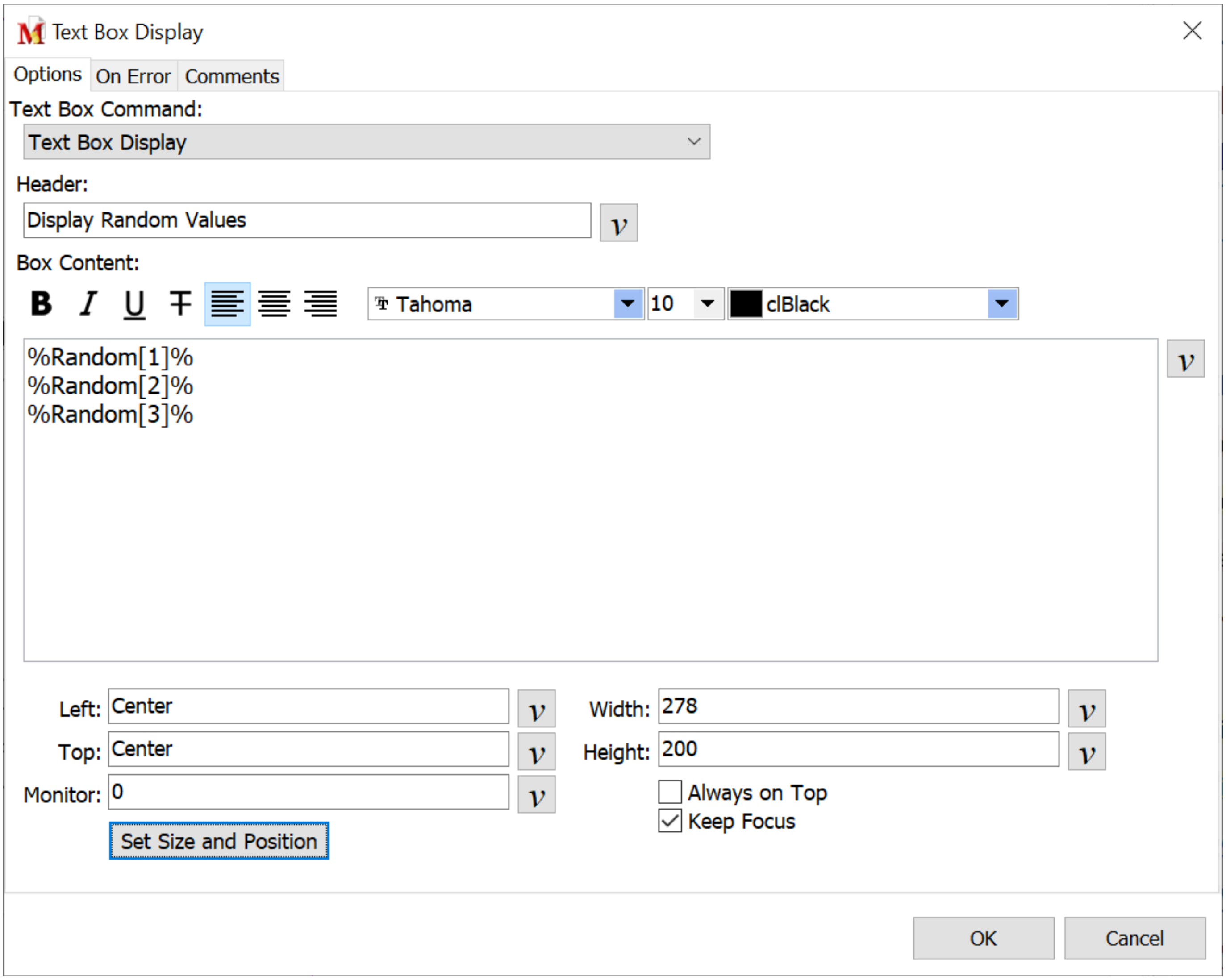
25. You should now have three identical lines of %Random[1]%.

26. In the second line, replace the 1 (one) with 2 (two) so that it reads

%Random[2]%.

27. In the third line, replace the 1 (one) with 3 (three) so that it reads %Random[3]%.

28. Click on the OK button to save and return to the Script Editor.



Text Box Command: Text Box Display

Header: Display Random Values

Box Content: Tahoma 10 clBlack

```
%Random[1]%
%Random[2]%
%Random[3]%
```

Left: Center

Top: Center

Monitor: 0

Width: 278

Height: 200

□ Always on Top

☑ Keep Focus

In the Script Editor, click on Script > Save Macro to save the macro just created.

Macro Script

The macro script should look like what is displayed below. If the macro does not look like this, arrange the commands so that they are in the following order.

```
Variable Set Integer %Random[1]% to a random value between 1 and 100
Variable Set Integer %Random[2]% to a random value between 1 and 100
Variable Set Integer %Random[3]% to a random value between 1 and 100
Text Box Display: Three Random Numbers
```

Testing the Macro

Press Ctrl + Shift + R to run the macro. A text box opens and displays three random numbers between 1 and 100.

- ✓ **Note:** If assigning original names to variables, rather than using T[1], D[1], etc., then the following characters are not allowed in the variable name: [ ] < > , . - + = \ | / * %

Advanced - ASCII Text File Process

This sample tutorial reads the contents of a text file in .csv format and inserts the data into another application. Formatting the text to one of the ASCII formats (comma delimited, tab delimited or ASCII delimited) is a quicker and more reliable way to transfer data than the copy and paste routine described in a previous tutorial. The macro reads directly from the file and inputs the results directly into the receiving application. No need to switch back and forth between applications - no copying from one program and pasting to another.

Creating a .csv file

The first step is to create a .csv file with data for the macro to read.

1. Copy the ZIP code data shown below to the clipboard. (Highlight the text in the ZIP Code Data box and press Ctrl + c.)
2. Open the notepad program. (Depending on your version of Windows, click on the Windows Start button, enter notepad in the Search field and press enter. Or right click on the Windows Start button, select Run, enter notepad and click OK.)
3. Place the cursor in the notepad program and press Ctrl + v to paste the text into notepad.
4. Click on File > Save As and make note or remember the location of where the file is about to be saved.
5. In the Name field remove .txt and type Utah ZIPS.csv.
6. Click on the Save button to save the file.

ZIP Code Data

```
84022,UT,DUGWAY,TOOELE,435,UTC -7:00,Y
84023,UT,DUTCH JOHN,DAGGETT,435,UTC -7:00,Y
84024,UT,ECHO,SUMMIT,435,UTC -7:00,Y
84025,UT,FARMINGTON,DAVIS,801,UTC -7:00,Y
84026,UT,FORT DUCHESNE,UINTAH,435,UTC -7:00,Y
84027,UT,FRUITLAND,DUCHESNE,435,UTC -7:00,Y
84028,UT,GARDEN CITY,RICH,435,UTC -7:00,Y
84029,UT,GRANTSVILLE,TOOELE,435,UTC -7:00,Y
84031,UT,HANNA,DUCHESNE,435,UTC -7:00,Y
84032,UT,HEBER CITY,WASATCH,435,UTC -7:00,Y
84033,UT,HENEFER,SUMMIT,435,UTC -7:00,Y
84034,UT,IBAPAH,TOOELE,435,UTC -7:00,Y
```

Setting up the Macro

1. From the Macro Express Pro - Explorer menu click on Macro > New Macro to open the Create New Macro window.
2. Hot Key should already be selected as the Activation.
3. Click on the HotKey edit box to place the cursor in this field.
4. Hold down the CTRL key, the SHIFT key and the Z key all at the same time and release them.
5. CTRL+SHIFT+Z should appear in the edit box. This is the hot key we will define for the macro.
6. Enter "Process ZIP Code Data" (without the quotes) in the Nickname field.
7. Click OK to open the Script Editor.
8. In the Script Editor click on the Scope tab and make sure that the macro scope is set to global.
9. Click on the Script tab to start building the macro.

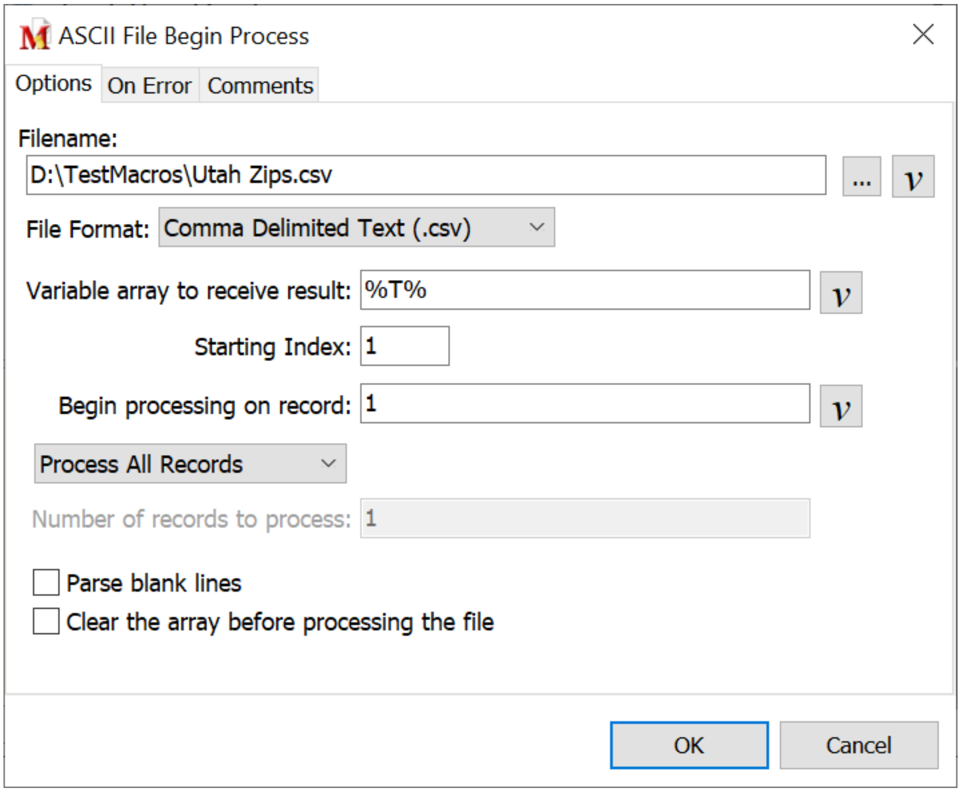
Building the Macro

Process the ASCII Text File

1. Scroll through the list of commands and expand the Files/Folders category. Double click on the ASCII File Begin Process command to open the window

of the same name.

2. Change the File Format option to Comma Delimited Text (.csv) from the drop down list.
3. Click on the Browse button to the right of the Filename field and navigate to the location of the saved .csv file.
4. Highlight the file and click on the Open button to place the name of the file and file path into the Filename edit box.
5. Place the cursor in the box next to Variable array to receive result and then click on the Variables button to open the Insert Variable window.
6. Click on the + sign next to Type: Text Variables to expand the category.
7. Click on the T to place the letter T in the Variable Name field.
8. Click OK to save and return to the ASCII File Begin Process window.
9. Leave the Starting Index set to 1.
10. Leave the Begin Processing on Record: field set to 1 in order to start processing the first line of the text file.
11. Make sure the Process All Records option is selected.
12. Click OK to save and return to the Script Editor.



Filename: D:\TestMacros\Utah Zips.csv

File Format: Comma Delimited Text (.csv)

Variable array to receive result: %T%

Starting Index: 1

Begin processing on record: 1

Process All Records

Number of records to process: 1

☐ Parse blank lines

☐ Clear the array before processing the file

✓ **Note:** The T variable is an array that by default allows up to 99 variables - T[1], T[2] and so on up to T[99]. We could also have named our own variable and created the array by clicking on the Add button in the Insert Variable window. This process is described in the Advanced - Using Integer Variables tutorial.

The window should look like the above when finished, other than the location of the Utah ZIPS.csv file.

Insert the text into another application

1. Scroll through the list of commands and expand the Keyboard category.

   Double click on the Text Type command to open a window of the same name.
2. Click on the Variables button at the bottom of the window to open the Insert Variable command.
3. Click on the + sign next to Type: Text Variables to expand this category.
4. Click on the T to place T[ ]in the Variable Name field.
5. Enter the number 1 between the brackets so that it looks like T[1].
6. Click OK to save and return to the Text Type window.
7. Highlight the %T[1]% found in the Text Type window and press Ctrl c to copy

to the clipboard.

8. Press Enter to move the cursor to the line below the %T[1] found on the first line and press Ctrl v to paste.
9. %T[1]% should now be displayed on two lines.
10. Move to line 3 and paste again.
11. Repeat the process so that %T[1]% is displayed seven times, each underneath the other.
12. On the last line, press the Enter key twice to move the cursor two lines below the last line of text.
13. Go to line 2 and change %T[1]% to %T[2]%.
14. Go to line 2 and change %T[1]% to %T[3]% and so on.
15. Click OK to save and return to the Script Editor.

End Process ASCII Text File

1. Scroll through the list of commands and expand the Files/Folders category. Double click on the ASCII File End Process command to insert the command at the end of the script.
2. Click on Script > Save Macro to save the macro just created.

Macro Script

The macro should look like the following in the Script Editor. If the macro doesn't look like the script shown below, arrange the commands so that they are in the following order. The file path in the first command most likely will be different, based on where you saved the Utah ZIPS.csv file.

```
ASCII File Begin Process: C:\Insight
Software\Utah ZIPS.csv
  Text Type (Simulate Keystrokes):
%T[1]%
  %T[2]%
  %T[4]%
  %T[5]%
  %T[6]%
  %T[7]%


ASCII File End Process
```

Playing Back the Macro

Start the WordPad program. (Depending on your version of Windows, click on the Windows Start button, enter wordpad in the Search field and press enter. Or right click on the Windows Start button, select Run, enter wordpad and click OK.) Make sure the text cursor is in the WordPad program and then press CTRL+SHIFT+Z to play back the macro.

The data is inserted in a vertical format as shown below.

```
84022
UT
DUGWAY
TOOELE
435
UTC -7:00
Y


84023
UT
DUTCH JOHN
DAGGETT
435
UTC -7:00
Y
```

- ✓ **Note:** This is a simple example of what can be done using the Process an ASCII Text File command. When entering the data into a database, choose which variables to insert and in what order, or add

Tabs to move from field to field, etc. Create macros to update the record in the database, wait for a fresh screen, insert the second line of the text file into the database, and so on.

Using Window Controls

This is an example of using Window Controls with Windows 10. The macro sends text to notepad while the program window is minimized to the task bar.

A copy of notepad needs to be open. To do this type notepad in the "Type here to search box" in the Task Bar and press Enter.

Select a Macro Activation

For this example we will create a hot key activated macro.

1. From the Macro Express - Explorer click on the New Macro icon to open the Create New Macro window.
2. HotKey should be the default Activation.
3. Click on the HotKey edit box to place the cursor in this field.
4. Press the Ctrl + Alt + 4 keys and release them. Ctrl + Alt + 4 should appear in the edit box.
5. Enter "Text to Notepad" (without the quotes) in the Macro Nickname edit field.
6. Click OK to save the changes and open the Script Editor.

Set the Macro Scope

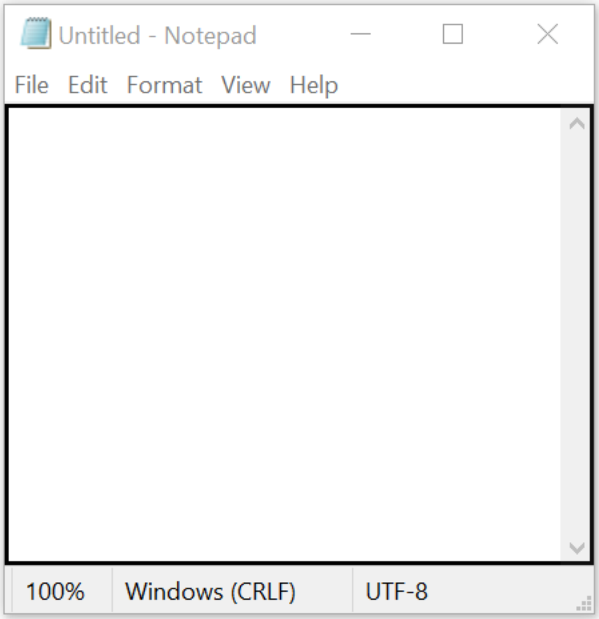
1. With the Script Editor open, click on the Scope tab.
2. Set the Current Scope to Global.

Writing the Macro

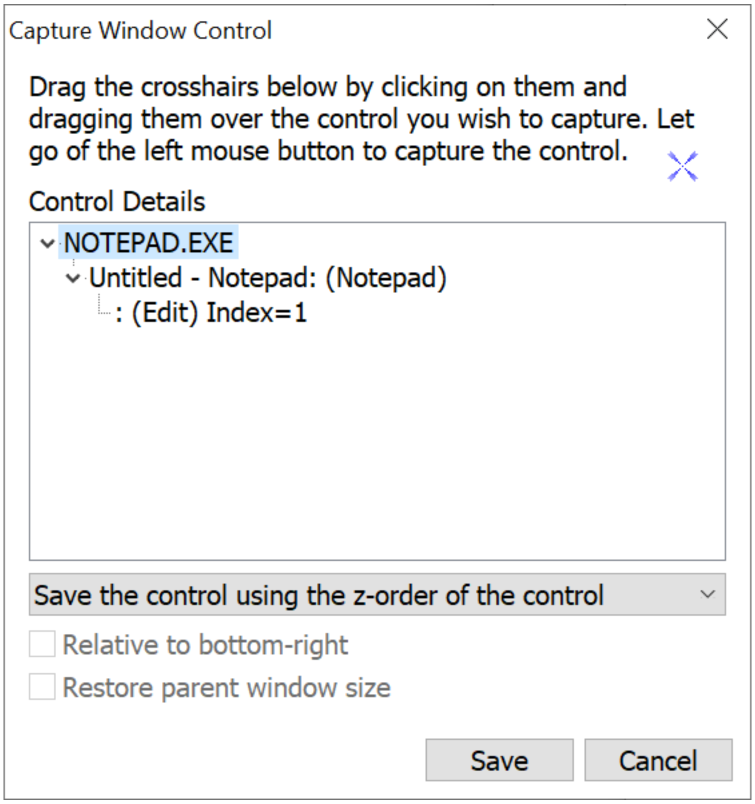
Setting the Control Variable

1. Click on the Script tab.
2. Expand the Window Controls category of commands.
3. Double click on the Get Control command to open the Get Control window.
4. Select the Hide Macro Express option.
5. Click on the Variables button to the right of the Store the Control In field.

6. Expand the Control Variables list and click on the C variable.
7. In the Variable Name field enter the number 1 between the brackets to show C[1] and click OK to save.
8. Click on the Launch Utility button to open the Capture Window Control window.
9. If notepad is not visible on the screen open the program window.
10. Click on the blue X crosshairs on the Capture Window Control window and drag it over top of the notepad edit field. It will be outlined in black as shown.



11. Release the mouse button to insert the control information into the Control Details panel.


Capture Window Control

Drag the crosshairs below by clicking on them and dragging them over the control you wish to capture. Let go of the left mouse button to capture the control.

Control Details

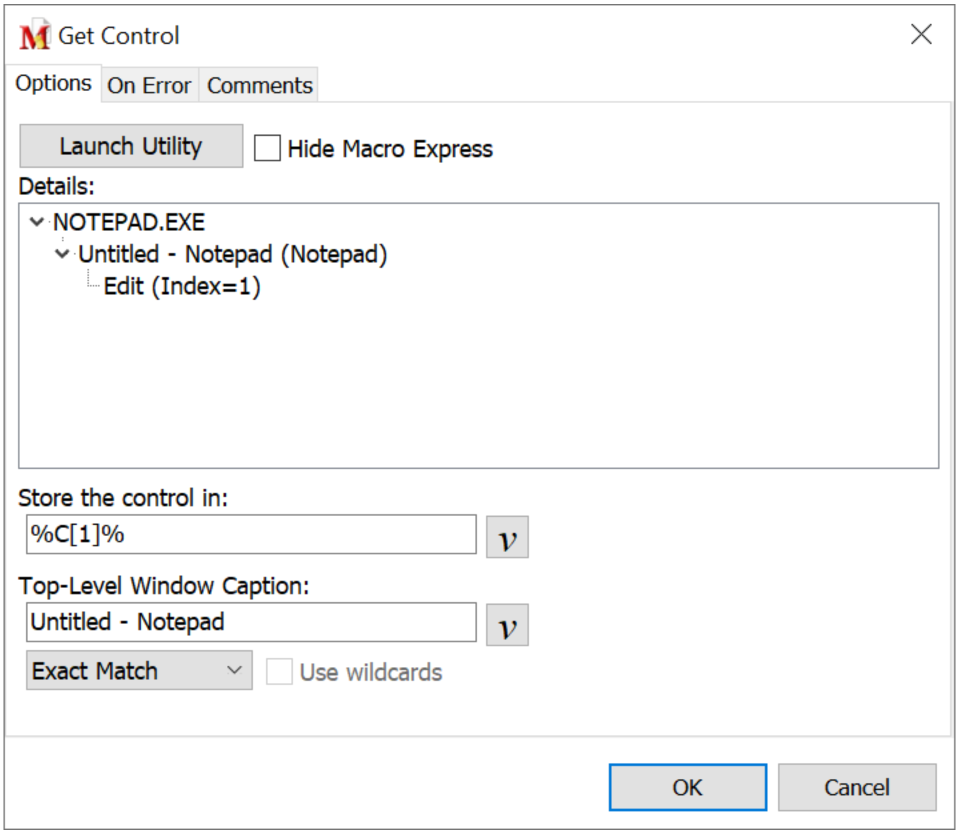
- NOTEPAD.EXE
  - Untitled - Notepad: (Notepad)
    - : (Edit) Index=1

Save the control using the z-order of the control

□ Relative to bottom-right

□ Restore parent window size

12. Click the save button to save the information and return to the Get Control Window which should now look like the following image.


Get Control

□ Hide Macro Express

Details:

- NOTEPAD.EXE
  - Untitled - Notepad (Notepad)
    - Edit (Index=1)

Store the control in: %C[1]%

Top-Level Window Caption: Untitled - Notepad

Exact Match

□ Use wildcards

13. Click OK to save and return to the Script Editor.

Sending Text to the Control

1. Expand the Keyboard Category of commands and double click on the Text Type command to open the Text Type window.
2. Enter some text in the edit field, such as This is a test.
3. Click on the Simulate Keystrokes drop down list and select Send Text Directly to Control.
4. Click on the Variables button to the right of the Control field.
5. Expand the Control Variables list and click on the C variable.
6. In the Variable Name field enter the number 1 between the brackets to show C[1] and click OK to save the variable selection.
7. Click OK in the Text Type window to save the change and return to the Script Editor.

Macro Script

When finished the macro script should look like the following example. If something is not in the correct order, highlight the command and click on the Up or Down arrows to the right of the Script and move the command to its proper location.

```
Get Control: (NOTEPAD.EXE) Using z-order -> %C[1]%
Text Type (Send Text Directly to Control %C[1]%): This is a
test.
```

Run the Macro

Make sure that notepad is running, but minimized to the task bar. Press the Ctrl + Alt + 4 hotkey. Open Notepad from the task bar. You will notice that the text has been inserted into notepad. The program window does not need to have focus in order for the text to be inserted.

Playing Back Macros

Macro Activation

Below is a chart listing the options available to activate a macro.

Activation	Description
Hot Key	A hot key is a key combination such as CTRL+ALT+A, CTRL+Right Mouse Click, etc. The macro activates when the keys are pressed and released.
Shortkey	A shortkey is a set of characters such as abc, myname, etc. The macro activates when the keys are typed into an application.
Schedule	The macro runs as a scheduled event. These events can be scheduled to play at startup of Macro Express, hourly, daily (choose any or all days of the week), weekly, monthly, one given time, every "x" minutes or after "x" minutes of inactivity.
Window Title	This option looks for a specific window title. The macro activates when the window either gains focus, loses focus, opens or closes, depending on the option selected.
Mouse Event	Use either the left, middle or right mouse buttons to activate the macro. Select an area of the screen to click on or designate a window part to click on, such as scroll bar, title bar, buttons, etc. to activate the macro.
Control Event	The Control Event option activates a macro when a specific Windows control or the window that contains the control gains focus. The control may be a window button, edit box or another identifiable portion of a window.
Directory Modification	Directory Modification looks for changes to files in a folder or to the folder itself. The macro activates when a change is made to the specified folder or to a file within that folder. Some of the available options include the macro looking for changes to file size, attributes, write time, security descriptor or checking if a file has been renamed or deleted.
System Event	Activate a macro when a System Event, such as a Windows message, resolution change or color change occurs.

Process Event	The Process Event activation method initiates a macro when a Process (an executable program or service) first starts or when it closes.
TCP/IP	This options looks for the TCP/IP stack to become available or unavailable.
Clipboard Text	Activate a macro based on the contents of the clipboard text. Select whether the clipboard does or does not match what was selected as the clipboard trigger. Or choose whether the clipboard contains or does not contain specific text.
Floating Menu	A Floating Menu is a menu of macros displayed as a floating tool bar and sits on top of other open applications. The floating menu either displays icons (representing the macros), the first part of the text of the macro or the macro nickname and scope. Clicking on one of the macros or typing a corresponding number or letter activates that macro from the floating menu.
No Activation	No Activation indicates that there is not an activation associated with the macro. In such cases, the macro may be included as a subroutine of another macro using the Macro Run command. Or the macro may be used in a popup menu of macros and no other macro activation is desired for the macro.
Popup Menu	A popup menu of macros is a list of macros displayed in a window. Clicking on one of the macros or typing a corresponding number or letter activates that macro from the popup menu. The popup menu itself is activated by either a hot key or by right clicking on a notification area icon. The popup menu may show icons (representing the macros), the first part of the text of the macro or the macro nickname and scope.
Command Line Parameters	Invoke a macro using a command line parameter. The specific command line parameters to invoke a macro are the /A and /MXE commands. Both the macexp.exe and the meproc.exe programs may be used. See the command line parameters help section for more detail.
Windows Explorer / File Explorer	Double clicking on a playable macro file (.MXE) from within Windows Explorer/File Explorer executes the macro.

Run Macro Now	Run a macro from the Macro Express - Explorer by using the Run Macro Now option. Right click on the macro in the Macro Explorer and then left click on the Run Macro Now menu item. Or highlight the macro in the Macro Explorer list and click on the Run Now button.
Windows API Calls	This option is only intended for the programmer who wants to add macro functionality directly from their own programs. While the command line parameter options can be used via another program, the following method may be much quicker and remove the overhead of having to run an external program such as macexp.exe or meproc.exe. Instruct the Macro Express Player to directly execute a macro in the currently loaded macro file by specifying its nickname or instruct it to run a playable macro. The macro nickname is case sensitive. If more than one macro shares the same nickname, only the first one found in the macro file is activated.

Windows API Examples

The following examples demonstrate the basic steps of how to activate a macro using Windows API messages and are presented in several different programming languages. These examples assume familiarity with basic Windows API commands and terminology. The process involves posting a series of messages directly to the Macro Express Player.

These examples use the message WM_USER+20 to run a macro within the currently loaded macro file. To run a playable macro, change the messages in the examples from WM_USER+20 to WM_USER+21.

Delphi Example

```
var
  i     : integer;
  S     : string;
  Hwnd  : THandle;
begin
  Hwnd := FindWindow('TMainWin', 'Macro Express Player');
  if(IsWindow(Hwnd)) then
  begin
    S := MacroName;

    for i := 1 to length(S) do
      PostMessage(Hwnd, WM_USER+20, ord(S[i]), 0);
    PostMessage(Hwnd, WM_USER+20, 0, 0);
  end
  else
    ShowMessage('Macro Express is not running');
end;
```

C/C++/Visual C Example

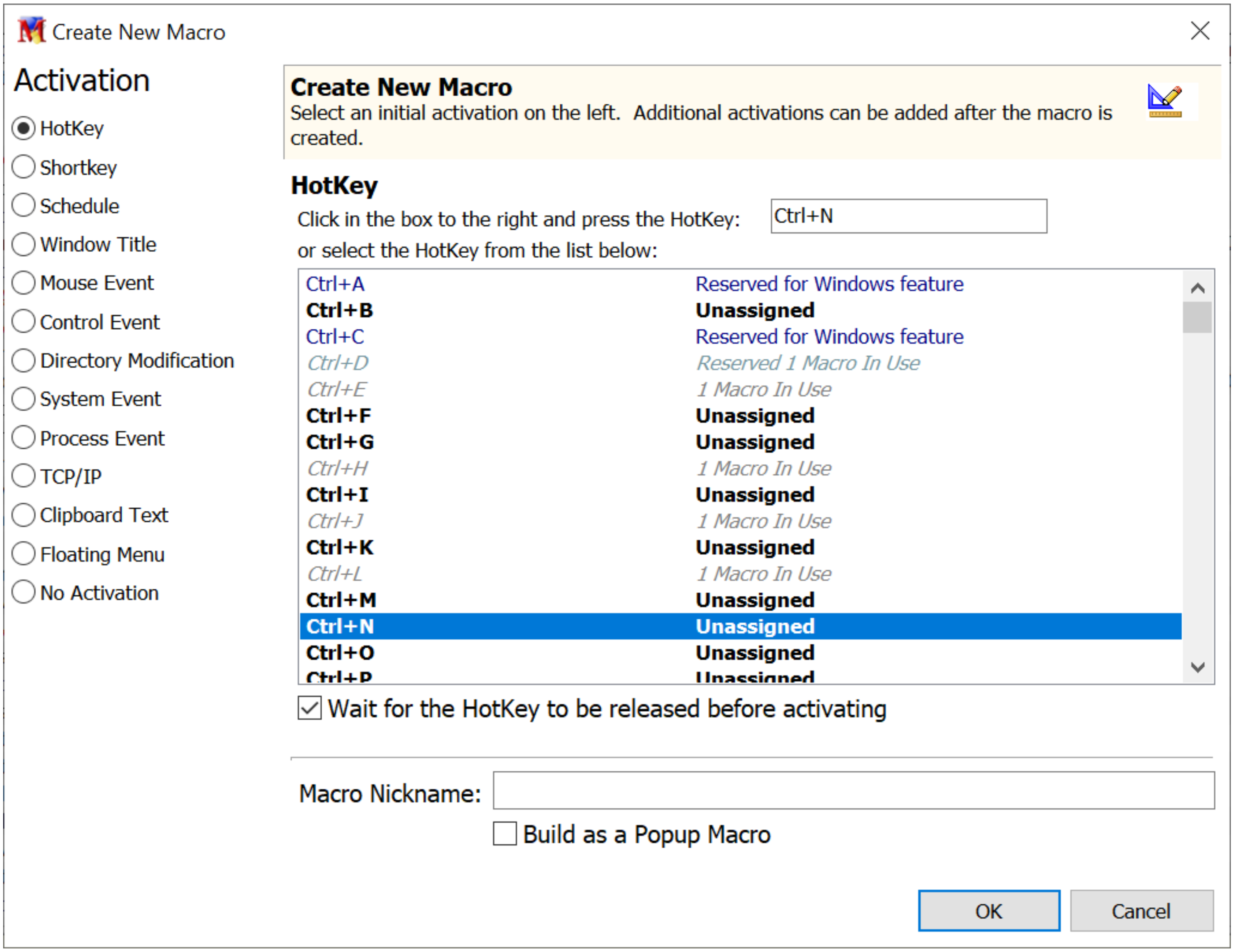
```
HWND hwnd = FindWindow("TMainWin", "Macro Express
Player");
if( IsWindow(hwnd) ) {
  for(int x = 0; x < strlen(s); x ++)
    PostMessage(hwnd, WM_USER+20, s[i], 0);
  PostMessage(hwnd, WM_USER+20, 0, 0);
}
else
  MessageBox(GetForegroundWindow(), "Macro Express is
not running", "Error", 0);
```

VBA Example

```
HWND hwnd = FindWindow("TMainWin", "Macro Express
Player");
if( IsWindow(hwnd) ) {
  for(int x = 0; x < strlen(s); x ++)
    PostMessage(hwnd, WM_USER+20, s[i], 0);
  PostMessage(hwnd, WM_USER+20, 0, 0);
}
else
  MessageBox(GetForegroundWindow(), "Macro Express is
not running", "Error", 0);


Private Declare Function FindWindow _
  Lib "User32" Alias "FindWindowA" ( _
       ByVal lpClassName As String, _
       ByVal lpWindowName As String) As Long

Private Declare Function PostMessage _
  Lib "User32" Alias "Porterage" ( _
       ByVal Wand As Long, _
       ByVal MSG As Long, _
       ByVal warm As Long, _
       ByVal Parma As Long) As Long

Private Sub Unsacred)
  'Runs a macro express macro

  Const WM_USER = &H400
  Const Command = WM_USER+20
  Const Macroname = "Macroname"

  Dim hwnd, I, Result As Long

  hwnd = FindWindow("TMainWin", "Macro Express Player")
  For I = 1 To Len(Macroname)
    Result = PostMessage(hwnd, Command,
Asc(Mid(Macroname, I, 1)), 0)
  Next I
  Result = PostMessage(hwnd, Command, 0, 0)
End Sub
```

Hot Keys

A hot key is a combination of keys that, when pressed together, plays back a macro. Macro Express allows hot key combinations ranging from a single key up to four keys pressed simultaneously.



Create New Macro

Activation

- ● HotKey
- ○ Shortkey
- ○ Schedule
- ○ Window Title
- ○ Mouse Event
- ○ Control Event
- ○ Directory Modification
- ○ System Event
- ○ Process Event
- ○ TCP/IP
- ○ Clipboard Text
- ○ Floating Menu
- ○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

HotKey

Click in the box to the right and press the HotKey: Ctrl+N

or select the HotKey from the list below:

Ctrl+A	Reserved for Windows feature
Ctrl+B	**Unassigned**
Ctrl+C	Reserved for Windows feature
Ctrl+D	*Reserved 1 Macro In Use*
Ctrl+E	*1 Macro In Use*
Ctrl+F	**Unassigned**
Ctrl+G	**Unassigned**
Ctrl+H	*1 Macro In Use*
Ctrl+I	**Unassigned**
Ctrl+J	*1 Macro In Use*
Ctrl+K	**Unassigned**
Ctrl+L	*1 Macro In Use*
Ctrl+M	**Unassigned**
Ctrl+N	**Unassigned**
Ctrl+O	**Unassigned**
Ctrl+P	**Unassigned**

☑ Wait for the HotKey to be released before activating

Macro Nickname:

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

To activate a new macro with a hot key, select the HotKey option in the Create New Macro window. Place the cursor in the edit box and hold down the hot key combination wanted. That combination is inserted in the edit box and will be the combination used to activate the macro.

If unsure which combination to use, scroll through the list of keys and highlight the one wanted.

Available Hot Keys

Hot keys designated as Reserved for Windows are used by Windows for specific functions, but may be overridden by Macro Express. For example, Ctrl + C is used to copy highlighted text to the clipboard, but the key may be assigned to Macro Express to perform a different function.

Hot Keys shown as Restricted to Windows are reserved by Windows for its own internal use. Depending on the version of Windows being used, these hot keys may not activate Macro Express macros. They will only perform the function assigned by Windows.

Wait for the Hot Key to be Released before Activating

This default setting requires that all keys in the hot key sequence be released in order for the macro to start playing. De-selecting this option allows the macro to start playing as soon as the hot key sequence is pressed.

Macro Nickname

Optionally assign a name to the macro to later help more readily determine what the macro does.

Build as a Popup Menu

Select this option to open the Menu Builder window and create a popup menu.

Hot Key Combinations

Macro Express Pro supports 846 different hot key combinations. Below is a comprehensive list of available hot keys:

Ctrl+A
Ctrl+B
Ctrl+C
Ctrl+D
Ctrl+E
Ctrl+F
Ctrl+G
Ctrl+H
Ctrl+I
Ctrl+J

Alt+Shift+U
Alt+Shift+V
Alt+Shift+W
Alt+Shift+X
Alt+Shift+Y
Alt+Shift+Z
Alt+Shift+0
Alt+Shift+1
Alt+Shift+2
Alt+Shift+3

Ctrl+`
Alt+`
Ctrl+Shift+`
Alt+Shift+`
Ctrl+Alt+`
Ctrl+Alt+Shift+`
KeyPad /
Ctrl+Keypad /
Alt+Keypad /
Shift+KeyPad /

Win+Ctrl+8
Win+Ctrl+9
Win+Ctrl+F1
Win+Ctrl+F2
Win+Ctrl+F3
Win+Ctrl+F4
Win+Ctrl+F5
Win+Ctrl+F6
Win+Ctrl+F7
Win+Ctrl+F8

Ctrl+K
Ctrl+L
Ctrl+M
Ctrl+N
Ctrl+O
Ctrl+P
Ctrl+Q
Ctrl+R
Ctrl+S
Ctrl+T
Ctrl+U
Ctrl+V
Ctrl+W
Ctrl+X
Ctrl+Y
Ctrl+Z
Ctrl+0
Ctrl+1
Ctrl+2
Ctrl+3
Ctrl+4
Ctrl+5
Ctrl+6
Ctrl+7
Ctrl+8
Ctrl+9
Ctrl+F1
Ctrl+F2
Ctrl+F3
Ctrl+F4
Ctrl+F5
Ctrl+F6
Ctrl+F7
Ctrl+F8
Ctrl+F9
Ctrl+F10
Ctrl+F11
Ctrl+F12
Alt+A
Alt+B
Alt+C
Alt+D
Alt+E
Alt+F
Alt+Shift+4
Alt+Shift+5
Alt+Shift+6
Alt+Shift+7
Alt+Shift+8
Alt+Shift+9
Alt+Shift+F1
Alt+Shift+F2
Alt+Shift+F3
Alt+Shift+F4
Alt+Shift+F5
Alt+Shift+F6
Alt+Shift+F7
Alt+Shift+F8
Alt+Shift+F9
Alt+Shift+F10
Alt+Shift+F11
Alt+Shift+F12
Ctrl+Alt+Shift+A
Ctrl+Alt+Shift+B
Ctrl+Alt+Shift+C
Ctrl+Alt+Shift+D
Ctrl+Alt+Shift+E
Ctrl+Alt+Shift+F
Ctrl+Alt+Shift+G
Ctrl+Alt+Shift+H
Ctrl+Alt+Shift+I
Ctrl+Alt+Shift+J
Ctrl+Alt+Shift+K
Ctrl+Alt+Shift+L
Ctrl+Alt+Shift+M
Ctrl+Alt+Shift+N
Ctrl+Alt+Shift+O
Ctrl+Alt+Shift+P
Ctrl+Alt+Shift+Q
Ctrl+Alt+Shift+R
Ctrl+Alt+Shift+S
Ctrl+Alt+Shift+T
Ctrl+Alt+Shift+U
Ctrl+Alt+Shift+V
Ctrl+Alt+Shift+W
Ctrl+Alt+Shift+X
Ctrl+Alt+Shift+Y
Ctrl+Alt+Shift+Z
Ctrl+Shift+Keypad /
Alt+Shift+Keypad /
Ctrl+Alt+Keypad /
Ctrl+Alt+Shift+Keypad /
KeyPad *
Ctrl+Keypad *
Alt+Keypad *
Shift+KeyPad *
Ctrl+Shift+Keypad *
Alt+Shift+Keypad *
Ctrl+Alt+Keypad *
Ctrl+Alt+Shift+Keypad *
KeyPad -
Ctrl+Keypad -
Alt+Keypad -
Shift+KeyPad -
Ctrl+Shift+Keypad -
Alt+Shift+Keypad -
Ctrl+Alt+Keypad -
Ctrl+Alt+Shift+Keypad -
KeyPad +
Ctrl+Keypad +
Alt+Keypad +
Shift+KeyPad +
Ctrl+Shift+Keypad +
Alt+Shift+Keypad +
Ctrl+Alt+Keypad +
Ctrl+Alt+Shift+Keypad +
Insert
Delete
Home
End"
Page Up
Page Down
Left Arrow
Right Arrow
Up Arrow
Down Arrow
Ctrl+Page Up
Alt+Page Up
Shift+Page Up
Ctrl+Shift+Page Up
Alt+Shift+Page Up
Ctrl+Alt+Page Up
Win+Ctrl+F9
Win+Ctrl+F10
Win+Ctrl+F11
Win+Ctrl+F12
Win+Alt+A
Win+Alt+B
Win+Alt+C
Win+Alt+D
Win+Alt+E
Win+Alt+F
Win+Alt+G
Win+Alt+H
Win+Alt+I
Win+Alt+J
Win+Alt+K
Win+Alt+L
Win+Alt+M
Win+Alt+N
Win+Alt+O
Win+Alt+P
Win+Alt+Q
Win+Alt+R
Win+Alt+S
Win+Alt+T
Win+Alt+U
Win+Alt+V
Win+Alt+W
Win+Alt+X
Win+Alt+Y
Win+Alt+Z
Win+Alt+0
Win+Alt+1
Win+Alt+2
Win+Alt+3
Win+Alt+4
Win+Alt+5
Win+Alt+6
Win+Alt+7
Win+Alt+8
Win+Alt+9
Win+Alt+F1
Win+Alt+F2
Win+Alt+F3
Win+Alt+F4

Alt+G
Alt+H
Alt+I
Alt+J
Alt+K
Alt+L
Alt+M
Alt+N
Alt+O
Alt+P
Alt+Q
Alt+R
Alt+S
Alt+T
Alt+U
Alt+V
Alt+W
Alt+X
Alt+Y
Alt+Z
Alt+0
Alt+1
Alt+2
Alt+3
Alt+4
Alt+5
Alt+6
Alt+7
Alt+8
Alt+9
Alt+F1
Alt+F2
Alt+F3
Alt+F4
Alt+F5
Alt+F6
Alt+F7
Alt+F8
Alt+F9
Alt+F10
Alt+F11
Alt+F12
Ctrl+Shift+A
Ctrl+Shift+B

Ctrl+Alt+Shift+0
Ctrl+Alt+Shift+1
Ctrl+Alt+Shift+2
Ctrl+Alt+Shift+3
Ctrl+Alt+Shift+4
Ctrl+Alt+Shift+5
Ctrl+Alt+Shift+6
Ctrl+Alt+Shift+7
Ctrl+Alt+Shift+8
Ctrl+Alt+Shift+9
Ctrl+Alt+Shift+F1
Ctrl+Alt+Shift+F2
Ctrl+Alt+Shift+F3
Ctrl+Alt+Shift+F4
Ctrl+Alt+Shift+F5
Ctrl+Alt+Shift+F6
Ctrl+Alt+Shift+F7
Ctrl+Alt+Shift+F8
Ctrl+Alt+Shift+F9
Ctrl+Alt+Shift+F10
Ctrl+Alt+Shift+F11
Ctrl+Alt+Shift+F12
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Shift+F1
Shift+F2
Shift+F3
Shift+F4
Shift+F5
Shift+F6
Shift+F7
Shift+F8
Shift+F9
Shift+F10

Ctrl+Alt+Shift+Page Up
Ctrl+Page Down
Alt+Page Down
Shift+Page Down
Ctrl+Shift+Page Down
Alt+Shift+Page Down
Ctrl+Alt+Page Down
Ctrl+Alt+Shift+Page Down
Ctrl+Left Arrow
Alt+Left Arrow
Shift+Left Arrow
Ctrl+Shift+Left Arrow
Alt+Shift+Left Arrow
Ctrl+Alt+Left Arrow
Ctrl+Alt+Shift+Left Arrow
Ctrl+Right Arrow
Alt+Right Arrow
Shift+Right Arrow
Ctrl+Shift+Right Arrow
Alt+Shift+Right Arrow
Ctrl+Alt+Right Arrow
Ctrl+Alt+Shift+Right Arrow
Ctrl+Up Arrow
Alt+Up Arrow
Shift+Up Arrow
Ctrl+Shift+Up Arrow
Alt+Shift+Up Arrow
Ctrl+Alt+Up Arrow
Ctrl+Alt+Shift+Up Arrow
Ctrl+Down Arrow
Alt+Down Arrow
Shift+Down Arrow
Ctrl+Shift+Down Arrow
Alt+Shift+Down Arrow
Ctrl+Alt+Down Arrow
Ctrl+Alt+Shift+Down Arrow
KeyPad 0
Ctrl+Keypad 0
Alt+Keypad 0
Ctrl+Alt+Keypad 0
KeyPad 1
Ctrl+Keypad 1
Alt+Keypad 1
Ctrl+Alt+Keypad 1

Win+Alt+F5
Win+Alt+F6
Win+Alt+F7
Win+Alt+F8
Win+Alt+F9
Win+Alt+F10
Win+Alt+F11
Win+Alt+F12
ESC
LMouse
Shift+LMouse
Ctrl+LMouse
Alt+LMouse
Ctrl+Shift+LMouse
Ctrl+Alt+LMouse
Shift+Alt+LMouse
RMouse
Shift+RMouse
Ctrl+RMouse
Alt+RMouse
Ctrl+Shift+RMouse
Ctrl+Alt+RMouse
Shift+Alt+RMouse
MMouse
Shift+MMouse
Ctrl+MMouse
Alt+MMouse
Ctrl+Shift+MMouse
Ctrl+Alt+MMouse
Shift+Alt+MMouse
Win+Space
Space
Enter
KeyPad .
Alt+KeyPad .
Ctrl+KeyPad .
Win+KeyPad .
Win+KeyPad 0
Win+KeyPad 1
Win+KeyPad 2
Win+KeyPad 3
Win+KeyPad 4
Win+KeyPad 5
Win+KeyPad 6

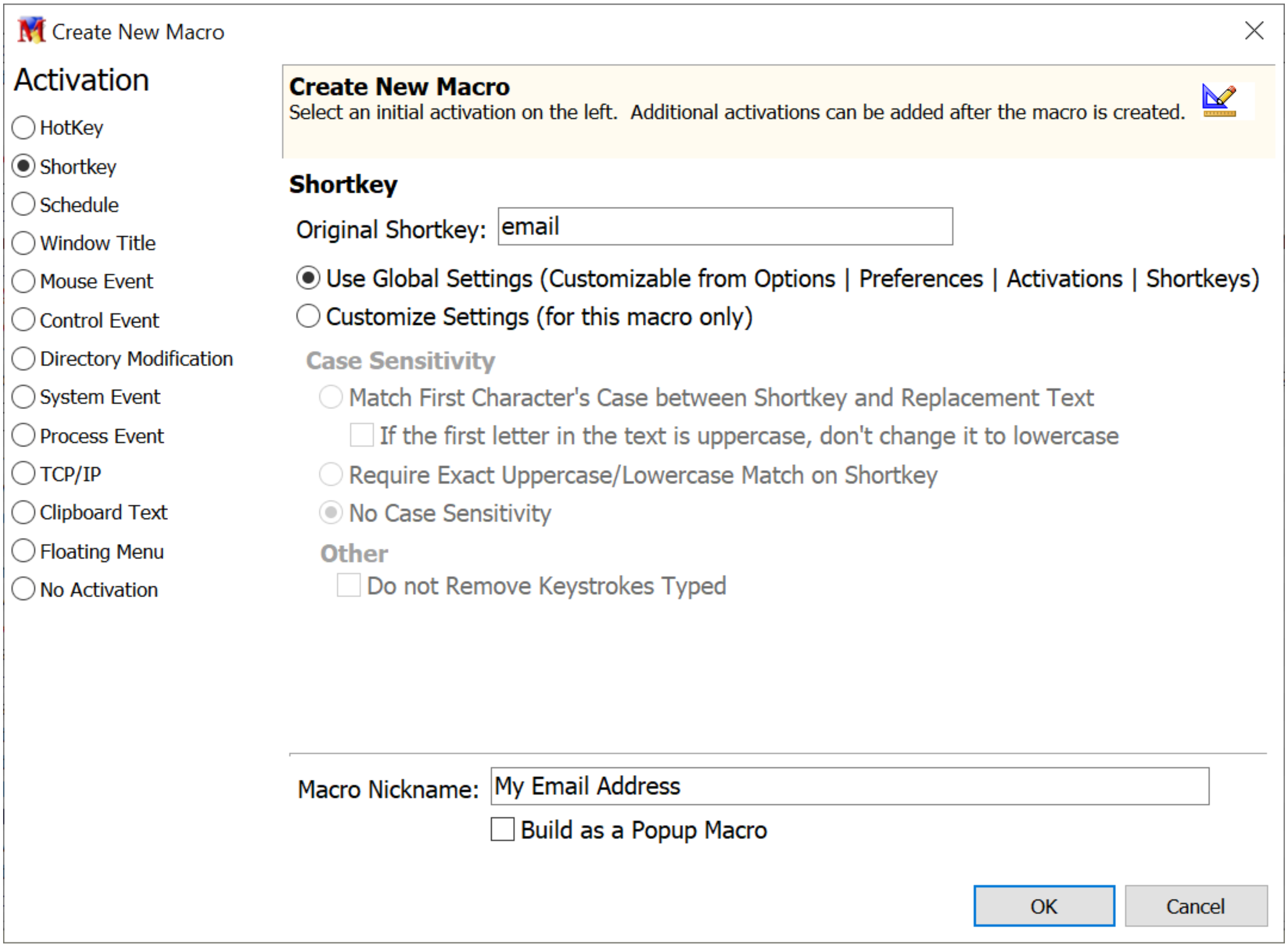
Ctrl+Shift+C
Ctrl+Shift+D
Ctrl+Shift+E
Ctrl+Shift+F
Ctrl+Shift+G
Ctrl+Shift+H
Ctrl+Shift+I
Ctrl+Shift+J
Ctrl+Shift+K
Ctrl+Shift+L
Ctrl+Shift+M
Ctrl+Shift+N
Ctrl+Shift+O
Ctrl+Shift+P
Ctrl+Shift+Q
Ctrl+Shift+R
Ctrl+Shift+S
Ctrl+Shift+T
Ctrl+Shift+U
Ctrl+Shift+V
Ctrl+Shift+W
Ctrl+Shift+X
Ctrl+Shift+Y
Ctrl+Shift+Z
Ctrl+Shift+0
Ctrl+Shift+1
Ctrl+Shift+2
Ctrl+Shift+3
Ctrl+Shift+4
Ctrl+Shift+5
Ctrl+Shift+6
Ctrl+Shift+7
Ctrl+Shift+8
Ctrl+Shift+9
Ctrl+Shift+F1
Ctrl+Shift+F2
Ctrl+Shift+F3
Ctrl+Shift+F4
Ctrl+Shift+F5
Ctrl+Shift+F6
Ctrl+Shift+F7
Ctrl+Shift+F8
Ctrl+Shift+F9
Ctrl+Shift+F10
Shift+F11
Shift+F12
Ctrl+Enter
Alt+Enter
Shift+Enter
Ctrl+Shift+Enter
Alt+Shift+Enter
Ctrl+Alt+Enter
Ctrl+Alt+Shift+Enter
Ctrl+Tab
Shift+Tab
Ctrl+Shift+Tab
Ctrl+Alt+Tab
Ctrl+Alt+Shift+Tab
Ctrl+Backspace
Alt+Backspace
Shift+Backspace
Ctrl+Shift+Backspace
Alt+Shift+Backspace
Ctrl+Alt+Backspace
Ctrl+Alt+Shift+Backspace
Ctrl+Space
Alt+Space
Shift+Space
Ctrl+Shift+Space
Alt+Shift+Space
Ctrl+Alt+Space
Ctrl+Alt+Shift+Space
Ctrl+Insert
Alt+Insert
Shift+Insert
Ctrl+Shift+Insert
Alt+Shift+Insert
Ctrl+Alt+Insert
Ctrl+Alt+Shift+Insert
Ctrl+Delete
Alt+Delete
Shift+Delete
Ctrl+Shift+Delete
Alt+Shift+Delete
Ctrl+Home
Alt+Home
Shift+Home
Ctrl+Shift+Home
KeyPad 2
Ctrl+Keypad 2
Alt+Keypad 2
Ctrl+Alt+Keypad 2
KeyPad 3
Ctrl+Keypad 3
Alt+Keypad 3
Ctrl+Alt+Keypad 3
KeyPad 4
Ctrl+Keypad 4
Alt+Keypad 4
Ctrl+Alt+Keypad 4
KeyPad 5
Ctrl+Keypad 5
Alt+Keypad 5
Ctrl+Alt+Keypad 5
KeyPad 6
Ctrl+Keypad 6
Alt+Keypad 6
Ctrl+Alt+Keypad 6
KeyPad 7
Ctrl+Keypad 7
Alt+Keypad 7
Ctrl+Alt+Keypad 7
KeyPad 8
Ctrl+Keypad 8
Alt+Keypad 8
Ctrl+Alt+Keypad 8
KeyPad 9
Ctrl+Keypad 9
Alt+Keypad 9
Ctrl+Alt+Keypad 9
Win+A
Win+B
Win+C
Win+D
Win+E
Win+F
Win+G
Win+H
Win+I
Win+J
Win+K
Win+L
Win+KeyPad 7
Win+KeyPad 8
Win+KeyPad 9
Win+KeyPad /
Win+KeyPad *
Win+KeyPad -
Win+KeyPad +
Win+Ctrl+KeyPad .
Win+Ctrl+KeyPad 0
Win+Ctrl+KeyPad 1
Win+Ctrl+KeyPad 2
Win+Ctrl+KeyPad 3
Win+Ctrl+KeyPad 4
Win+Ctrl+KeyPad 5
Win+Ctrl+KeyPad 6
Win+Ctrl+KeyPad 7
Win+Ctrl+Keypad 8
Win+Ctrl+KeyPad 9
Win+Ctrl+KeyPad /
Win+Ctrl+KeyPad *
Win+Ctrl+KeyPad -
Win+Ctrl+KeyPad +
Win+Alt+KeyPad .
Win+Alt+KeyPad 0
Win+Alt+KeyPad 1
Win+Alt+KeyPad 2
Win+Alt+KeyPad 3
Win+Alt+KeyPad 4
Win+Alt+KeyPad 5
Win+Alt+KeyPad 6
Win+Alt+KeyPad 7
Win+Alt+KeyPad 8
Win+Alt+KeyPad 9
Win+Alt+KeyPad /
Win+Alt+KeyPad *
Win+Alt+KeyPad -
Win+Alt+KeyPad +
0
1
2
3
4
5
6

Ctrl+Shift+F11
Ctrl+Shift+F12
Ctrl+Alt+A
Ctrl+Alt+B
Ctrl+Alt+C
Ctrl+Alt+D
Ctrl+Alt+E
Ctrl+Alt+F
Ctrl+Alt+G
Ctrl+Alt+H
Ctrl+Alt+I
Ctrl+Alt+J
Ctrl+Alt+K
Ctrl+Alt+L
Ctrl+Alt+M
Ctrl+Alt+N
Ctrl+Alt+O
Ctrl+Alt+P
Ctrl+Alt+Q
Ctrl+Alt+R
Ctrl+Alt+S
Ctrl+Alt+T
Ctrl+Alt+U
Ctrl+Alt+V
Ctrl+Alt+W
Ctrl+Alt+X
Ctrl+Alt+Y
Ctrl+Alt+Z
Ctrl+Alt+0
Ctrl+Alt+1
Ctrl+Alt+2
Ctrl+Alt+3
Ctrl+Alt+4
Ctrl+Alt+5
Ctrl+Alt+6
Ctrl+Alt+7
Ctrl+Alt+8
Ctrl+Alt+9
Ctrl+Alt+F1
Ctrl+Alt+F2
Ctrl+Alt+F3
Ctrl+Alt+F4
Ctrl+Alt+F5
Ctrl+Alt+F6
Alt+Shift+Home
Ctrl+Alt+Home
Ctrl+Alt+Shift+Home
Ctrl+End
Alt+End
Shift+End
Ctrl+Shift+End
Alt+Shift+End
Ctrl+Alt+End
Ctrl+Alt+Shift+End
Ctrl+.
Alt+.
Ctrl+Shift+.
Alt+Shift+.
Ctrl+Alt+.
Ctrl+Alt+Shift+.
Ctrl+,
Alt+,
Ctrl+Shift+,
Alt+Shift+,
Ctrl+Alt+,
Ctrl+Alt+Shift+,
Ctrl+/
Alt+/
Ctrl+Shift+/
Alt+Shift+/
Ctrl+Alt+/
Ctrl+Alt+Shift+/
Ctrl+;
Alt+;
Ctrl+Shift+;
Alt+Shift+;
Ctrl+Alt+;
Ctrl+Alt+Shift+;
Ctrl+"
Alt+"
Ctrl+Shift+"
Alt+Shift+"
Ctrl+Alt+"
Ctrl+Alt+Shift+"
Ctrl+\
Alt+\
Ctrl+Shift+\
Alt+Shift+\
Win+M
Win+N
Win+O
Win+P
Win+Q
Win+R
Win+S
Win+T
Win+U
Win+V
Win+W
Win+X
Win+Y
Win+Z
Win+0
Win+1
Win+2
Win+3
Win+4
Win+5
Win+6
Win+7
Win+8
Win+9
Win+F1
Win+F2
Win+F3
Win+F4
Win+F5
Win+F6
Win+F7
Win+F8
Win+F9
Win+F10
Win+F11
Win+F12
Win+Ctrl+A
Win+Ctrl+B
Win+Ctrl+C
Win+Ctrl+D
Win+Ctrl+E
Win+Ctrl+F
Win+Ctrl+G
Win+Ctrl+H
7
8
9
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
Shift+0
Shift+1
Shift+2
Shift+3
Shift+4
Shift+5
Shift+6
Shift+7
Shift+8
Shift+9
Shift+A
Shift+B
Shift+C
Shift+D
Shift+E

Ctrl+Alt+F7
Ctrl+Alt+F8
Ctrl+Alt+F9
Ctrl+Alt+F10
Ctrl+Alt+F11
Ctrl+Alt+F12
Alt+Shift+A
Alt+Shift+B
Alt+Shift+C
Alt+Shift+D
Alt+Shift+E
Alt+Shift+F
Alt+Shift+G
Alt+Shift+H
Alt+Shift+I
Alt+Shift+J
Alt+Shift+K
Alt+Shift+L
Alt+Shift+M
Alt+Shift+N
Alt+Shift+O
Alt+Shift+P
Alt+Shift+Q
Alt+Shift+R
Alt+Shift+S
Alt+Shift+T
Ctrl+Alt+\
Ctrl+Alt+Shift+\
Ctrl+[
Alt+[
Ctrl+Shift+[
Alt+Shift+[
Ctrl+Alt+[
Ctrl+Alt+Shift+[
Ctrl+]
Alt+]
Ctrl+Shift+]
Alt+Shift+]
Ctrl+Alt+]
Ctrl+Alt+Shift+]
Ctrl+-
Alt+-
Ctrl+Shift+-
Alt+Shift+-
Ctrl+Alt+-
Ctrl+Alt+Shift+-
Ctrl+=
Alt+=
Ctrl+Shift+=
Alt+Shift+=
Ctrl+Alt+=
Ctrl+Alt+Shift+=
Win+Ctrl+I
Win+Ctrl+J
Win+Ctrl+K
Win+Ctrl+L
Win+Ctrl+M
Win+Ctrl+N
Win+Ctrl+O
Win+Ctrl+P
Win+Ctrl+Q
Win+Ctrl+R
Win+Ctrl+S
Win+Ctrl+T
Win+Ctrl+U
Win+Ctrl+V
Win+Ctrl+W
Win+Ctrl+X
Win+Ctrl+Y
Win+Ctrl+Z
Win+Ctrl+0
Win+Ctrl+1
Win+Ctrl+2
Win+Ctrl+3
Win+Ctrl+4
Win+Ctrl+5
Win+Ctrl+6
Win+Ctrl+7
Shift+F
Shift+G
Shift+H
Shift+I
Shift+J
Shift+K
Shift+L
Shift+M
Shift+N
Shift+O
Shift+P
Shift+Q
Shift+R
Shift+S
Shift+T
Shift+U
Shift+V
Shift+W
Shift+X
Shift+Y
Shift+Z
Backspace
;
'

Shortkey

A shortkey is a set of characters such as abc, myname, etc. When the characters are typed into an application, the macro runs. To create a new macro with a shortkey, select the Shortkey option to display the shortkey edit box. In this field enter a series of characters from two to 32 characters in length.



Create New Macro

Activation

○ HotKey
● Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Shortkey

Original Shortkey: email

● Use Global Settings (Customizable from Options | Preferences | Activations | Shortkeys)
○ Customize Settings (for this macro only)

Case Sensitivity

○ Match First Character's Case between Shortkey and Replacement Text
□ If the first letter in the text is uppercase, don't change it to lowercase
○ Require Exact Uppercase/Lowercase Match on Shortkey
● No Case Sensitivity

Other

□ Do not Remove Keystrokes Typed

Macro Nickname: My Email Address

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

✓ **Note:** Before using the Shortkey option be sure to first define the shortkey default settings. Click on Options > Preferences > Activations - Shortkeys tab to customize the settings.

Shortkey Specifications

Up to 32 characters may be entered as an original shortkey. In addition, up to 3 Prefix Keys may optionally be used and are applied to all original shortkeys created. The following Prefix characters may be entered and used:

- The letters A through Z (both upper and lower case)

- The numbers 0 through 9
- The following characters: ! @ # $ % ^ & _ = < > ? . ; , * - / + ` ' ")] and space bar

Alternatively use the suffix key option and press the space bar, the enter key or a suffix character of . , : ; ? ! or - after typing the shortkey to activate the macro.

Macro Nickname

Optionally assign a name to the macro to more readily determine what the macro does.

Build as a Popup Menu

Select this option to open the Menu Builder window and create a popup menu.

Schedule

Schedule Overview

Scheduled Macros are macros that can be scheduled to play back at a specific time. Just set the schedule for the macro and it will play back unattended at the designated time. No need to press a hot key or click on an icon. The macro plays back at the selected time. In order for the macro to execute, Macro Express must be running in the background.

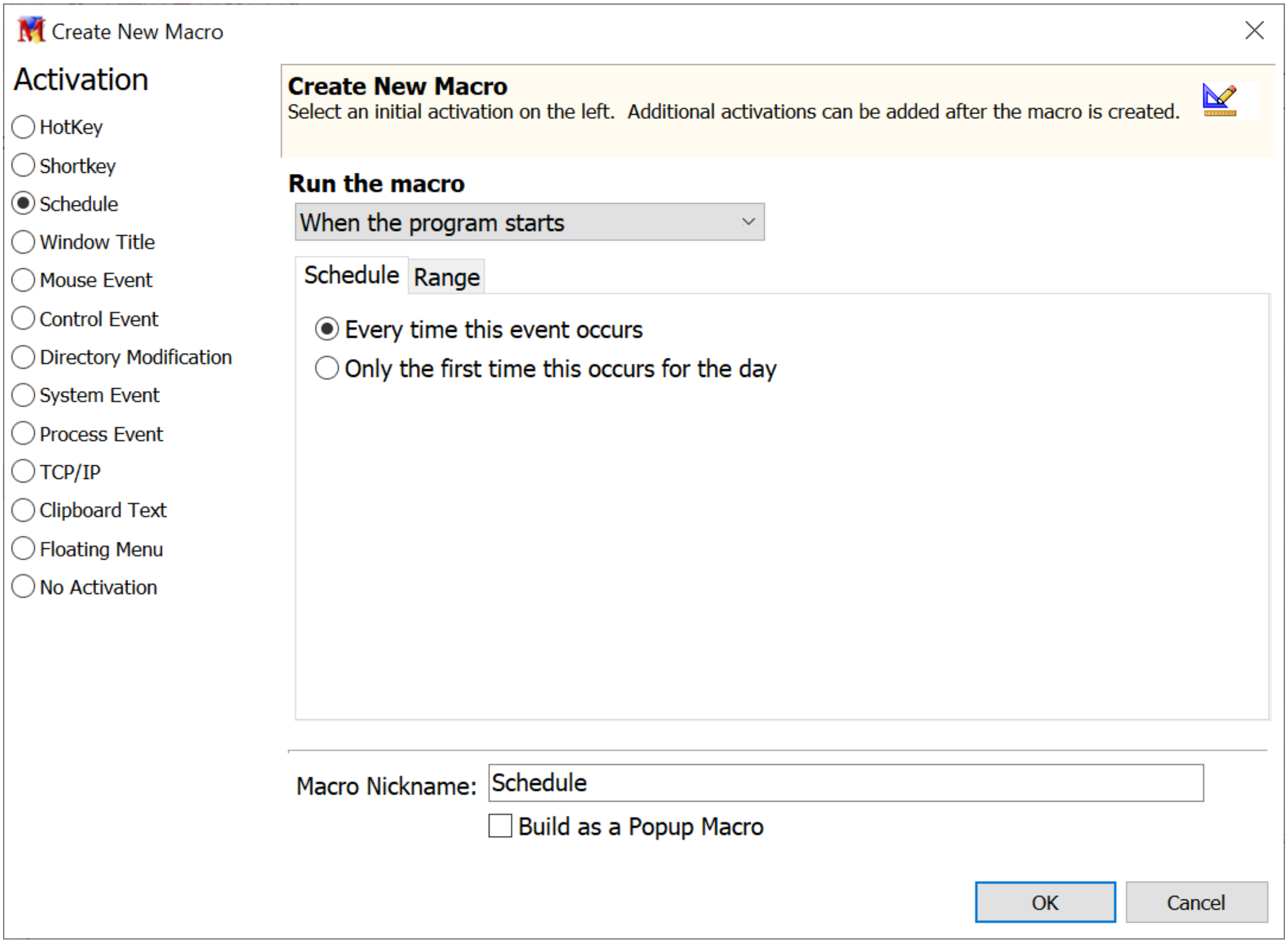
There are a number of scheduling options available. These are briefly described below. Click on the links for more detailed information.

Schedule	Description
At Startup	At Startup instructs Macro Express to immediately execute the mac Express first starts up. If Macro Express is in the Start Up group, th would play back when Windows first starts.
Once	Running a macro Once plays the macro back on one specific date ar
Hourly	An Hourly macro runs the macro repeatedly at a specific time each
Daily	A Daily macro runs the macro once each day on the days of the wee
Weekly	A Weekly macro runs once a week at the time specified.
Monthly	A Monthly macro runs at a specific time on the same day of the mor
System Idle	Setting the schedule with the System Idle option plays back the ma has been idle (no keyboard or mouse activity) for the number of mi option allows up to 720 minutes after the computer has been idled.
When Specified Time has Elapsed	This option repetitively runs a macro every XX number of minutes c intervals can be between 1 second and 99,999 minutes, 59 seconds scheduled to run every 1 second and the macro takes 5 seconds to function will not operate properly. If planning to use macros that will run more frequently than every 1 change the Scheduler Frequency default setting. The default instruc check every 10 seconds to see if a timed macro is ready to play bac be reduced in order for scheduled macros to play back more freque
When Macro File Loaded	This option instructs the macro to play back when the macro file cor first loaded.

- ✓ **Note:** Screen Savers prevent a Scheduled Macro from operating correctly. Focus cannot be given to a program if a screen saver is running. Also, the security measures built into the "Lock Workstation" function of Windows prevent Macro Express from executing a macro.

When the Program Starts

Running a macro At Startup instructs Macro Express to immediately execute the macro when Macro Express first starts up. If Macro Express is placed in the Start Up folder, the macro plays back as soon as Windows starts and loads Macro Express.



Create New Macro

Activation

○ HotKey
○ Shortkey
● Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro: When the program starts

Schedule | Range

● Every time this event occurs
○ Only the first time this occurs for the day

Macro Nickname: Schedule

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Select the preferred schedule option.

Every Time this Event Occurs

This option plays back the scheduled macro every time Macro Express is started.

Only the First Time this Occurs for the Day

This option plays back the scheduled macro the first time Macro Express is started for the day. If the program is restarted, the macro does not play back.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

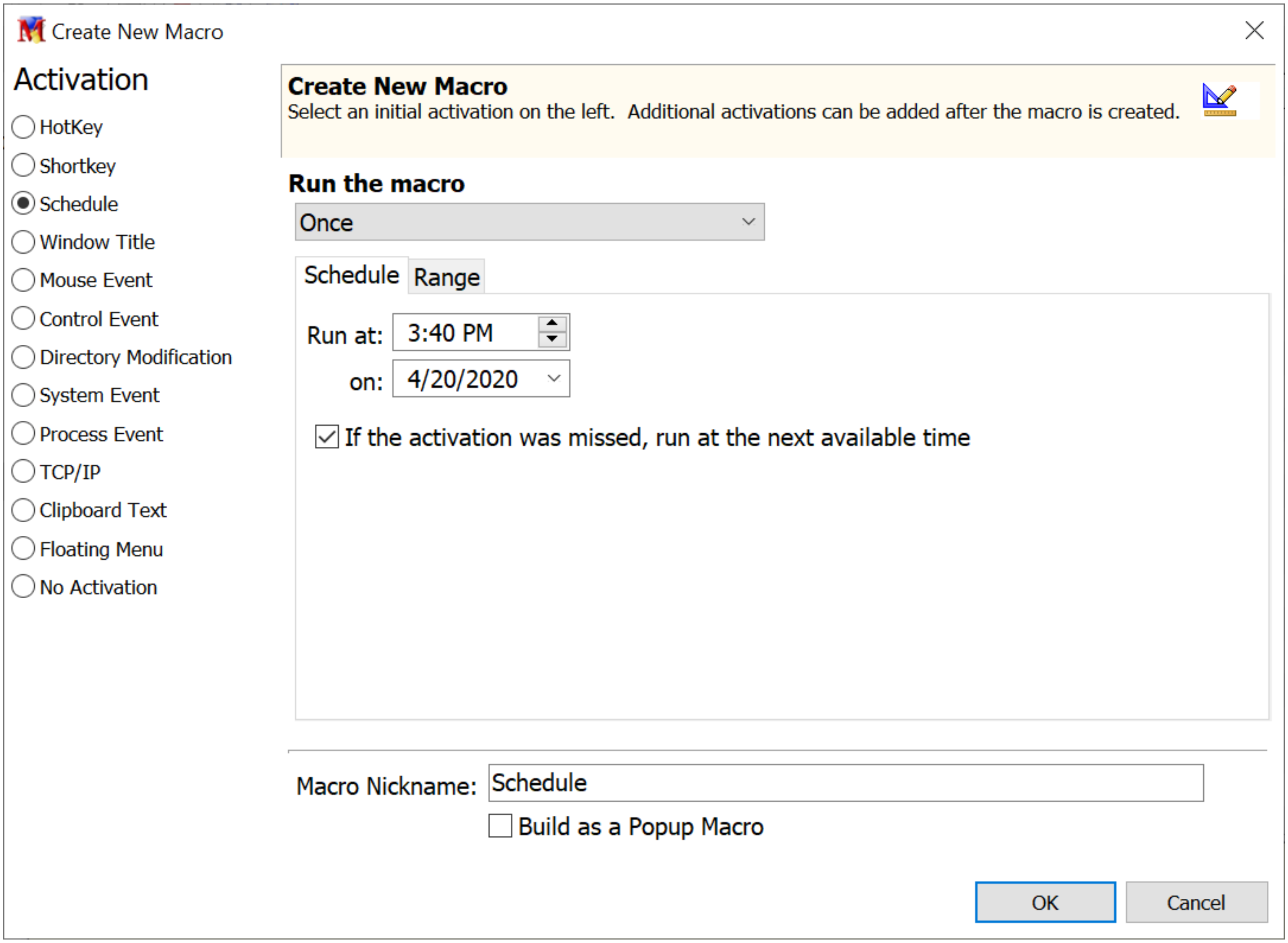
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

Once

Use the Once schedule to play only once at one specific date and time. The macro will not play again, unless the schedule is reset.



Create New Macro

Activation

○ HotKey
○ Shortkey
● Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro
Once

Schedule | Range

Run at: 3:40 PM
on: 4/20/2020

☑ If the activation was missed, run at the next available time

Macro Nickname: Schedule

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Select the Date and Time for the macro to run. Once the macro is run the first time, it will not run again.

If the activation was missed, run at the next available time

If the computer was shut down or Macro Express was closed at the scheduled macro run, this option runs the macro later when the computer is started or Macro Express is reopened.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

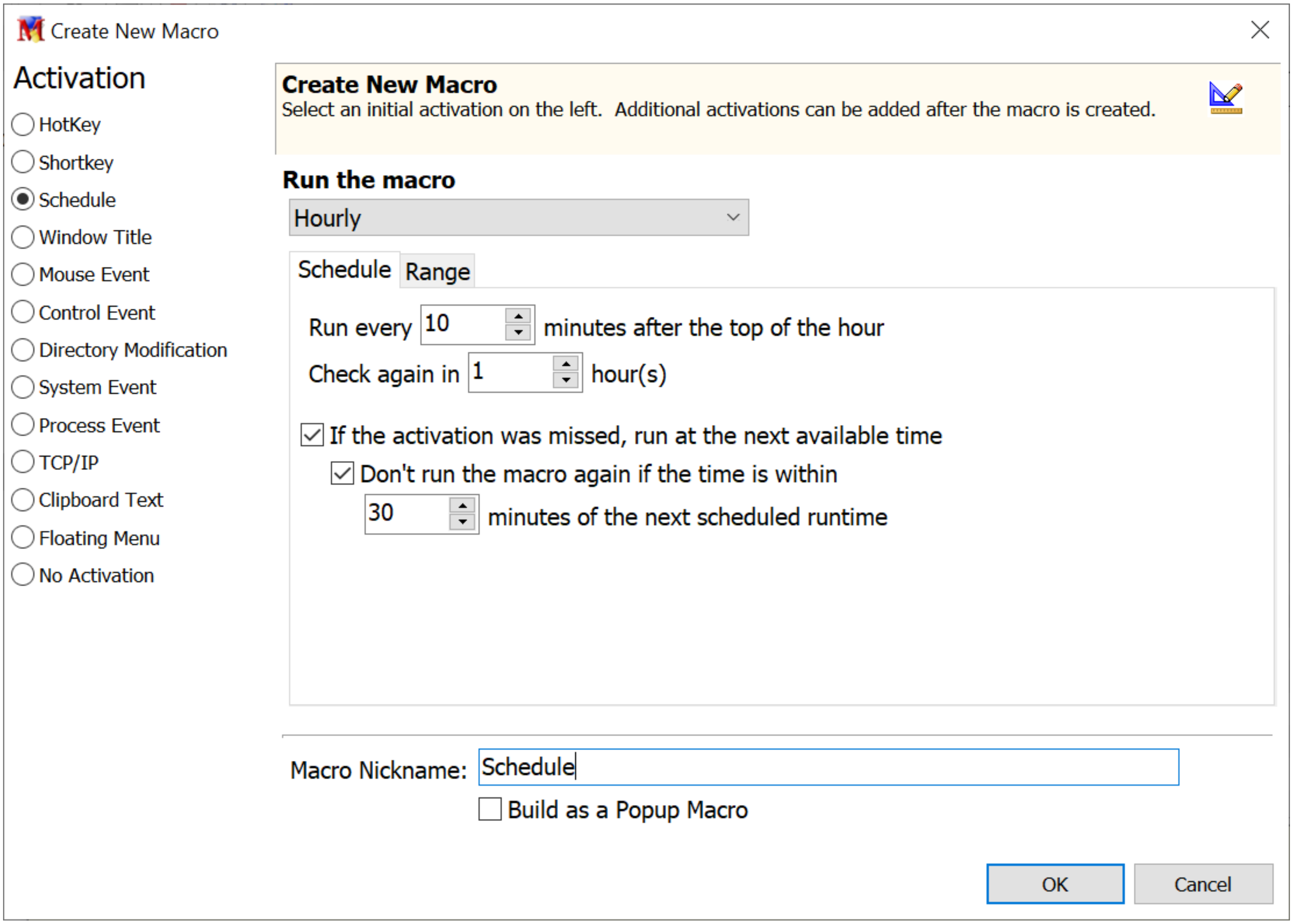
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

Hourly

The Hourly schedule runs a macro at a specific time each hour or every xx number of hours.



Create New Macro

Activation

○ HotKey
○ Shortkey
● Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro

Hourly

Schedule | Range

Run every 10 minutes after the top of the hour
Check again in 1 hour(s)

☑ If the activation was missed, run at the next available time
☑ Don't run the macro again if the time is within
30 minutes of the next scheduled runtime

Macro Nickname: Schedule
□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Run Every [ ] Minutes After the Top of the Hour

Enter the number of minutes after the hour when the macro is set to start. If the macro is to start at 12:15, 1:15, 2:15, etc., then insert 15 into the edit field.

Check Again in [ ] Hours

The default value is one, which means that the macro runs every hour. Insert the number 2 for the macro to run every two hours, 3 for every three hours, etc.

If the activation was missed, run at the next available time

If the computer was shut down or Macro Express was closed at the scheduled macro run, this option runs the macro later when the computer is started or Macro Express is reopened.

Don't Run the Macro Again if the Time is Within [ ] Minutes of the Next Scheduled Runtime

If the scheduled activation was missed, it may not be desirable to run the macro twice within a short period of time, such as 12:40 and 1:00. Insert a value such as 30 in this field. For example, if the computer restarts at 12:40, the macro does not run because the time is within 30 minutes of the next scheduled macro at 1:00.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

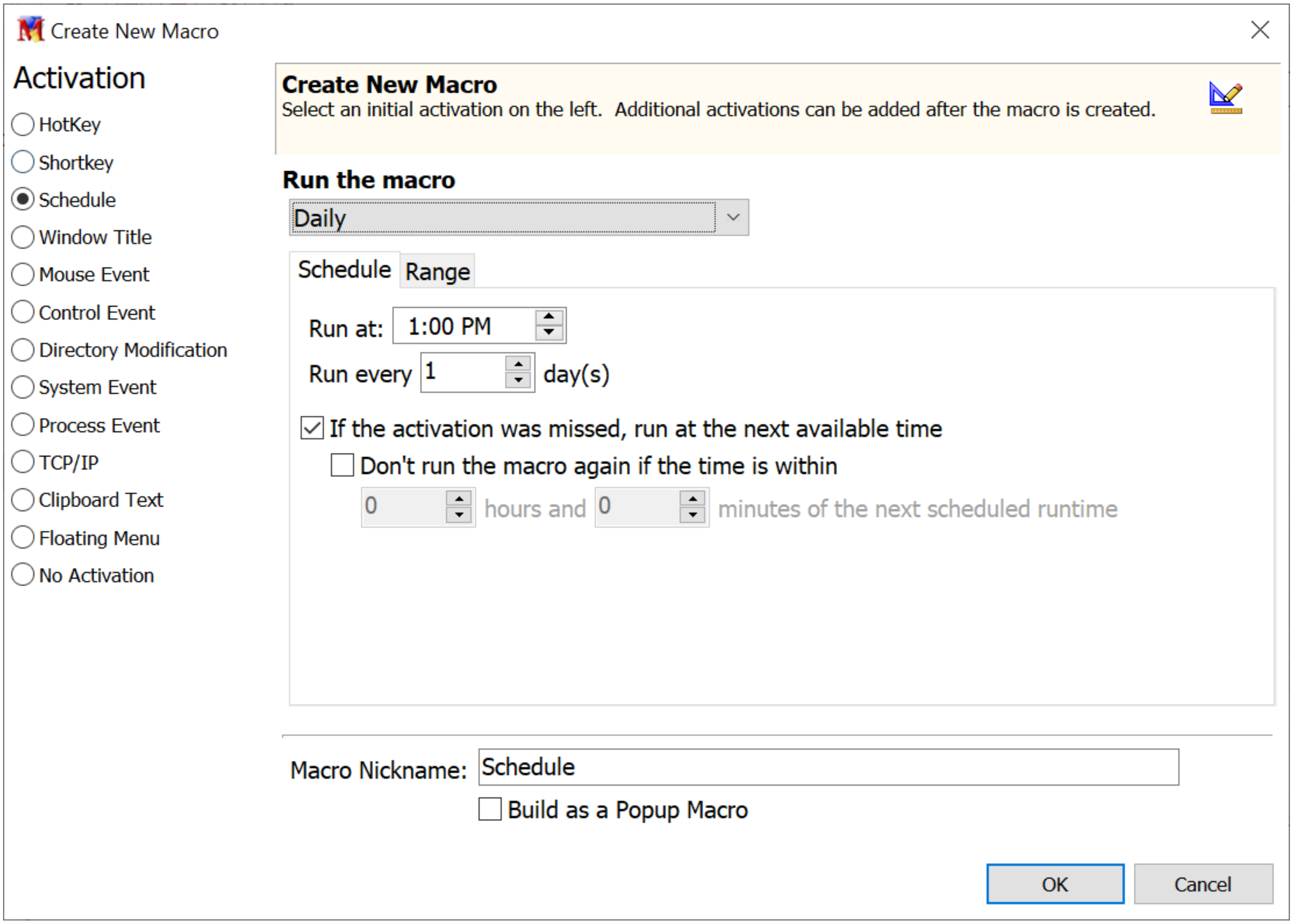
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

Daily

The Daily schedule runs a macro once a day at the time specified.



Create New Macro

Activation

○ HotKey
○ Shortkey
● Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro
Daily

Schedule | Range

Run at: 1:00 PM

Run every 1 day(s)

☑ If the activation was missed, run at the next available time

□ Don't run the macro again if the time is within

0 hours and 0 minutes of the next scheduled runtime

Macro Nickname: Schedule

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Run At

Enter the time that the macro is to run each day.

Run Every [   ] Days

The default value is one, which means that the macro runs every day. Insert the number 2 for the macro to run every two days, 3 for every three days, etc.

If the activation was missed, run at the next available time

If the computer was shut down or Macro Express was closed at the scheduled macro run, this option runs the macro later when the computer is started or Macro Express is reopened.

Don't Run the Macro Again if the Time is Within [ ] Hours and [ ] Minutes of the Next Scheduled Runtime

If the scheduled activation was missed, it may not be desirable to run the macro twice within a relatively short period of time. For example, let's say that the value is set to 12 hours, the macro is scheduled to run at noon and the computer is shut down before noon. If the computer is not restarted until 6:00am the following day, the skipped macro does not run because the time is within 12 hours of the next scheduled playback at noon.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

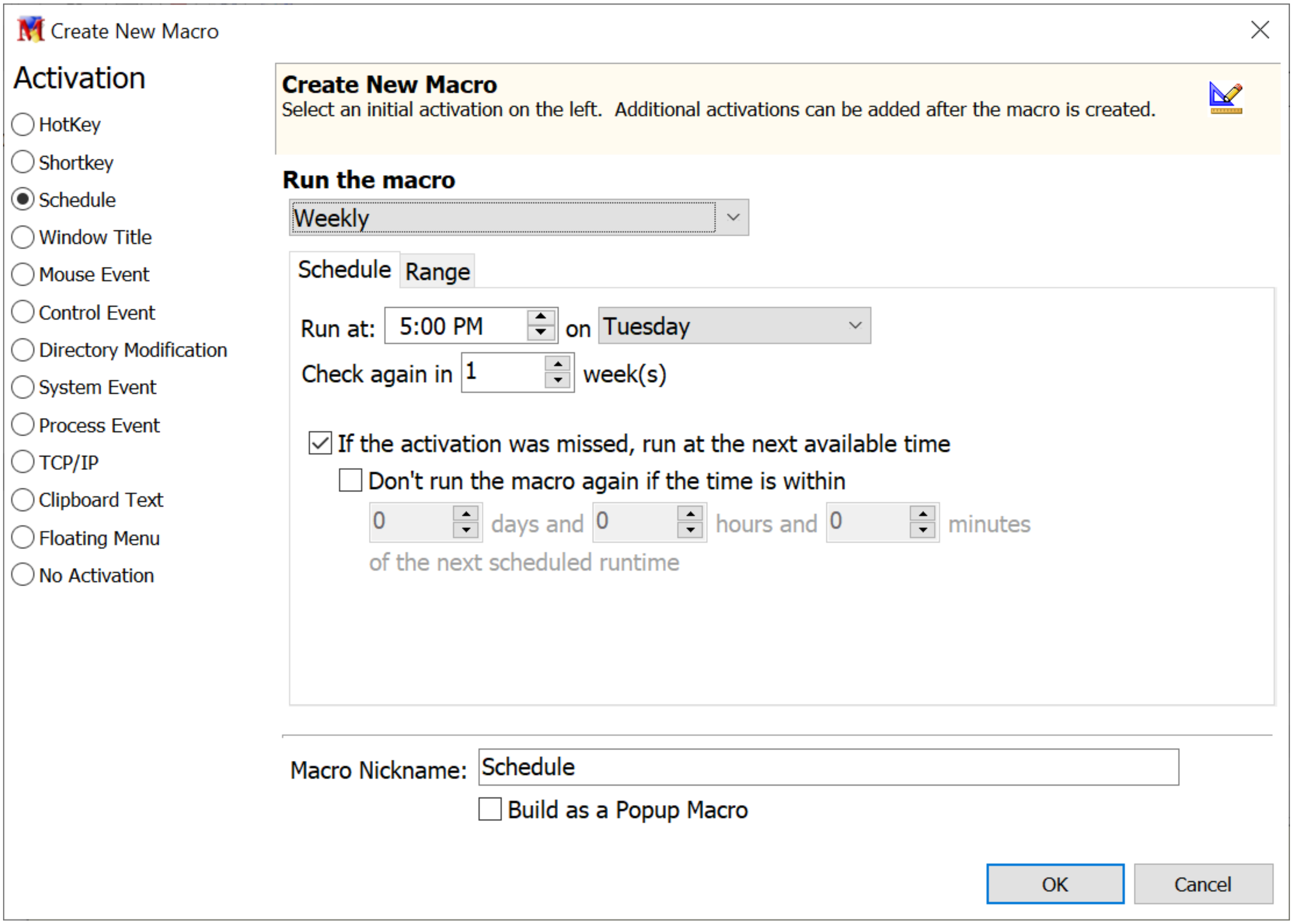
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

Weekly

The Weekly schedule runs a macro once a week at the time specified.



Create New Macro

Activation

○ HotKey
○ Shortkey
● Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro
Weekly

Schedule | Range

Run at: 5:00 PM on Tuesday
Check again in 1 week(s)

☑ If the activation was missed, run at the next available time
□ Don't run the macro again if the time is within
0 days and 0 hours and 0 minutes
of the next scheduled runtime

Macro Nickname: Schedule
□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Run At

Enter the time of day and the day of the week that the macro is to run.

Check Again in [ ] Weeks

The default value is one, which means that the macro plays every week. Insert the number "2" to have the macro run every two weeks, "3" for every three weeks, etc.

If the activation was missed, run at the next available time

If the computer was shut down or Macro Express was closed at the scheduled macro run, this option runs the macro later when the computer is started or Macro Express is reopened.

Don't Run the Macro Again if the Time is Within [ ] Days and [ ] Hours and [ ] Minutes of the Next Scheduled Runtime

If the scheduled activation was missed, it may not be desirable to run the macro twice within a relatively short period of time. For example, insert a value of "2' in the days field. If the computer is restarted on Monday after being on vacation for a week and the macro is scheduled to run Tuesday, the skipped macro from the prior week does not run because the date is within 2 days of the next scheduled macro set for Tuesday.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

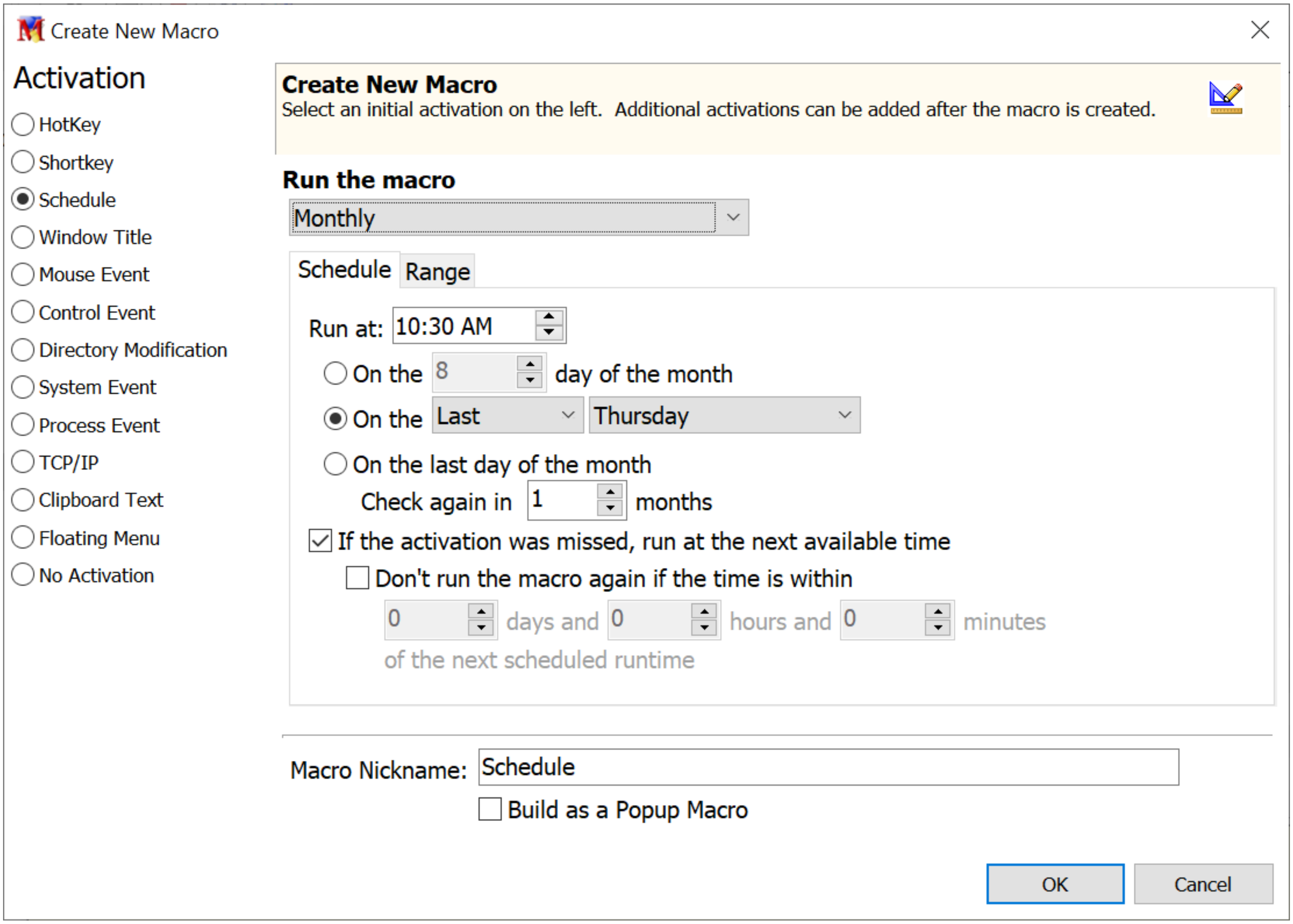
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

Monthly

This schedule runs a macro once a month at the selected time.



Create New Macro

Activation

- ○ HotKey
- ○ Shortkey
- ● Schedule
- ○ Window Title
- ○ Mouse Event
- ○ Control Event
- ○ Directory Modification
- ○ System Event
- ○ Process Event
- ○ TCP/IP
- ○ Clipboard Text
- ○ Floating Menu
- ○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro
Monthly

Schedule | Range

Run at: 10:30 AM

○ On the 8 day of the month

● On the Last Thursday

○ On the last day of the month

Check again in 1 months

☑ If the activation was missed, run at the next available time

□ Don't run the macro again if the time is within

0 days and 0 hours and 0 minutes

of the next scheduled runtime

Macro Nickname: Schedule

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Run At

To set the monthly schedule first enter the time of day that the macro is to play back. Then select from one of the three monthly date options explained below.

On the [ ] Day of the Month

Select a day of the month to have the macro run, selecting from day 1 - 31.

✓ **Note:** If the 31st day of the month is chosen, the macro will not run during the months that only have 28 or 30 days. Select the Last Day of the Month option instead.

On the First - Last

Use this option to select the first, second, third, fourth, or last day of any day of the week. For example choose to have the macro run on the second Tuesday of each month, the third Friday of the month, etc.

On the Last Day of the Month

Use this option to schedule the macro to run the last day of the month.

Check Again in [ ] Months

The default value is one, which means that the macro is scheduled to run every month. Insert the number "2" to have the macro run every two months, "3" for every three months, etc.

If the activation was missed, run at the next available time

If the computer was shut down or Macro Express was closed at the scheduled macro run, this option runs the macro later when the computer is started or Macro Express is reopened.

Don't Run the Macro Again if the Time is Within [ ] Days and [ ] Hours and [ ] Minutes of the Next Scheduled Runtime

If the scheduled activation was missed, it may not be desirable to run the macro twice within a relatively short period of time. If this is the case, then insert a value such as 10 in the days field. If the computer is restarted on the second Monday after being on vacation and the macro is scheduled to run on the third Tuesday, the skipped macro will not run because the time is within 10 days of the next scheduled macro set for Tuesday.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

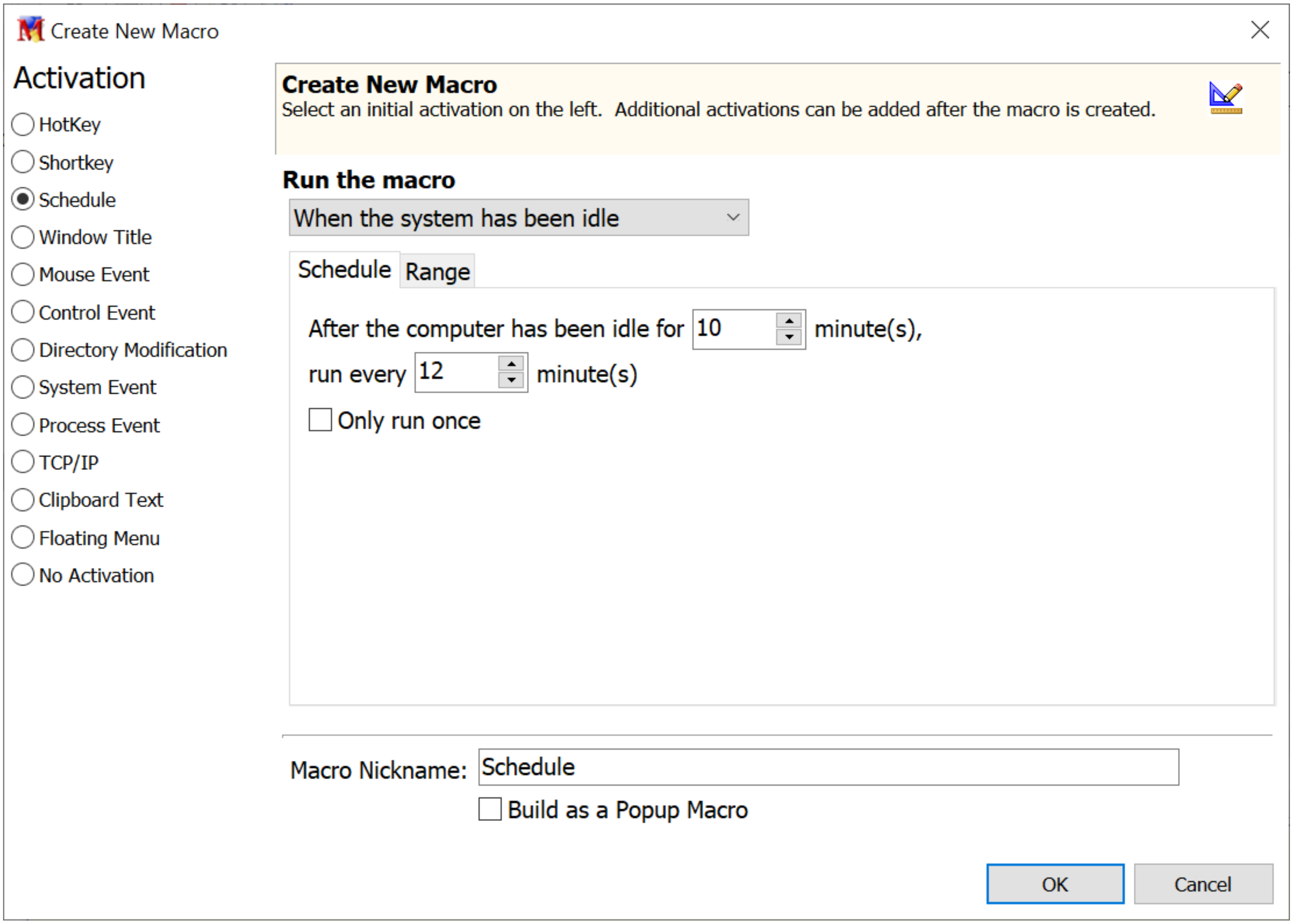
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

System Idle

Setting the schedule with the When the System Has Been Idle option plays back the macro after the computer has been idle (no keyboard or mouse activity) for the number of minutes specified. This option allows for up to 720 minutes after the computer has been idled.



Create New Macro

Activation

○ HotKey
○ Shortkey
● Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro

When the system has been idle

Schedule | Range

After the computer has been idle for 10 minute(s),
run every 12 minute(s)
□ Only run once

Macro Nickname: Schedule
□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

After the Computer has been Idle for [ ] Minutes

Insert the number of minutes the computer must be idle (no keyboard or mouse activity) before the macro starts.

Run Every [ ] Minutes

This option plays back the macro every X number of minutes if the macro does not use the keyboard or mouse to change the idle state of the computer.

Example

In the image above, the scheduler waits for the system to be idle for 10 minutes. The macro waits 12 minutes before running the macro.

If the macro does not require use of the keyboard or mouse, it plays back again in 12 minutes.

If the macro does change the idle status of the computer, then the macro does not play back again until the system has been idle for 10 minutes followed by the 12 minute wait period.

Only Run Once

Select this option to only run the macro once after the computer has been idle for the number of minutes specified.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

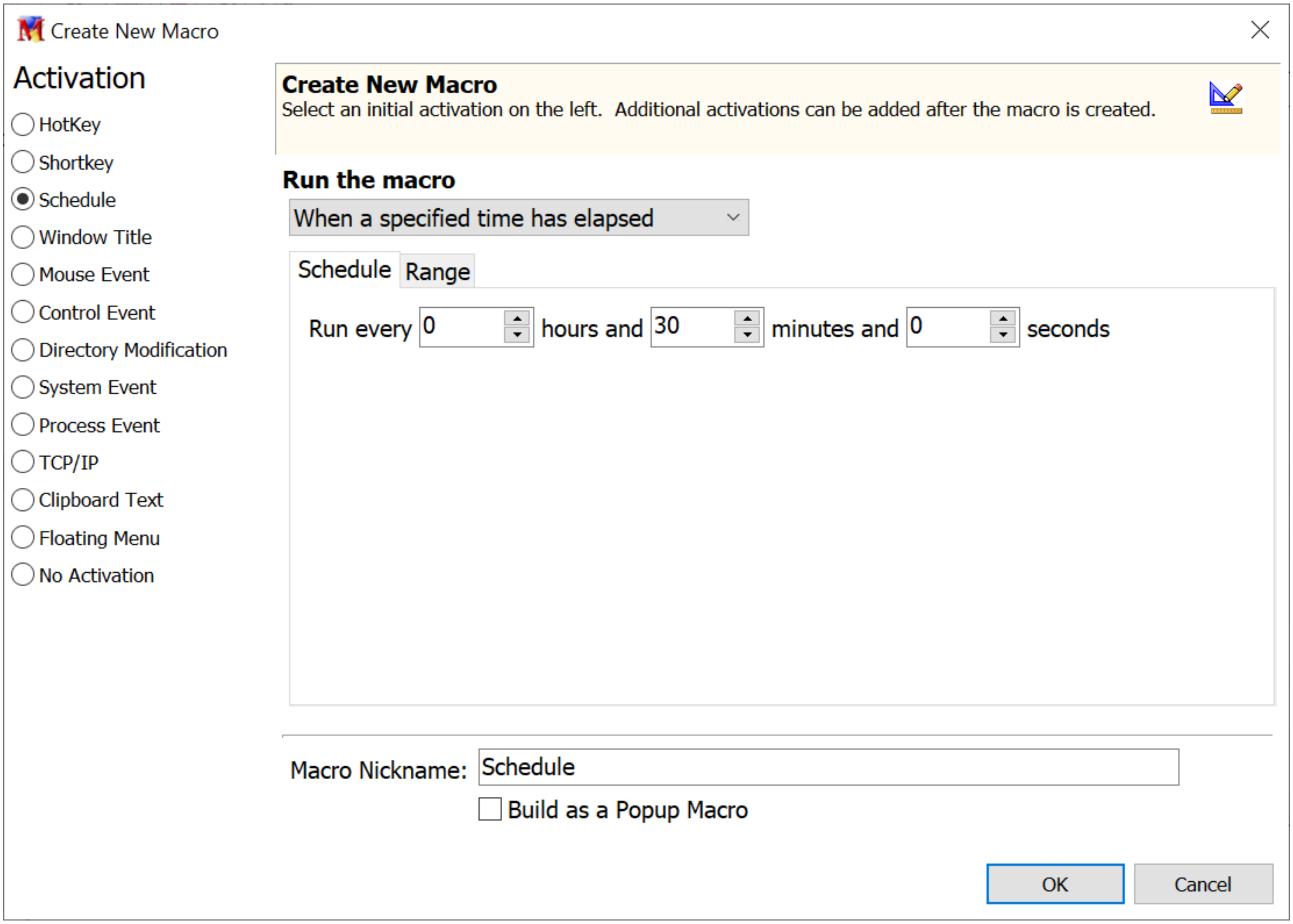
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

When a Specified Time has Elapsed

The When a Specified Time has Elapsed option repetitively runs a macro every XX number of hours, minutes and/or seconds.



Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro
When a specified time has elapsed

Run every 0 hours and 30 minutes and 0 seconds

Macro Nickname: Schedule

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Run Every

Enter the time between every scheduled run of the macro. If the macro is to run every 30 minutes, then set the hours and seconds to 0. Set the minutes to 30.

- ✓ **Note 1:** If a macro is scheduled to run every 1 second and the macro takes 5 seconds to execute, then this function will not operate properly.

- ✓ **Note 2:** If the macro is scheduled to run more frequently than every 10 seconds, it is necessary to change the Scheduler Frequency default setting. This setting instructs Macro Express to check every 10 seconds to see if a scheduled macro is ready to play back. Reduce this number in order for scheduled macros to play back more frequently.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

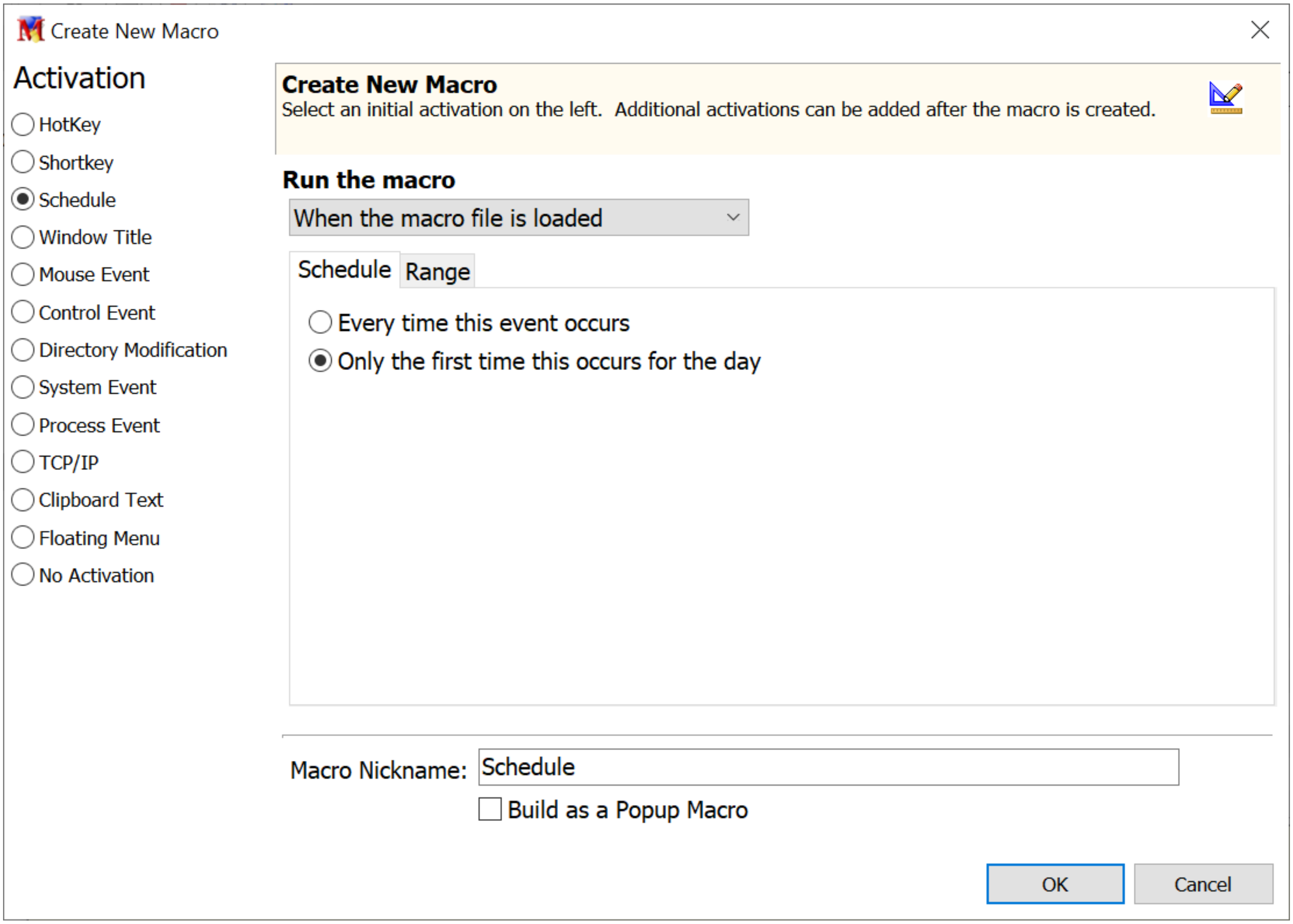
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

When the Macro File is Loaded

This is similar to the At Startup schedule with one difference. Any time that the macro file containing the scheduled macro is loaded, the macro runs. When Macro Express loads, the file opens and the macro plays back.



Create New Macro

Activation

○ HotKey
○ Shortkey
● Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Run the macro
When the macro file is loaded

Schedule | Range

○ Every time this event occurs
● Only the first time this occurs for the day

Macro Nickname: Schedule
□ Build as a Popup Macro

OK | Cancel

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Schedule

Every Time this Event Occurs

Select this option to play back the macro every time the file loads. If another macro file is opened and then the original file containing the scheduled macro is reopened, the macro executes. Or if Macro Express is shut down during the day and later restarted, the macro runs.

Only the First Time this Occurs for the Day

This option limits the playback to just once a day. The first time the file loads for the day, the macro will play back - but not afterwards, until the next day.

Range

Select a date and time range for the macro to play back from the three options available in the drop down list. These range options are described below.

Run the Macro Within this Date Range

Select a range of dates during which the macro may run. The macro plays back during the hours of the day and for the duration of the days selected. Select the Start Date when the macro is to first start running. Then select the End Date when the macro ceases to run.

Run Between These Times Each Day

Select a start and an end time for the macro to run.

Run the Macro Between the Dates and Time

Select a starting date and time. Then select an ending date and time. The macro is available to run anytime between the dates and times specified.

Run the Macro Indefinitely

Run the Macro Indefinitely does not stop the macro on a specific date. It continues to run at the appointed time until the schedule is canceled or the macro is disabled or deleted.

Start Calculating the Next Run Time At

Select the date and time to start calculating the next run of the macro.

Run the Macro Between These Hours of the Day

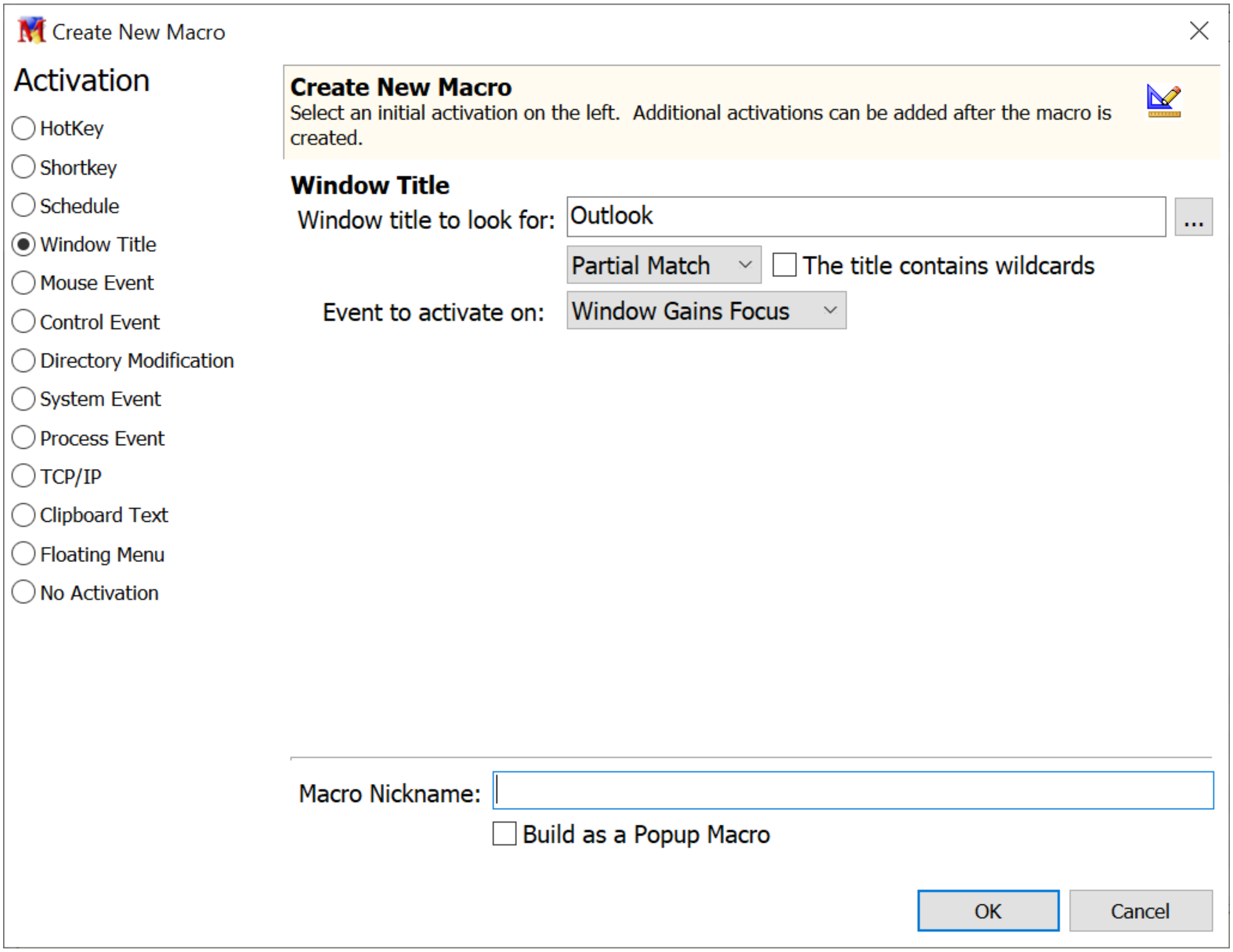
Optionally choose to have the macro only available to run during certain hours of the day. Insert the start and end times.

Run the Macro on These Days of the Week

Select which days of the week to run the macro. For example, remove the check marks from Saturday and Sunday if the macro is not needed on weekends.

Window Title

The Window Title activation instructs the macro to execute any time a specific event occurs for the window title selected.



Create New Macro

Activation

- ○ HotKey
- ○ Shortkey
- ○ Schedule
- ● Window Title
- ○ Mouse Event
- ○ Control Event
- ○ Directory Modification
- ○ System Event
- ○ Process Event
- ○ TCP/IP
- ○ Clipboard Text
- ○ Floating Menu
- ○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Window Title

Window title to look for: Outlook ...

Partial Match □ The title contains wildcards

Event to activate on: Window Gains Focus

Macro Nickname:

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Window Title to Look For

In the edit box, enter the title of the Window for the macro to find. The window title is that which appears in the caption bar of a window. Macro Express examines all window titles running on the computer in an attempt to find a match.

Click on the Browse button to see a list of all windows currently running on the computer. This includes both windows that are visible when open and other system windows that are not seen, though open. Select the window and click OK.

Partial Match/Exact Match

The Window Title does not need to be an exact match. A partial match will work. For example, if looking for the Windows Notepad program, one could insert "notepad" or "notep". This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard includes the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Event to Activate On

Select from one of four different events to trigger the macro. These are described below.

Event	Description
Window Gains Focus	The macro starts as soon as the specified window becomes the active window and has focus.
Window Loses Focus	The macro begins playback as soon as the window is minimized or sent to the background.
Window is Closed	The macro starts when the defined window is closed.
Window is Opened	The macro starts when the specified window is open.

Macro Nickname

Optionally assign a name to the macro to more readily determine what the macro does.

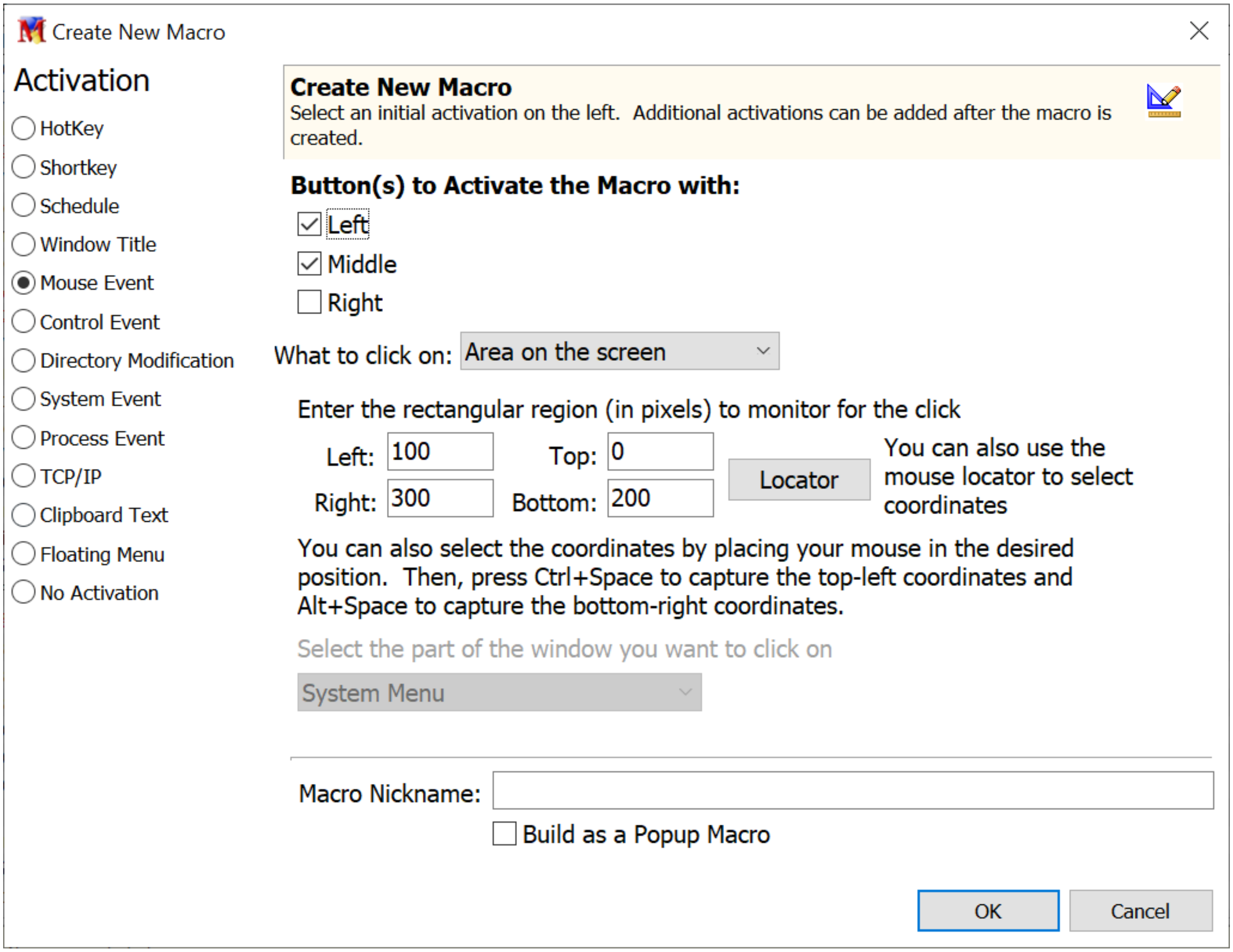
Build as a Popup Menu

Select this option to open the Menu Builder window and create a popup menu.

Mouse Event

Create macros that run when the mouse button or a combination of mouse buttons is clicked.

- ✓ **Note:** Windows may prevent some mouse activations from working based on various combinations of mouse buttons, window parts, etc. that are selected.



Create New Macro

Activation

○ HotKey
○ Shortkey
○ Schedule
○ Window Title
● Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Button(s) to Activate the Macro with:

☑ Left
☑ Middle
□ Right

What to click on: Area on the screen

Enter the rectangular region (in pixels) to monitor for the click

Left: 100 Top: 0
Right: 300 Bottom: 200

You can also use the mouse locator to select coordinates

You can also select the coordinates by placing your mouse in the desired position. Then, press Ctrl+Space to capture the top-left coordinates and Alt+Space to capture the bottom-right coordinates.

Select the part of the window you want to click on: System Menu

Macro Nickname:

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Button(s) to Activate the Macro with

Macro activations may be assigned to the Left, Middle or Right mouse buttons. If selecting more than one mouse button, such as in the example above, then the macro will activate by pressing either the Left or the Middle mouse button.

What to click on

After assigning the mouse button or buttons to press, select what to click on in order to trigger the macro - Area on Screen, Area on Window or Window Part. These are explained below.

Area on Screen

Use this option to define regions on the screen that can be assigned as a mouse activated macro. Clicking on the assigned area triggers the macro. This option requires the input of four pixel coordinates - Left, Top, Right, and Bottom. Click on the area within these four pixel coordinates to start the macro.

Area on Window

Use this option to define regions on a window that can be assigned as a mouse activated macro. Clicking on this area of the window triggers the macro. This option requires the input of four pixel coordinates - Left, Top, Right, and Bottom. Click on the area within these four pixel coordinates to start the macro.

Locator

Click on the Locator button to launch this small utility to quickly help define the pixel coordinates. After the locator is running, move the mouse to the upper left corner of the region to be defined. Press Ctrl+Space to insert the pixel coordinates in the Left and Top fields. Then move the mouse to the lower right corner of the region to be defined. Press Alt+Space to insert the pixel coordinates into the Right and Bottom fields. Press OK to save the coordinates.

This action defines a square or rectangular area on the screen. Click within the defined square or rectangular area with the designated mouse button to start the macro.

Window Part

Define a macro to start by clicking on a specific part of a window. Parts include the window border, window title bar, window menu, horizontal scroll bar, vertical scroll bar, system menu, minimize button and maximize button.

✓ **Note:** A few options have been excluded because they will interfere with Windows. These include such things as Left Mouse with the Maximize or Minimize buttons.

Macro Nickname

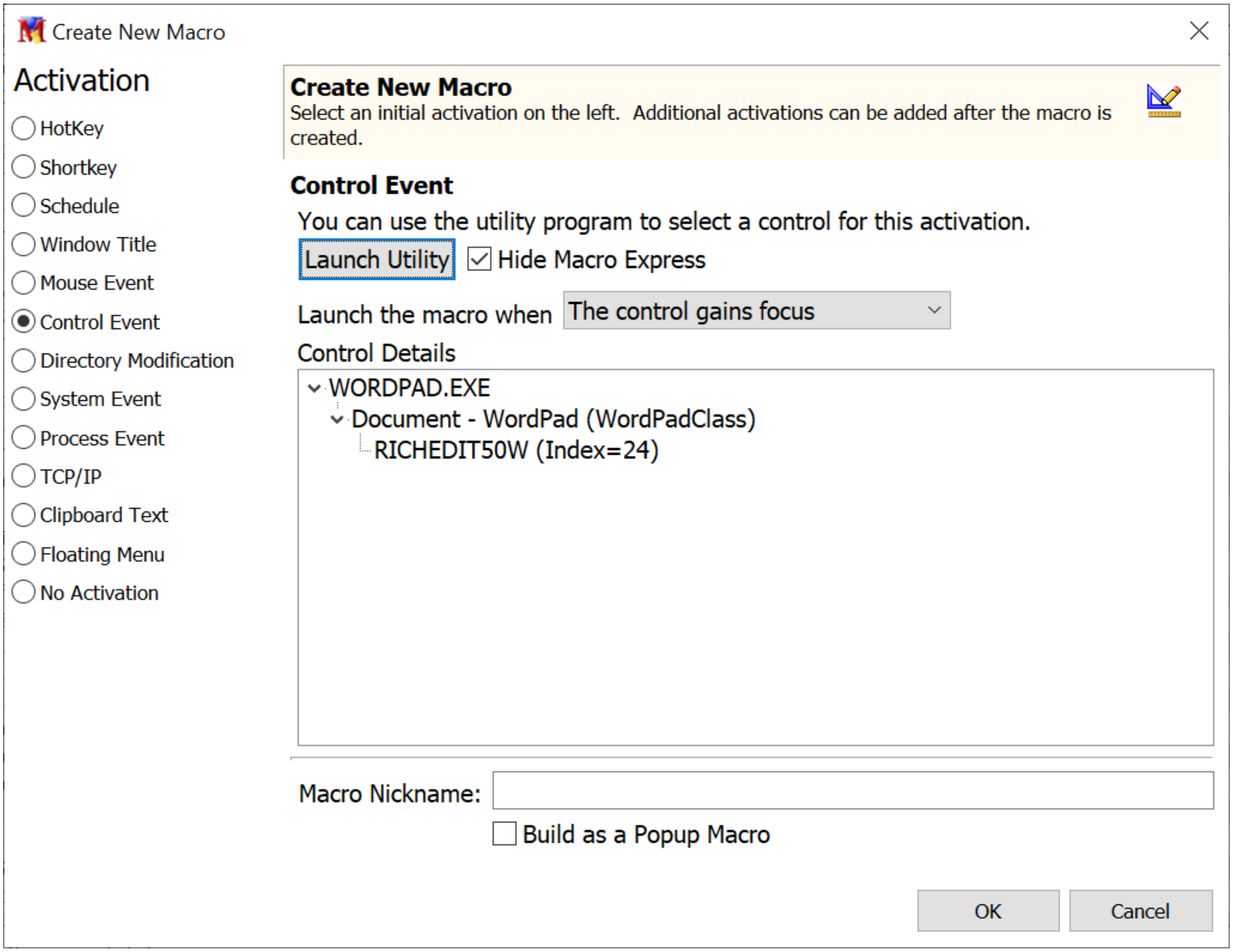
Optionally assign a name to the macro to more readily determine what the macro does.

Build as a Popup Menu

Select this option to open the Menu Builder window and create a popup menu.

Control Event

For our purposes, Control refers to the identifiable parts of a specific application window. Controls may include such things as a button, an edit box, a toolbar, a window title bar or others. The control may be part of the main application window or be part of a child window (a window within a window).



Create New Macro

Activation

○ HotKey
○ Shortkey
○ Schedule
○ Window Title
○ Mouse Event
● Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Control Event

You can use the utility program to select a control for this activation.

Launch Utility ☑ Hide Macro Express

Launch the macro when: The control gains focus

Control Details

```
WORDPAD.EXE
  Document - WordPad (WordPadClass)
    RICHEDIT50W (Index=24)
```

Macro Nickname:

□ Build as a Popup Macro

OK Cancel

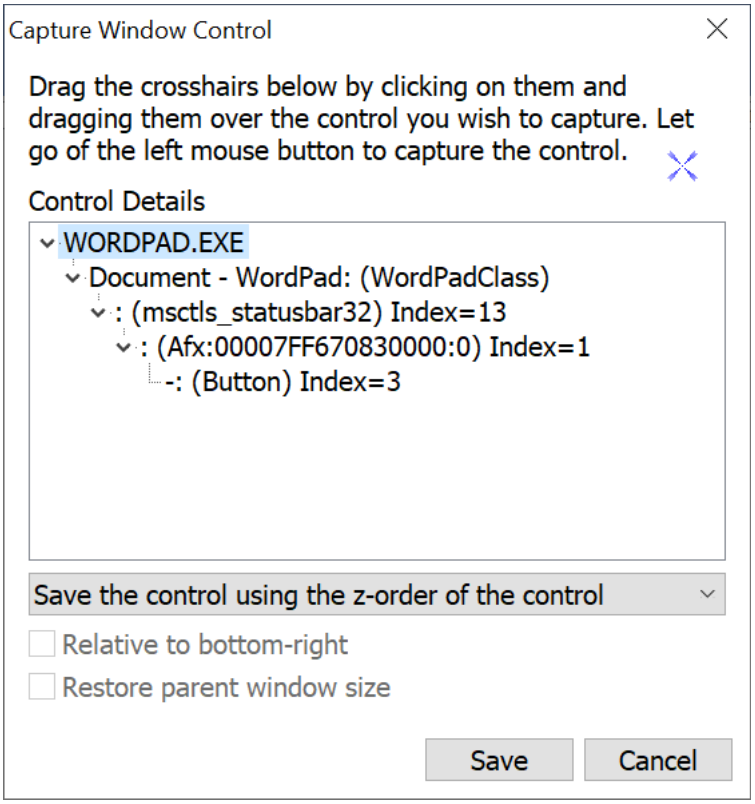
Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

With Control Activation selected, the macro plays back when the control gains focus or the window with the control gains focus. In the example above the control name is listed. The program .exe is displayed followed by the window component that must be active in order for the macro to play back.

Launch Utility

The first step is to select the control that will activate the macro. Below is an image of the Capture Window Control window that appears when launching the control utility.

To select a Window Control place the mouse cursor on the crosshair image, hold down the left mouse button and drag the mouse over the control. When dragging the crosshairs over title bars, buttons, edit boxes, forms, toolbars, etc., the outline of the controls are highlighted. Once the control needed is highlighted, release the left mouse button. Click on the Save button to return to the Create New Macro window. The control details are saved in the edit box.



Capture Window Control

Drag the crosshairs below by clicking on them and dragging them over the control you wish to capture. Let go of the left mouse button to capture the control.

Control Details

```
WORDPAD.EXE
  Document - WordPad: (WordPadClass)
    : (msctls_statusbar32) Index=13
      : (Afx:00007FF670830000:0) Index=1
        -: (Button) Index=3
```

Save the control using the z-order of the control

☐ Relative to bottom-right

☐ Restore parent window size

Save the control using the z-order of the control

This method saves the control information based on the order in which the controls are created or manipulated by the parent program. In some applications, the z-order may change without notice or the user being able to determine that it has changed.

Save the control using the contents of the control

This option makes the capture of a control more exact by also capturing the text found in the control. So for example, assume capturing an edit box with the word Test in it. During playback the macro looks for the edit box and verifies that it contains the word Test. If the edit box text is changed to something else, such as Reset or left blank, then the control that was captured, such as C[1], will not match the actual control in the window - edit box.

Save the control using its coordinates

There are some controls that change information each time the window with the control is opened or the program with the control is restarted as mentioned above. This renders the information saved to the control obsolete and the control will not be found the next time the macro is run. This Coordinates option saves the control information based on the location of the control in the window.

Relative to the bottom-right

Typically the coordinates of the control are measured from the top, left corner of the window that contains the control. Use this alternative method to obtain the position of the control from the bottom, right corner of the window.

Restore parent window size

This option restores the size of the parent window (the window that contains the control) to the size of the window when the original capture of the control was made.

✓ **Note:** Not every button, menu item or component on a program is actually a Control. Sometimes the only control in a given program is the window or dialog that pops up when the program starts. We have also noticed that many of Microsoft's programs seem to have fewer controls than programs from other manufacturers. How the controls work will depend upon the way the program in use is designed.

Hide Macro Express

Hide Macro Express does what it implies. The program hides or gets out of the way after pressing the Launch Get Control Utility. This makes it easier to locate the application containing the control to capture.

Launch the Macro When

After selecting the control, decide when to activate the macro. Choose either when the control gains focus or when the window that contains the control gains focus.

The Control Gains Focus

This option requires that the control be part of the active window and the control gain focus. For example the mouse cursor could be in an edit box, giving focus to this control. Or a button may be highlighted and have control.

The Control's Window Gains Focus

With this option the macro starts when the window containing the selected control gains focus.

Control Details

This edit box lists the details of the selected control using the Launch Utility. The information should match what was listed in the Capture Window Control window.

Macro Nickname

Optionally assign a name to the macro to later help more readily determine what the macro does.

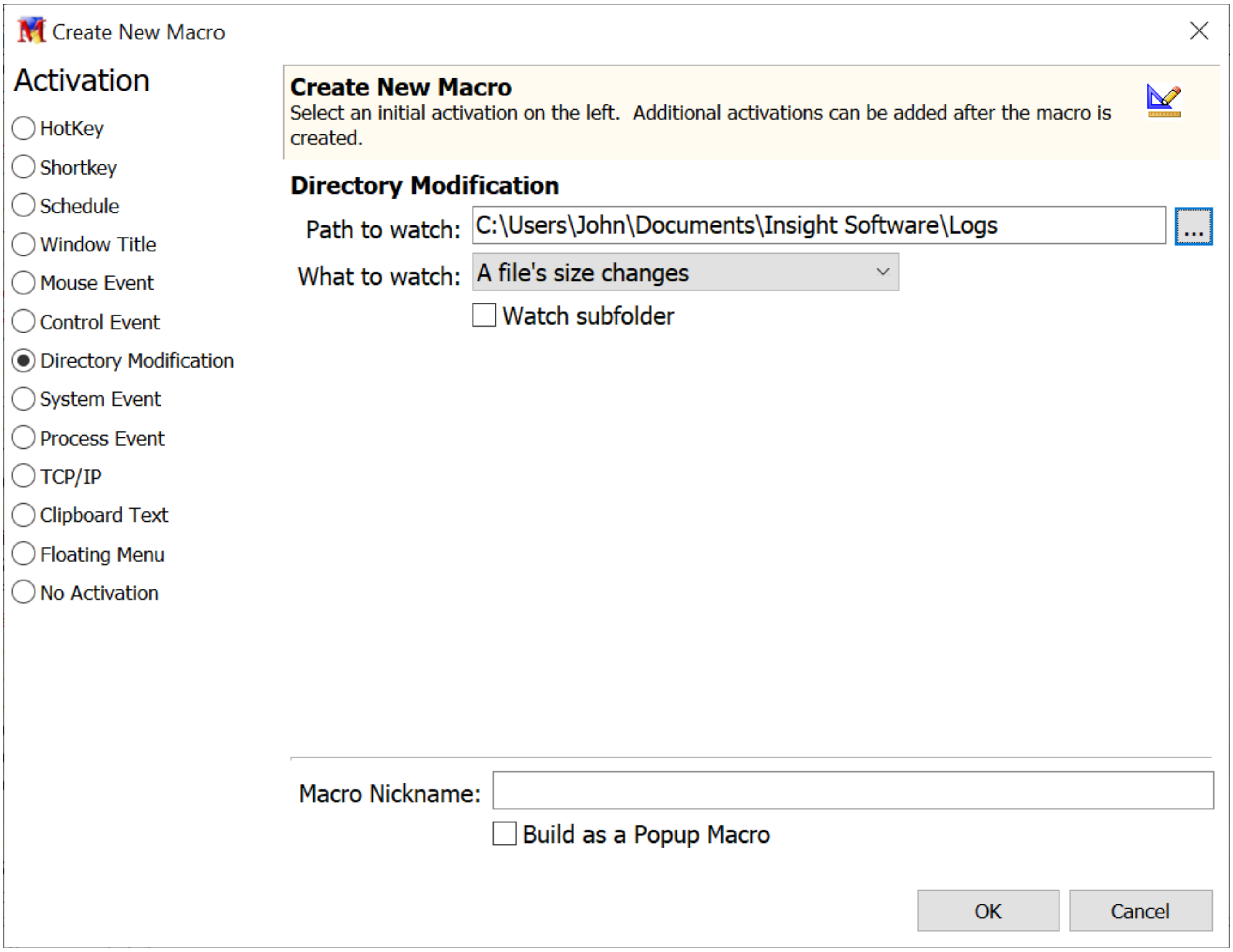
Build as a Popup Menu

Select this option and click on the OK button to open the Menu Builder and create a popup menu.

Directory Modification

This activation type looks for changes to files in a folder or to the folder. A macro starts when a change is made to a file in the selected folder or to the folder itself. Some of the available options include the macro looking for changes to file size, attributes, write time, security descriptor or checking if a file has been renamed or deleted.

- ✓ **Note:** There is a maximum of 64 Directory Modification macros that may be active at any one time. Any macros over the 64 limit are ignored. This total spans multiple macro files that may be open.



Create New Macro

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Activation

○ HotKey
○ Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
● Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Directory Modification

Path to watch: C:\Users\John\Documents\Insight Software\Logs

What to watch: A file's size changes

□ Watch subfolder

Macro Nickname:

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Path to Watch

Insert the folder path for the macro to monitor or click on the Browse button to select the folder path.

What to Watch

Select from a list of file activities that the macro can monitor. These activities are explained below.

Create, Rename or Delete a File

The macro activates when a file in the selected folder is either created, renamed or deleted.

Create or Delete a Directory

The macro starts when the specified folder is created or deleted.

File or Directory Attributes Change

The macro plays back when the attributes of a file in the folder or the folder itself is changed. The file attributes may include the Read-Only flag, Archive setting, etc.

A File's Size Changes

Use this option to initiate a macro when the size of a file in the selected folder changes.

A File's Last Write Time Changes

The macro starts when the last write time (or modified time) of a file in the folder changes.

A Security Descriptor Changes

The macro starts if one of the security options of a file in the folder or the folder itself has changed.

Watch Subfolders

Select this option to monitor any subfolders of the primary folder chosen.

Macro Nickname

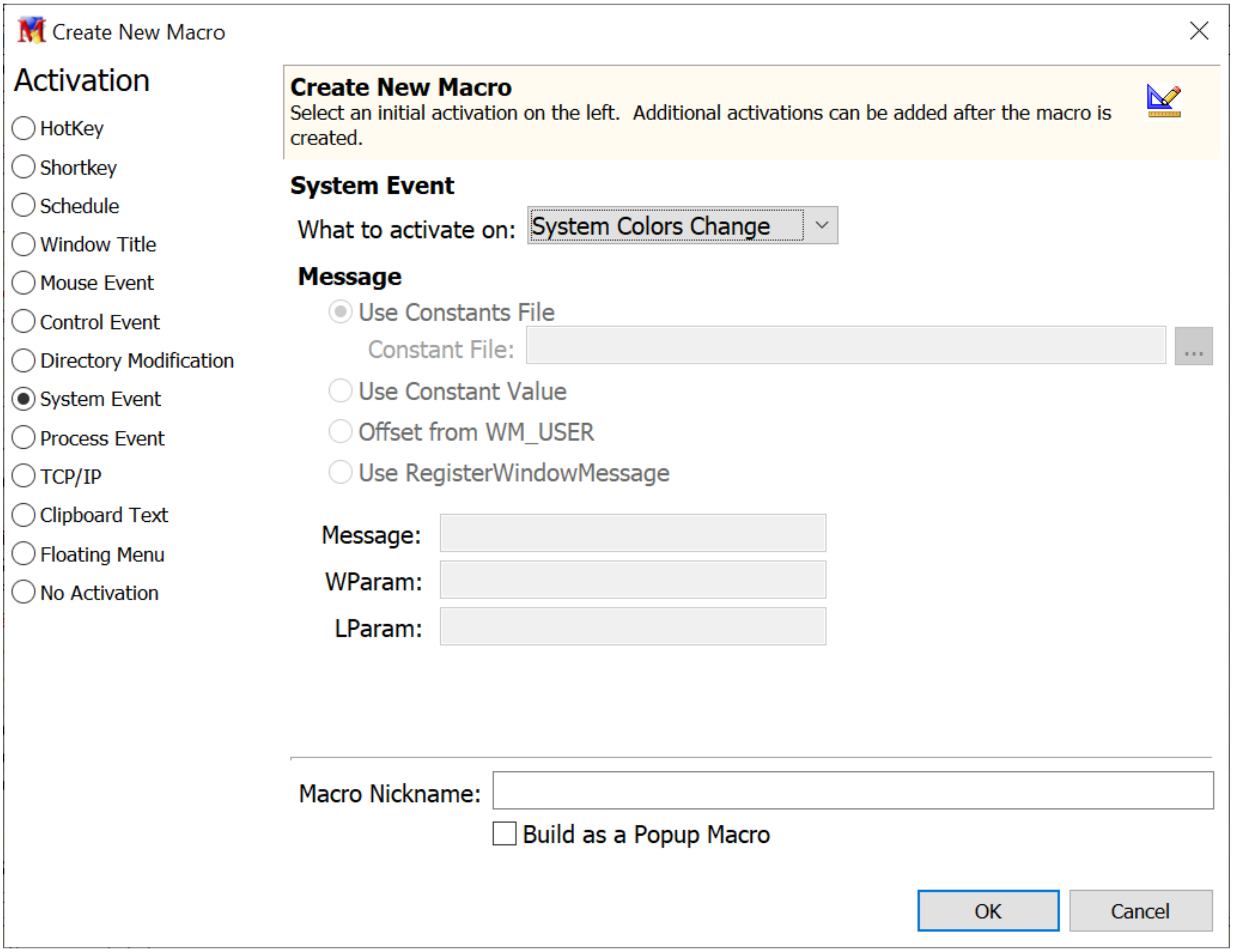
Optionally assign a name to the macro to later more readily help determine what the macro does.

Build as a Popup Menu

Select this option and click on the OK button to open the Menu Builder and create a popup menu.

System Event

The System Event activation looks for changes to the system. The macro starts when the specified system change occurs.



Create New Macro

Activation

○ HotKey
○ Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
● System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

System Event

What to activate on: System Colors Change

Message

● Use Constants File
Constant File:
○ Use Constant Value
○ Offset from WM_USER
○ Use RegisterWindowMessage

Message:
WParam:
LParam:

Macro Nickname:
□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Incoming Message

Windows is a message based operating system. There are messages broadcast throughout the system when certain events occur, such as when a hardware component is added or removed.

This is another way for custom applications to launch macros. Rather than passing the nickname through the postmessage routine, use this method to post a single message.

When using the Incoming Message activation, select from the four types of messages available.

Use Constants File

The user specifies an ini file containing a list of constant names and their values. The "Message to Post" field should contain a constant name from this file.

An example would be the WM_LBUTTONDOWN message in the PostMessage.ini file. In the file, the following line appears:

```
WM_LBUTTONDOWN=513
```

This means that when the user specifies WM_LBUTTONDOWN as the message, message 513 is actually sent.

Constants File

The user specifies in the "Message to Post" field a message number to send to the specified control.

Offset from WM_USER

WM_USER is a Windows constant that marks the beginning of the messages that programmers can use for anything. In this case, the user specifies either a positive or negative number that is added to WM_USER. Currently, Windows defines WM_USER as 1024.

Use RegisterWindowMessage

This command allows the user to enter a constant string that can be used by Windows to generate a message. This is primarily used to communicate with other applications that have registered the same message.

Microsoft Definition of RegisterWindowMessage

The RegisterWindowMessage function defines a new window message that is guaranteed to be unique throughout the system. The message value can be used when sending or posting messages.

The RegisterWindowMessage function is typically used to register messages for communicating between two cooperating applications.

If two different applications register the same message string, the applications return the same message value. The message remains registered until the session ends.

Only use RegisterWindowMessage when more than one application must process the same message. For sending private messages within a window class, an application can use any integer in the range WM_USER through 0x7FFF.

Message

The message input into this field comes from one of three sources. The message will be a constant from the Constants file, a string passed to RegisterWindowMessage, or a value from WM_USER.

WParam - LParam

These fields allow the user to specify the WParam and LParam parameters. The contents of these parameters depend upon the message being sent. These values can be found in the Platform API library at https://msdn.microsoft.com.

✓ **Note:** Consult https://msdn.microsoft.com for a list of messages available.

✓ **Note:** When Macro Express is installed, three .ini files (PostMessage.ini, PostCommCtrl.ini and PostRichEdit.ini) are also installed into the Macro Express folder. The default path is C:\Program Files (x86)\Macro Express Pro 6. These .ini files contain message definitions that can be used for reference. Double click on the files to open and view in Notepad or Wordpad.

System Colors Change

Use this option to start a macro when a change is made to the Windows system colors on the computer.

Resolution Change

Use this option to activate a macro when the system's screen resolution is changed.

Macro Nickname

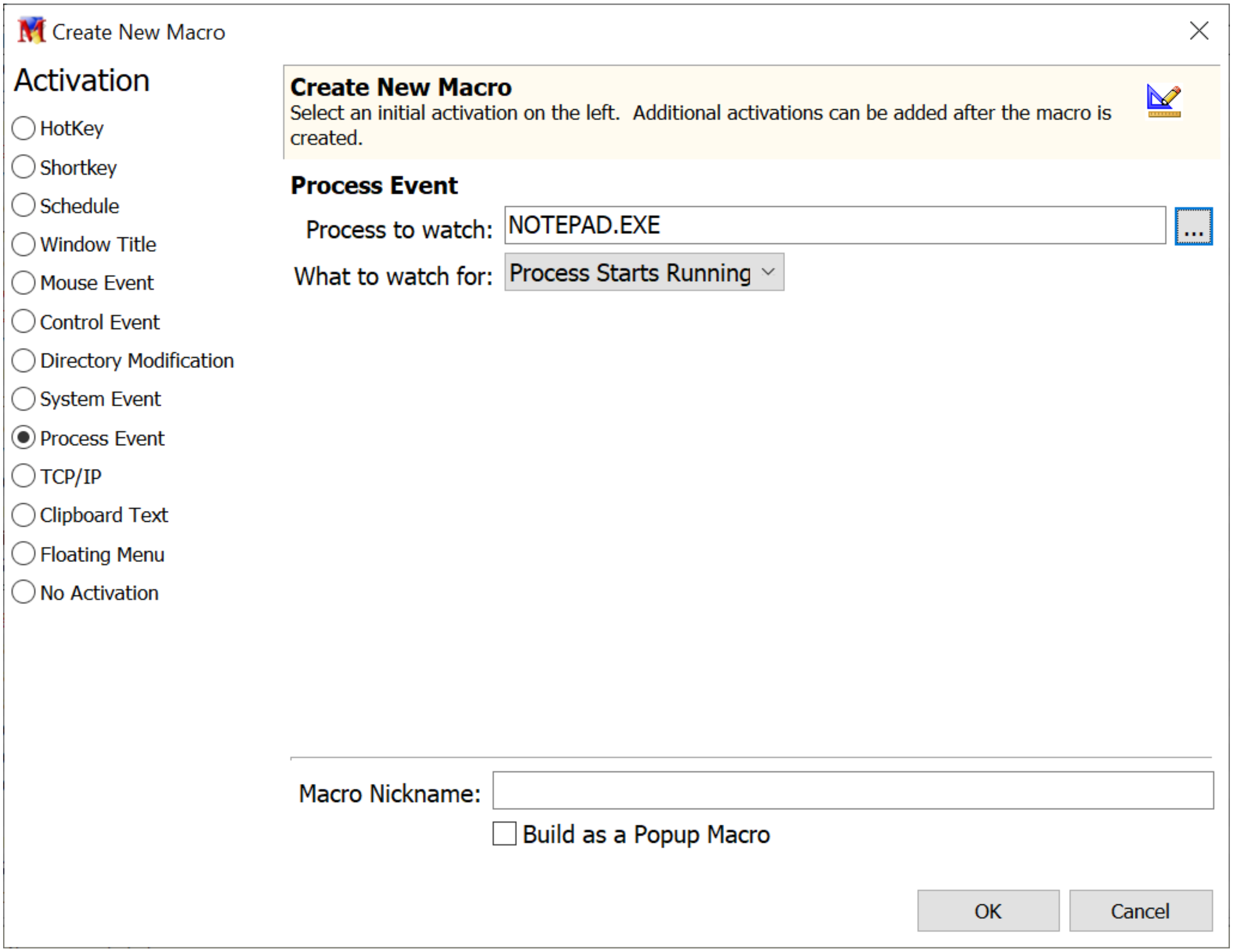
Optionally assign a name to the macro to later more help readily determine what the macro does.

Build as a Popup Menu

Select this option and click on the OK button to open the Menu Builder and create a popup menu.

Process Event

The Process Event activation method initiates a macro when a Process first starts or when it closes.



Create New Macro

Activation

○ HotKey
○ Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
● Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Process Event

Process to watch: NOTEPAD.EXE

What to watch for: Process Starts Running

Macro Nickname:

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Process to Watch

Click on the Browse button to select the process to monitor. Browse through a list of processes in the Start menu or from a list of processes currently running on the computer. Or look for where the process is installed on the computer.

What to Watch For

Process Starts Running

The macro activates as soon as the chosen process starts running.

Process Stops Running

The macro starts when a process ends or closes.

Macro Nickname

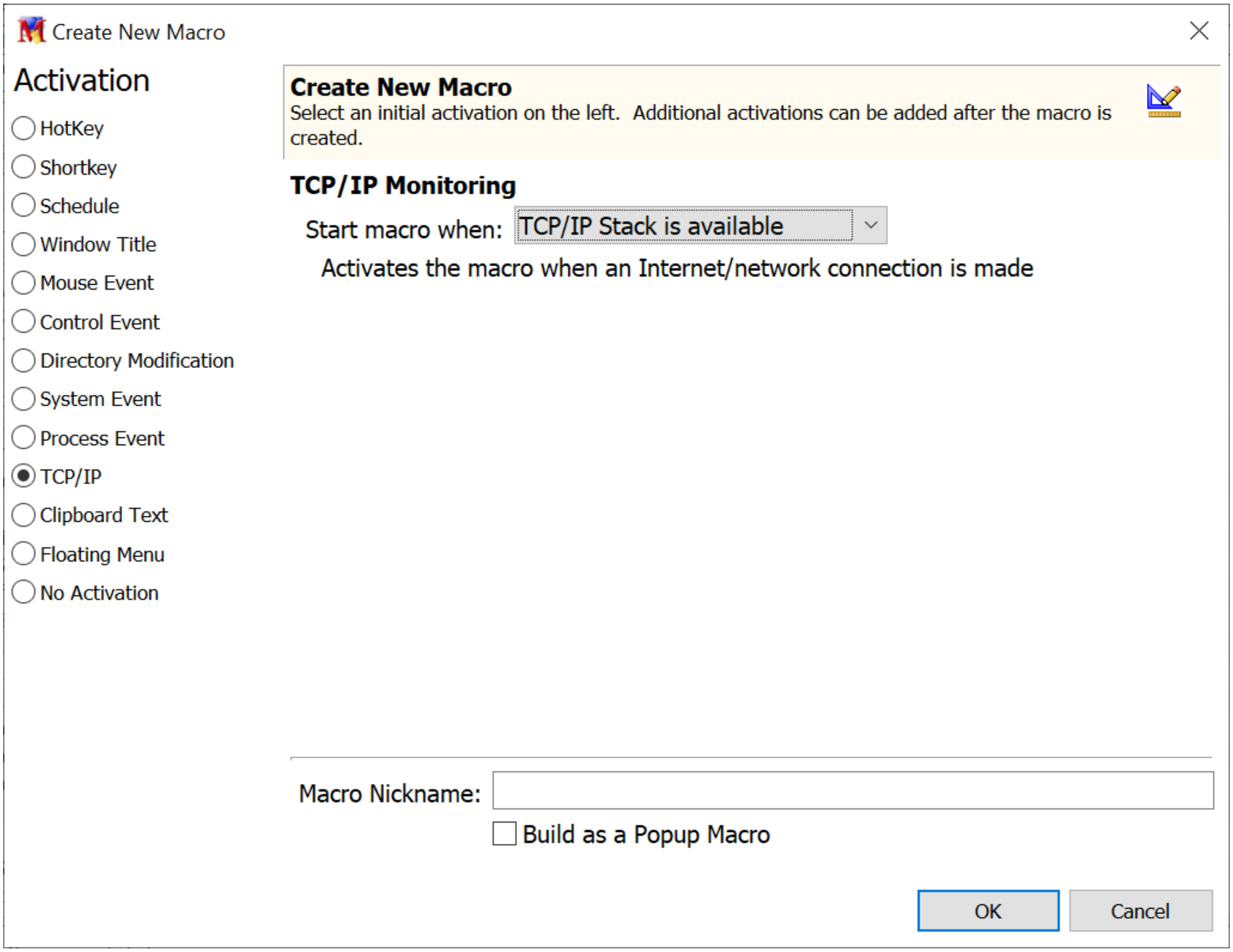
Optionally assign a name to the macro to later more readily help determine what the macro does.

Build as a Popup Menu

Select this option and click on the OK button to open the Menu Builder and create a popup menu.

TCP/IP

The TCP/IP activation monitors when an Internet or Network connection has been made or released.



Create New Macro

Activation

○ HotKey
○ Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
● TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

TCP/IP Monitoring

Start macro when: TCP/IP Stack is available

Activates the macro when an Internet/network connection is made

Macro Nickname:

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

TCP/IP Stack is Available

This option activates the macro when an Internet or Network connection is successful.

TCP/IP Stack is Unavailable

This option activates the macro when all Internet or Network connections are released.

Macro Nickname

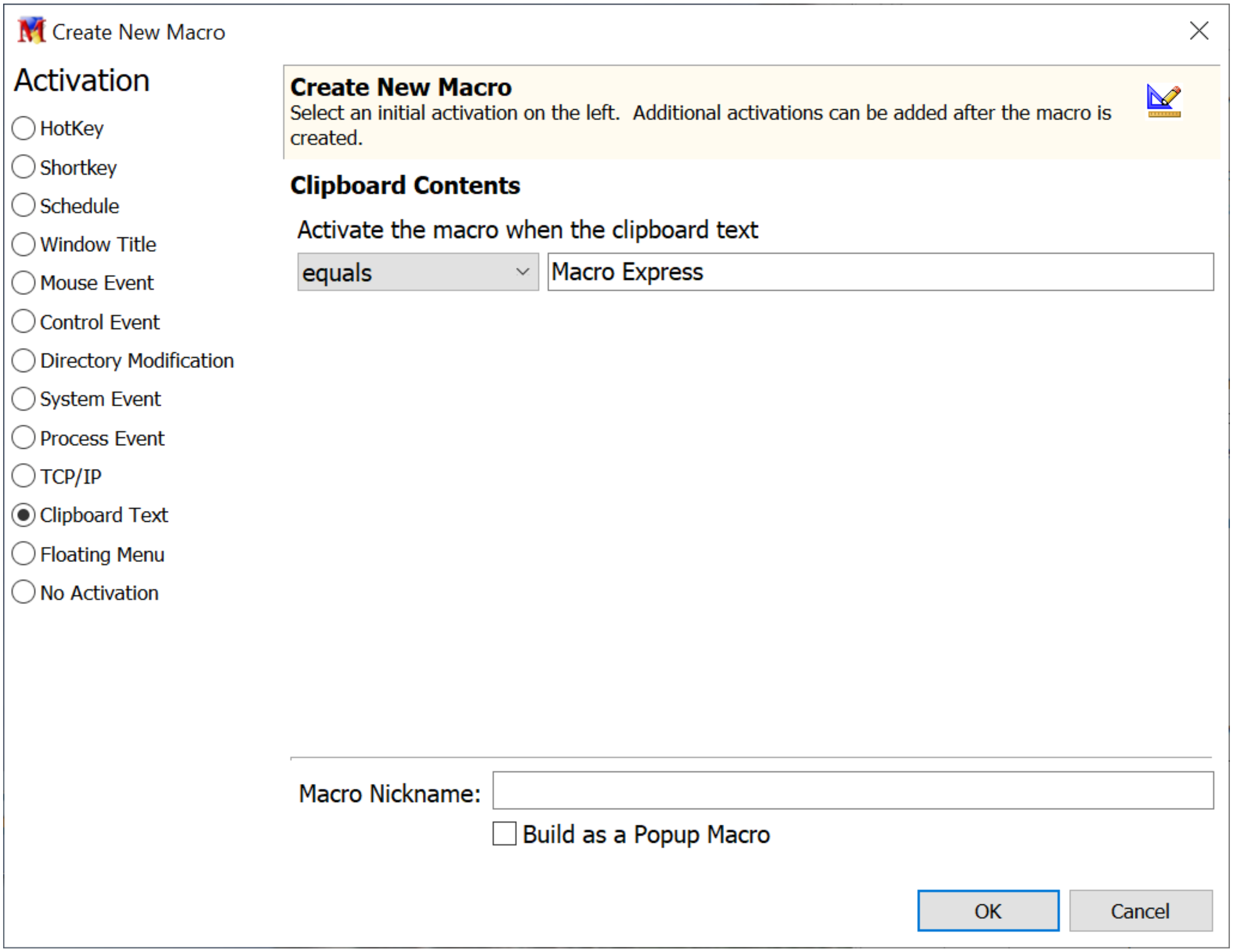
Optionally assign a name to the macro to later more readily help determine what the macro does.

Build as a Popup Menu

Select this option and click OK to open the Menu Builder and create a popup menu.

Clipboard Text

This activation method starts a macro based on the contents of the clipboard. Select whether the clipboard does or does not match the text entered into the edit field. Or choose whether the clipboard contains or does not contain specific text.



Create New Macro

Activation

○ HotKey
○ Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
● Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Clipboard Contents

Activate the macro when the clipboard text

equals | Macro Express

Macro Nickname:

□ Build as a Popup Macro

Macro Express Explorer > Macro > Add Macro - Or click on New Macro Icon

Equals / Does not Equal

These options require that the text in the clipboard match or not match the word or phrase selected as a trigger. For example, suppose Macro Express is listed as the clipboard text trigger and the Equals option is selected. The macro starts when the words Macro Express are copied to the clipboard.

The macro starts if the activate option is changed to Does not Equal and text other than Macro Express is copied to the clipboard.

- Use the Does not Equal option with caution as this may trigger macros unexpectedly when copying items to the clipboard.

Contains / Does not Contain

These options require that the text in the clipboard contain or not contain the word or phrase selected as a trigger. For example, suppose Macro Express is listed as the clipboard text trigger and the Contains option is selected. The macro starts when the words "I really like Macro Express" are copied to the clipboard. It plays back because the trigger - Macro Express - is part of the clipboard contents.

The macro starts if the option is changed to Does not Contain and text that does not include the words Macro Express is copied to the clipboard.

- ❖ Use the Does not Contain option with caution as this may be trigger macros unexpectedly when copying items to the clipboard.

Macro Nickname

Optionally assign a name to the macro to later more readily help determine what the macro does.

Build as a Popup Menu

Select this option and click on the OK button to open the Menu Builder and create a popup menu.

No Activation

It is possible to create a macro and not assign any activation key to the macro. There are two primary uses for the No Activation option. First, create generic macro templates that can be incorporated into other macros - a set of instructions useful for insertion into several other macros. The Macro Run command allows for inserting No Activation macros into other macro scripts.

A second use for this option is to place the macro in a popup or floating menu. A macro activation is not necessary if the macro is part of a popup or floating menu. The only way to invoke the macro would be through the menus.

Macro Nickname

Optionally assign a name to the macro to later more readily help determine what the macro does.

Build as a Popup Menu

Select this option and click on the OK button to open the Menu Builder and create a popup menu.

Stopping a Macro

Occasionally it may be necessary to stop a running macro before it has finished. There are several ways to stop a running macro.

Aborting a Macro

Below are several ways to stop a running macro.

- Press the default key sequence of Win + End.
- If the Running icon displays when a macro is running, right mouse click on the icon in the notification area.
- Change the default sequence to another combination in Options > Preferences > Playback > Miscellaneous.

Please note that macros that consist of many keystrokes can send the keystrokes rather quickly and they are buffered in Windows. Once such commands have been buffered in Windows, aborting a macro does not cancel them. In these cases, it may appear that the abort command did not work when in reality the macro has already finished issuing its commands and Windows is now processing them.

If more than one macro is running, all macros will be stopped when initiating any of the above mentioned actions.

✓ **Note:** The Timed Delay in Seconds and the Wait for Time to Elapse commands suspend Macro Express for the amount of time specified. During this delay period the macro cannot be terminated.

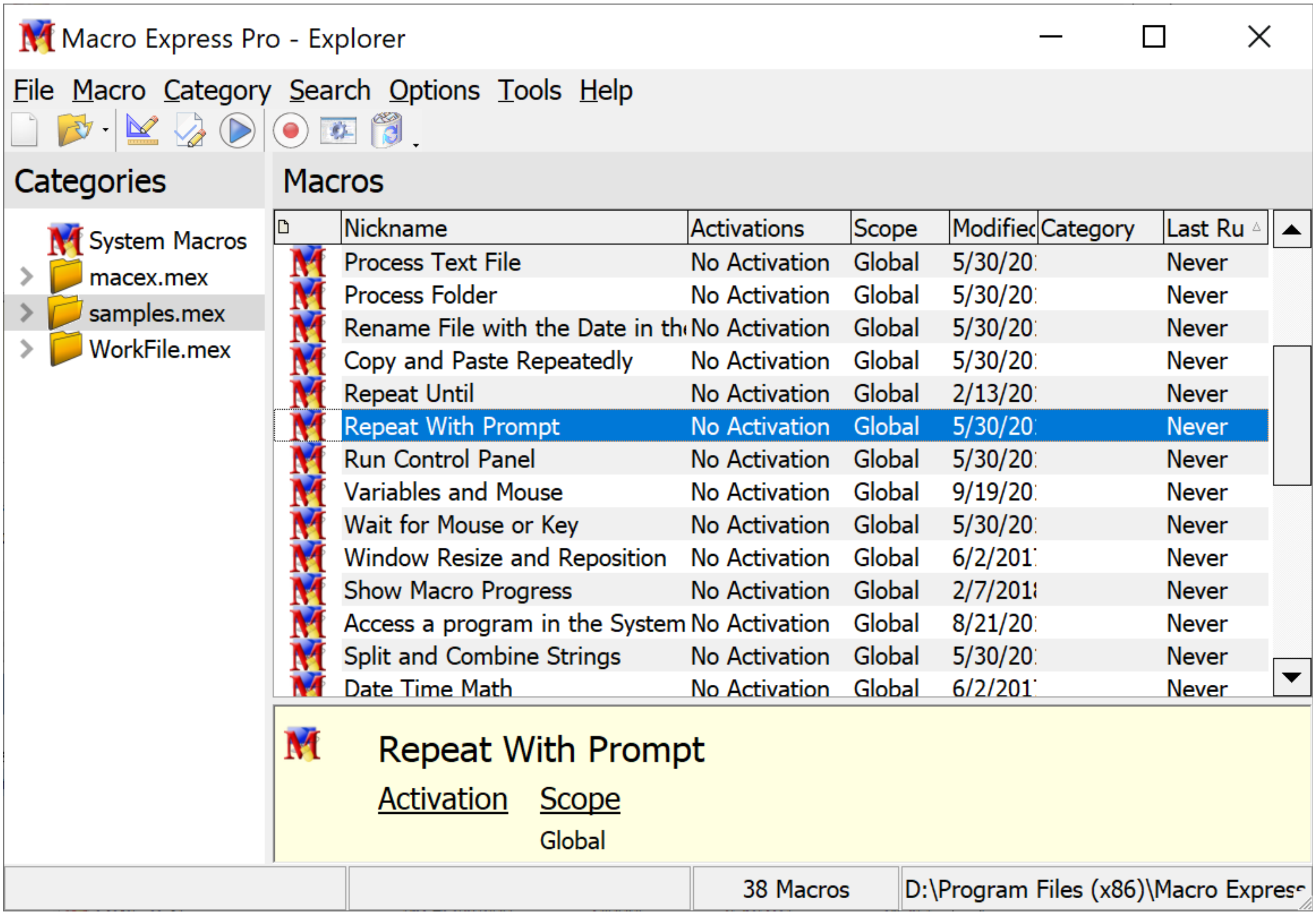
Macro Stop

The Macro Stop command stops the execution of a macro before the macro steps have completed. Insert the Macro Stop command into the macro script. Whenever this line in the macro is reached, the macro terminates. In most cases this command is used in conjunction with a conditional statement. For example, if a desired condition is not met, terminate the macro.

Using the Macro Explorer

Macro Explorer

By default Macro Express opens to the Macro Express Pro - Explorer window. This window displays a list of macros contained in the file currently open. The Macro Explorer is the starting point for creating new macros or editing existing ones.



Nickname	Activations	Scope	Modifiec	Category	Last Ru
Process Text File	No Activation	Global	5/30/20		Never
Process Folder	No Activation	Global	5/30/20		Never
Rename File with the Date in th	No Activation	Global	5/30/20		Never
Copy and Paste Repeatedly	No Activation	Global	5/30/20		Never
Repeat Until	No Activation	Global	2/13/20		Never
Repeat With Prompt	No Activation	Global	5/30/20		Never
Run Control Panel	No Activation	Global	5/30/20		Never
Variables and Mouse	No Activation	Global	9/19/20		Never
Wait for Mouse or Key	No Activation	Global	5/30/20		Never
Window Resize and Reposition	No Activation	Global	6/2/201		Never
Show Macro Progress	No Activation	Global	2/7/201		Never
Access a program in the System	No Activation	Global	8/21/20		Never
Split and Combine Strings	No Activation	Global	5/30/20		Never
Date Time Math	No Activation	Global	6/2/201		Never

Repeat With Prompt

Activation Scope

Global

38 Macros D:\Program Files (x86)\Macro Expres

Left click on the Macro Express icon in the notification area or press the system macro Ctrl + Alt + Shift + W

Actions

The icons at the top of the Macro Explorer window are the primary methods for navigating through the program. Clicking on any of these icons will load a different program function.

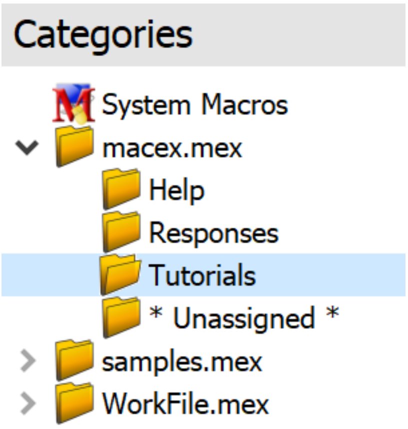
	New File	Create a new macro (.mex) file.

📂 ▾	Open File	Open an existing macro (.mex) file. Or click on the triangle button to view a list of recently used macro files and select one to open.
	New Macro	Click on the New Macro button to start the process of building a macro in the Script Editor or Direct Editor. The first step prompts to select an activation method and assign a name to the macro. The next step opens the Script Editor to begin creating the macro. The Script Editor is the default editor. If desired, click on the Direct Editor icon while in the Script Editor to switch to the Direct Editor.
▶	Run Now	Click this button to run the macro that is highlighted in the Macro Express Pro - Explorer window.
	Edit Macro	Highlight a macro from the list of already created macros. Click on the Edit Macro button to open the Script Editor and display the macro. Or double click on the macro in the list to open it in the Script Editor.
●	Capture	Starts the process for recording or capturing a macro. Keystrokes and mouse clicks/movements are recorded and saved.
	Quick Wizards	Opens the Quick Wizards dialog. There are 25+ different wizards to provide step by step instructions for creating various macro types.
	Recycle Bin	Macro Express has its own recycle bin. Any macros deleted from the Macro Explorer are sent to the Recycle Bin. If a macro is deleted by mistake it can be restored to its original file. Or the macros in the recycle bin may be removed permanently.

Categories

Use the Categories option to assign macros to user defined categories. With large numbers of macros created, the categories provide an easy way to locate macros for further editing, refinement, exporting, etc.

For example create a Password category to hold any macros containing important passwords or assign any MS Word specific macros to a MS Word category. This makes it easy to quickly find the macro needed.



Categories

System Macros
macex.mex
Help
Responses
Tutorials
* Unassigned *
samples.mex
WorkFile.mex

Upon installation there are three default categories - System Macros, All Categories and Unassigned. These categories may not be renamed or deleted. Additional categories may be created as shown in this example. Since Macro Express allows more than one macro file to be open at a time, multiple sets of categories may be displayed - a set for each open file.

- ✓ **Note:** A macro may only be assigned to one category.

Macros

The Macros section displays the list of macros in the file or category that is currently open. When saving macros in the editors, quick wizards or with a macro capture, the program will return to the Macro Explorer. The list of macro information is displayed in six columns.

Index

The index column lists the macros in the order that they were created. The first macro created in the file displays as number 1, the second macro as number 2, etc.

Nickname

Optionally assign a nickname to any macro to help more readily identify the purpose of the macro. Any not assigned a nickname display with a blank field.

Activation

The Activation column displays the method selected to execute the macro. This column displays the hot key combinations, any timed macros, the letter combinations used for any shortkeys or any other type of activation method chosen. If more than one activation is assigned to the macro, this column displays Multiple for that macro. If the Show Preview Pane option is selected highlight the macro to view all of the activations in this pane.

Scope

This column lists the macro scope, whether it is global, global except, window specific, program specific, a system macro, a popup menu or a floating menu.

Category

This column lists the category to which the macro was assigned.

Modified

The Modified column displays the date and time that the macro was last modified.

Last Run

The Last Run column displays the date and time that the macro was last run.

Sorting Macros

To quickly sort the list of macros displayed, click on one of the column headers, Nickname, Activation, Scope, Category, Modified or Last Run. The macros will be sorted based on the column header selected. Click on the header a second time to reverse the sort order.

Right Click Context Menu

Right mouse click on a macro in the macro list to open the context menu. The menu options include Run Macro Now, Copy Macro, Edit Macro, Enable/Disable Macro, Recapture Macro, Delete Macro and Macro Properties.

Keyboard Shortcuts

Below is a listing of the Keyboard Shortcuts available in the Macro Explorer.

Keystroke	Action
Ctrl+F	Find
F3	Find Again
Delete	Delete highlighted macro
Ctrl+P	Print macro list

These keyboard shortcuts are used to navigate through the program. These keystrokes perform the same function as clicking on the Actions Icons in the left hand column of the program.

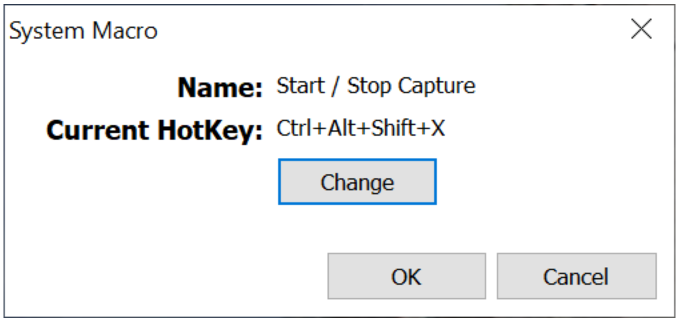
Keystroke	Action
Alt+E	Edit highlighted macro, when not in the Macro Recycle bi
Alt+E	Return to Macro Explorer from Recycle Bin
Alt+N	New Macro
Alt+P	Capture
Alt+Q	Quick Wizards
Alt+B	Recycle Bin

System Macros

Macro Express defines several system macros for a variety of program functions. The macros are stored in a System Macro file and are available with any macro file currently open. Each of the system macros is described below.

Reassigning a System Macro hot key

The system macros are assigned to hot keys but may be renamed at any time to something that is easier to remember. To rename a system macro, expand the System Macros category right mouse click on the macro. Left click on Properties from the menu to open the System Macro window as shown below. Click on the Change button and select another hot key combination. The new hot key will display in the Activations column of the Macro Express Pro - Explorer window.



System Macro

Name: Start / Stop Capture

Current HotKey: Ctrl+Alt+Shift+X

Macro Express Explorer > Open System Macros Category > Double Click on System Macro

Descriptions of the available system macros

System Macro	Default Hot Key	Description
Terminate Macro Express	Ctrl+Alt+Shift+V	Terminates Macro Express, Editor and Pl
Run Macro Express Editor	Ctrl+Alt+Shift+W	Activates the Macro Express - Explorer w Explorer program to the forefront from i minimized state.
Start/Stop Capture	Ctrl+Alt+Shift+X	Starts capturing keystrokes or mouse m macro. When finished capturing, press C to stop recording.
Resume Pause	Ctrl+Alt+Shift+Y	If a macro contains a Pause command p Ctrl+Alt+Shift+Y hot key to resume the Pause window.

Menu of Macros for Topmost Window	Ctrl+Alt+Shift+Z	Opens a popup menu listing all Window S Specific macros that pertain to the windo the hot key is pressed. For example, press the hotkey while not window and all Window or Program Spec notepad will be displayed in the menu. Optionally choose to display any macros Except that do not exclude the active wir display all global macros. Select to display the icons associated wit name of each macro in the menu, or the script.
Suspend / Resume Macro Express	Ctrl+Alt+Shift+U	Toggles between suspending Macro Expr active status. No macros will play back w suspended. The icon in the notification area is chang Macro Express is suspended.
Capture a Temporary Macro	Ctrl+Alt+Shift+T	This is a quick way to capture a macro fo assigning of hotkeys or checking settings macro is saved temporarily for playback
Play the Captured Temporary Macro	Ctrl+Alt+Shift+S	Starts the play back of the temporary m

File Menu

File Commands

File Menu

New File

Macro Express allows for creation of multiple files with different macros in each file. Click on the File > New File option to create a new file. A dialog box appears prompting for a file name and folder location.

Open File

Click on this option to open a previously created macro file. A dialog box appears prompting to select the file.

Close File

If more than one macro file is currently open, use this option to close one of the files. A dialog box appears prompting to select the file and choose from which folder.

Backup Macro File(s)

This option immediately backs up the macro file that is currently open. If using the automatic backup schedule, this option does disturb the schedule previously created. It uses the settings defined in the Backup Preference panel, such as where the backup is saved and how many backup copies are created.

Restore Macro File

Use this option to restore from a backup file. A Windows dialog will open prompting to select the backup file. The dialog opens to the folder designated in the Backup preferences panel for file backups. If the backup options were recently changed, then it may be necessary to browse to the location where previous backups were stored.

Performing the restore closes the file currently open and replaces it with the backup file selected.

Reopen

This is a convenient way to open a macro file previously used. This option displays the last several macro files opened. Click on the desired file to load into Macro Express.

This may close all other open macro files depending on the Close open macro files before loading another macro file preference setting selected.

Import

There are three different import options available in Macro Express. Click on the topic of interest below to obtain more information.

Import Macros
Import Text File
Import Playable Macro

Export

There are three different export options available in Macro Express. Click on the topic of interest below to obtain more information.

Export Macros
Output Macro Information
Export as Playable Macro

Print Macro List

Click on File > Print Macro List or click on the speed button to print a list of all macros currently assigned. The list contains the activation method, such as the hot key combination used, the nickname assigned, if any, and macro scope (global, system, window specific or program specific). The list includes the information as shown in the Macro Explorer window.

Printer Setup

Select this option to ensure printer settings are accurate before printing the macro list.

Terminate Macro Express Editor

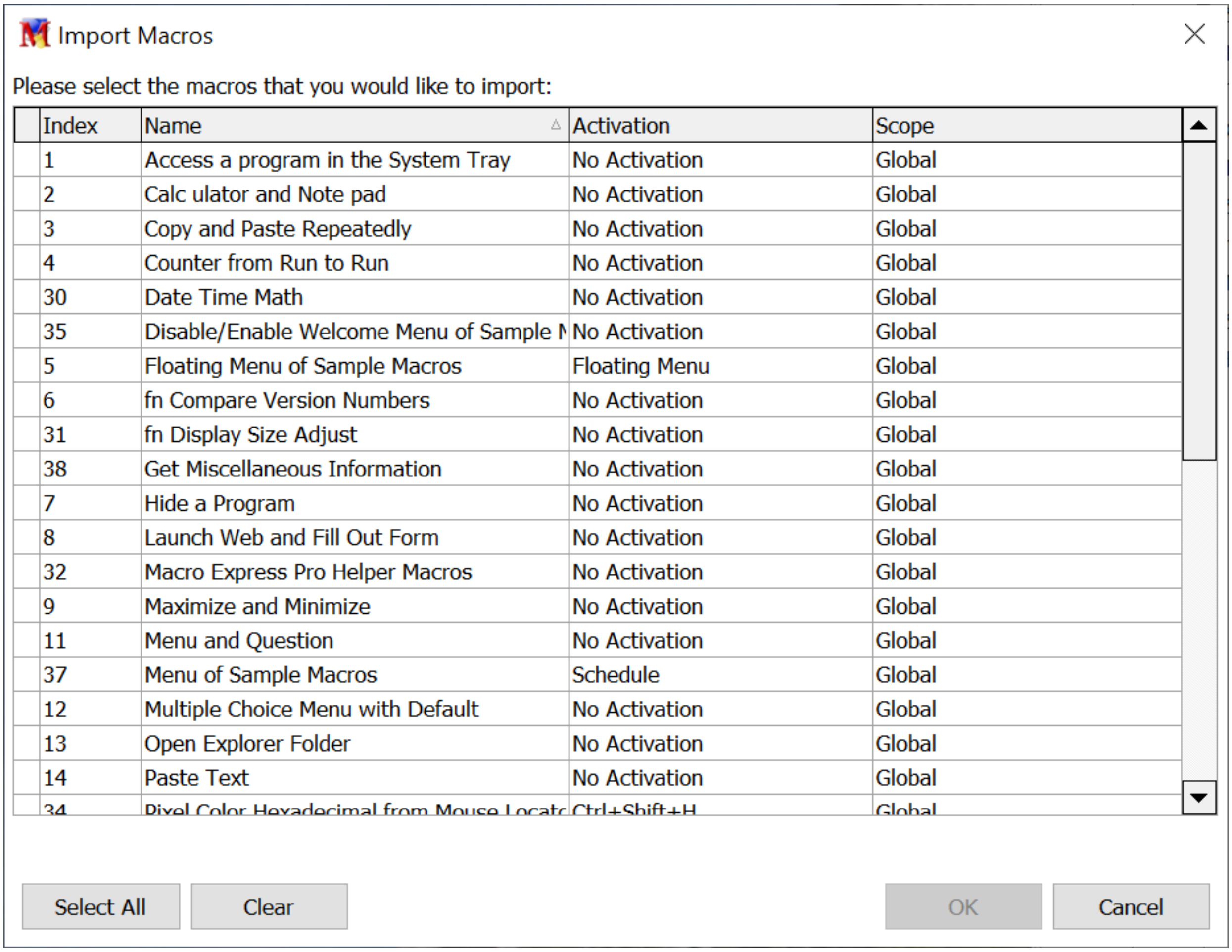
Selecting this option closes the Macro Express Pro - Explorer window but leaves the Player running. The macros still run with the Explorer window closed.

Terminate Macro Express Editor and Player

Selecting this option closes the Macro Express Pro - Explorer and Player. It shuts down the entire program, at which point no macros can run.

Import Macros

The import and export functions are convenient methods to transfer macros between Macro Express users. Use the Export command to save the macros to a disk or drive. On another computer import the macros from the disk, flash drive, or similar device.



Please select the macros that you would like to import:

Index	Name	Activation	Scope
1	Access a program in the System Tray	No Activation	Global
2	Calc ulator and Note pad	No Activation	Global
3	Copy and Paste Repeatedly	No Activation	Global
4	Counter from Run to Run	No Activation	Global
30	Date Time Math	No Activation	Global
35	Disable/Enable Welcome Menu of Sample M	No Activation	Global
5	Floating Menu of Sample Macros	Floating Menu	Global
6	fn Compare Version Numbers	No Activation	Global
31	fn Display Size Adjust	No Activation	Global
38	Get Miscellaneous Information	No Activation	Global
7	Hide a Program	No Activation	Global
8	Launch Web and Fill Out Form	No Activation	Global
32	Macro Express Pro Helper Macros	No Activation	Global
9	Maximize and Minimize	No Activation	Global
11	Menu and Question	No Activation	Global
37	Menu of Sample Macros	Schedule	Global
12	Multiple Choice Menu with Default	No Activation	Global
13	Open Explorer Folder	No Activation	Global
14	Paste Text	No Activation	Global
34	Pixel Color Hexadecimal from Mouse Locatc	Ctrl+Shift+H	Global

Macro Express Explorer > File > Import > Import Macros

To import macros, click on File > Import > Import Macros. A Select a File to Import window opens with a standard dialog box to browse through folders to find the correct file. All Macro Express macro files have a file extension of .mex. Highlight the file to import and click on Open. A list of all macros found in that file are displayed.

Please select the macros that you would like to import

Highlight the macros to import by clicking on each macro.

Select All

This option highlights all macros in the file so that all are imported.

Clear

This option clears any highlighted macros so that none are imported until a new selection is made.

Click on the OK button to start the import process. A progress bar displays during the import process until the Import Complete message appears. Click OK to return to the Macro Explorer. This finishes the process and the macros are now found in the current file.

✓ **Note:** If importing a popup menu, all of the macros contained in the menu must also be imported. Once imported, you will most likely need to rebuild the menu.

Import Text File

Selecting File > Import > Import Text File from the Macro Express Pro - Explorer menu imports a group of Shortkey or Hot Key macros. A standard Windows dialog box opens prompting to navigate and select the text file to import.

The text file must be in a specific format. An improper format will cause unpredictable results. The formatting of the lines in the text file is described below.

Line 1 - The first line requires an <ITEM NAME> tag followed by the shortkey or hotkey name. The hotkey name is the key sequence used and must be in the format found in the Create New Macro window. If a match is not made on one of these, then it is assumed that the name supplied is a shortkey.

Line 2 - This line contains the nickname assigned to the macro.

Line 3 - Line 3 contains the macro text and commands. This line may span multiple lines as shown in the examples below.

The <ITEM NAME> tag marks the start of a new macro in the text file.

Example

This example shows the format for two shortkey macros followed by a hot key macro to be imported.

```
<ITEM NAME>issadd
Our Address
Insight Software Solutions, Inc.
P.O. Box 106
Kaysville, UT 84037
<ITEM NAME>downm
Download Macro Express
You can download a 30 day evaluation copy of Macro Express
from www.macros.com/download.htm.
<ITEM NAME>Ctrl+Shift+m
ME Pro 6 Overview
Macro Express is the premier macro utility on the market.
With Macro Express, you can record, edit and play back
mouse and keyboard macros. Its powerful tools and robust
features will noticeably boost your productivity.
```

Import Playable Macro

A "Playable Macro" file format is available. Files with an .MXE file extension are similar to .EXE files. Double clicking on the .MXE file from within Windows Explorer plays back the macro.

Macro Express is not required to be running to execute the .MXE macro, though installation of the Macro Express player is necessary. The .MXE file may also be executed from a command line parameter to run a batch file.

Importing the macro

This option is used for importing a playable macro from another machine on a network or from a disk that contains a playable macro and converting it to a .MEX macro.

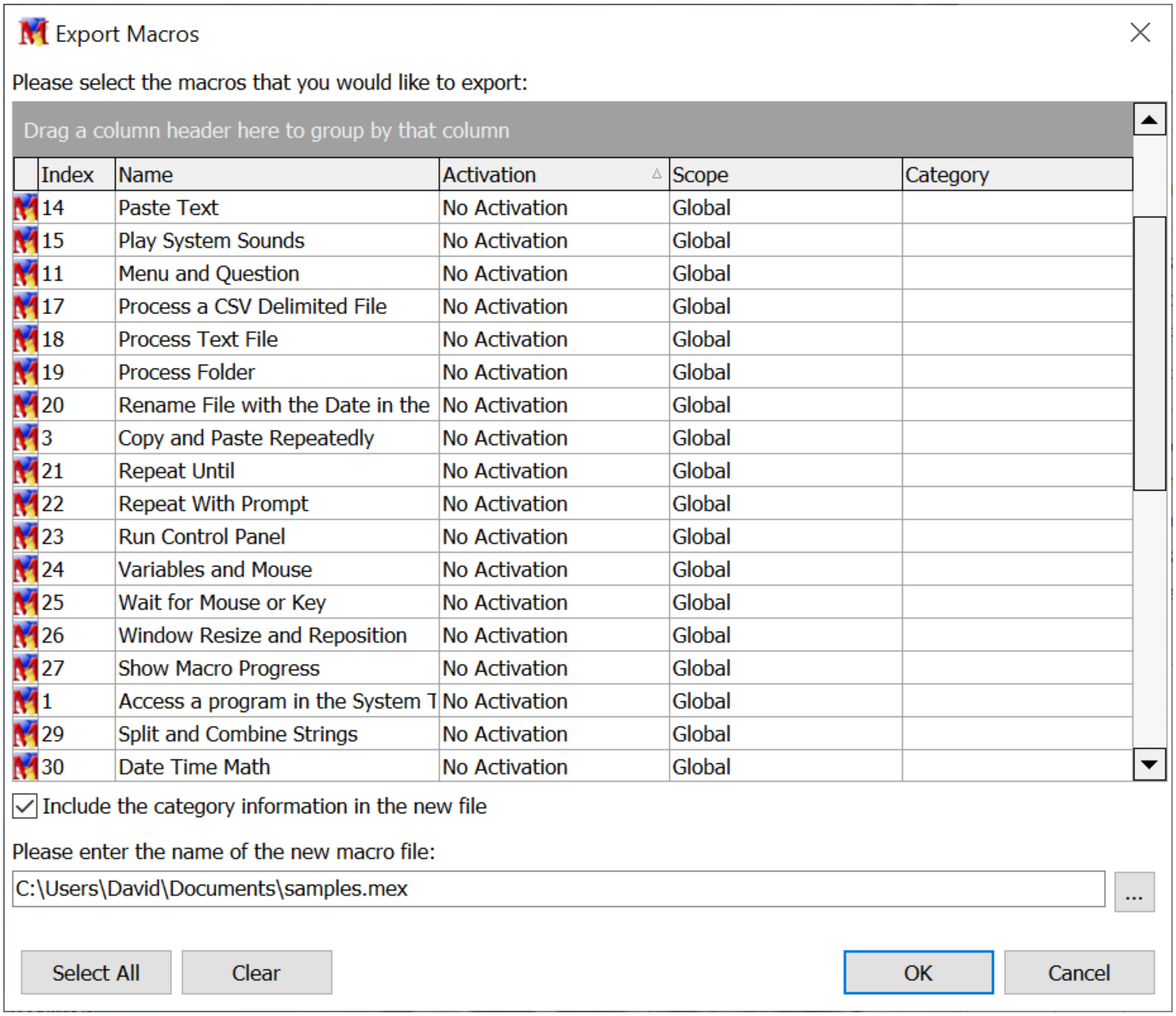
From the Macro Express - Explorer window click on File > Import > Import a Playable Macro. A dialog box prompts to search for the .MXE file to import. Once located, highlight the file to import and click OK to import the file.

The Copy Hot Key window opens prompting to assign a hot key or other method for executing the macro. The imported macro is then placed into the list of macros displayed in the Macro Express Pro - Explorer window.

Macro Express - Explorer > File > Import > Import a Playable Macro

Export Macros

The import and export functions are a convenient way to transfer macros between Macro Express users. Use the Export command to save macros to a disk or drive. On another computer Import the macros from the disk, flash drive, or similar device.



Please select the macros that you would like to export:

Index	Name	Activation	Scope	Category
14	Paste Text	No Activation	Global	
15	Play System Sounds	No Activation	Global	
11	Menu and Question	No Activation	Global	
17	Process a CSV Delimited File	No Activation	Global	
18	Process Text File	No Activation	Global	
19	Process Folder	No Activation	Global	
20	Rename File with the Date in the	No Activation	Global	
3	Copy and Paste Repeatedly	No Activation	Global	
21	Repeat Until	No Activation	Global	
22	Repeat With Prompt	No Activation	Global	
23	Run Control Panel	No Activation	Global	
24	Variables and Mouse	No Activation	Global	
25	Wait for Mouse or Key	No Activation	Global	
26	Window Resize and Reposition	No Activation	Global	
27	Show Macro Progress	No Activation	Global	
1	Access a program in the System T	No Activation	Global	
29	Split and Combine Strings	No Activation	Global	
30	Date Time Math	No Activation	Global	

☑ Include the category information in the new file

Please enter the name of the new macro file:

C:\Users\David\Documents\samples.mex

Macro Express Explorer > File > Export > Export Macros

Please Select the Macros that you Would Like to Export

To export macros, click on File > Export > Export Macros. A window opens showing all macros from the macro file currently in use. Highlight the macros to export.

Select All

This option highlights all macros in the file so that all are exported.

Clear

This option clears any highlighted macros so that none are exported until another selection is made.

Include the Category Information in the New File

This option saves the category information in the new file if the macros selected for export have been assigned to categories.

Please Enter the Name of the New Macro File

Enter the name and file path of the file where the macros will be exported. Or click on the browse button to navigate to the location of the file to be saved.

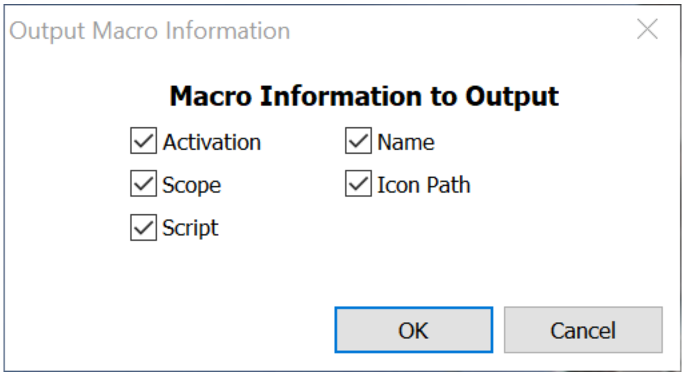
Click OK to export to the new file. The file is saved as a Macro Express macro (.mex) file which can be opened directly by Macro Express using the File > Open File menu or the macros may be imported.

- ✓ **Note:** If exporting a popup menu, all of the macros contained in the menu must also be exported. Most likely the menu will need to be rebuilt in the new file.

Output Macro Information

Use this option to select what information from the macro file to save to a text file. As shown in the window below, select which items to output and click OK. A standard dialog box prompts for a file name and folder destination for storing the data.

The newly generated text file contains selected information about each macro in the file.



Output Macro Information

Macro Information to Output

☑ Activation ☑ Name
☑ Scope ☑ Icon Path
☑ Script

Macro Express Explorer > File > Export > Output Macro Information

Macro Information to Output

Activation - Lists the Hot Key, Shortkey or other activation of the macro.

Scope - Lists the scope of the macro.

Script - Displays the macro script in Direct Editor format.

Name - Lists the name assigned to the macro.

Icon Path - Lists the names of the icons used.

The information saved may be printed in any text editor.

Export Playable Macro

A "Playable Macro" file format is available in Macro Express. Files with an .MXE file extension are similar to .EXE files. Double clicking on the .MXE file from within Windows Explorer, My Computer or the desktop plays back the macro.

Macro Express is not required to be running to execute the .MXE macro, though installation of the Macro Express player is necessary. The .MXE file may also be executed from a command line parameter to run a batch file or similar.

Exporting the macro

The Export as Playable Macro option exports (creates) any defined macro as a playable .MXE macro. From the Macro Express Pro - Explorer window highlight the macro to export and click on File > Export > Export as Playable Macro. Or right click on the macro and select Export > Export as Playable from the context menu.

A dialog box prompts for a file name and folder destination for storing the .MXE file. By default, the macro nickname is inserted as the file name but may be changed if desired.

Windows does not allow the following characters in a file name: | ? : \ / " < > and *. If the macro nickname contains any of these characters, they will be deleted from the default .MXE file name.

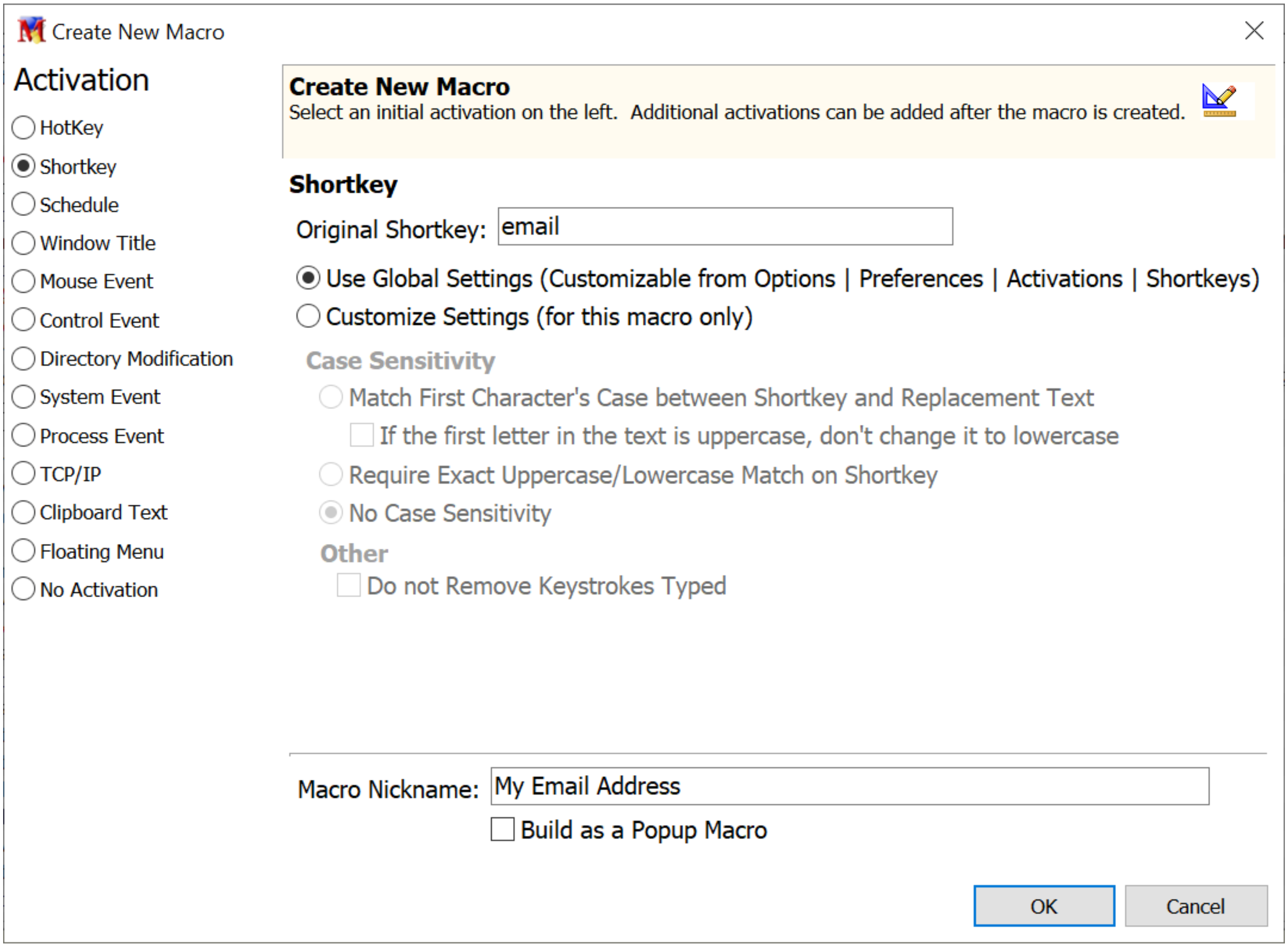
- ✓ **Note:** Any activation assigned to the original macro, such as a hot key, schedule or other, is not stored in the playable macro. Double click on the .MXE file to activate the macro.

- ✓ **Note:** Do not include any Macro Run commands in the playable macro. The macro from the Macro Run command is not compiled into the playable macro and will cause an error when running.

Highlight macro in Macro Express - Explorer > File > Export > Export as Playable Macro

New Macro

To create a new macro, click on Macro > New Macro from the menu or click on the New Macro icon to open the Create New Macro window. The first step is to select the type of macro activation needed. The hot key activation is the default selection. After assigning the activation type, enter a macro nickname and press OK to proceed to the Script Editor.



Create New Macro

Activation

○ HotKey
● Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text
○ Floating Menu
○ No Activation

Create New Macro
Select an initial activation on the left. Additional activations can be added after the macro is created.

Shortkey

Original Shortkey: email

● Use Global Settings (Customizable from Options | Preferences | Activations | Shortkeys)
○ Customize Settings (for this macro only)

Case Sensitivity

○ Match First Character's Case between Shortkey and Replacement Text
□ If the first letter in the text is uppercase, don't change it to lowercase
○ Require Exact Uppercase/Lowercase Match on Shortkey
● No Case Sensitivity

Other

□ Do not Remove Keystrokes Typed

Macro Nickname: My Email Address

□ Build as a Popup Macro

Macro Express Explorer > New Macro Icon or Macro > New Macro

Hot Keys

To activate a new macro with a hot key, make sure the Hot Key choice is selected. Place the cursor in the edit box and hold down the Hot Key combination to use. The combination pressed is placed in the edit box. When the macro is complete, press this hot key combination to activate the macro.

Or scroll through the list of keys and highlight the one wanted. This also places the hot key combination in the edit box.

- ✓ **Note:** Use the hot key combinations displayed in red with caution. These combinations may conflict with basic Windows functionality, such as the clipboard copy (Ctrl + c).

Shortkeys

Select the Shortkey option and type in the shortkey (up to a 32 letter code) to use to activate the macro. A shortkey is a set of characters such as abc, myname, etc. See the program specifications for a complete list of characters accepted as shortkeys.

The macro runs when the characters are typed into an application that accepts text input. The Shortkeys activation will not trigger a macro from the Windows desktop or other applications that do not accept text input.

- ✓ **Note:** Before using the Shortkeys feature click on Options > Preferences > Activations > Shortkeys tab to define the settings.

Scheduled Macro

Selecting Schedule activates the Scheduler options. Select the schedule wanted and click OK to build the macro in the Script Editor. The macro then plays back at the time designated.

Window Title

This option instructs a macro to execute any time the selected Window Title has focus. In other words, the macro runs once the window is activated.

In the edit box, enter the title of the Window that will start the macro. The title of the window is that which appears in the caption bar of a window. Macro Express examines all Window Titles running on the computer in an attempt to find a match. The Window Title does not need to be an exact match.

A partial match also works. For example, "notepad" or "notep" would be sufficient for the Windows Notepad program. This is particularly useful as some Window Titles change based on which file is open.

Case sensitivity is not required. For example, "NOTepad" works as well as "notepad".

Mouse Event

This option uses mouse clicks to activate macros. Two different methods are available.

1. Define regions on the screen that can be assigned to a mouse activation. Clicking on the defined area of the screen runs the macro.

2. Click on a specific part of a window to execute the macro. Window parts include the window border, window menu, window caption bar, horizontal scroll bar, vertical scroll bar, system button, minimize button and maximize button.

- ✓ **Note:** A few options have been excluded where they will interfere with essential functionality of the windows such as Left Mouse with the Maximize or Minimize buttons.

Macros may be assigned to the left, middle or right mouse button and may have a scope of global, window or program specific.

Control Event

The Control option activates a macro when a specific Window control becomes active or the control's window becomes active. The control may be a window button, edit box or another identifiable portion of a window.

Directory Modification

The Directory Modification activation looks for changes to files or folders. When a change is made to the file or folder specified, the macro will start. Among the options, the macro may look for changes to file size, attributes, write time or if a file has been renamed or deleted.

System Event

The System Event activation looks for changes to the system such as an incoming message, resolution change or system color change. When the specified system change occurs, the macro starts.

Process Event

The Process Event activation method starts a macro when a Process starts or shuts down.

TCP/IP

The TCP/IP activation monitors when an Internet or Network connection has been made or released.

Clipboard Text

The Clipboard Text activation method starts a macro based on the contents of the clipboard. Select whether the clipboard does or does not match the text supplied in the edit field. Or choose whether the clipboard contains or does not contain the text.

Floating Menu

A floating menu has the same form and options as a Popup Menu except that it always remains on top and is always accessible as long as the Macro Express Player is running. A floating menu can be minimized to just the caption bar to free up space if needed. Multiple floating menus can be created. The last location of the floating menu is retained and used with subsequent runs of Macro Express. Use the Build Macro Menu window to select which macros to include in the floating menu.

None

This option does not assign an activation to the macro. There are at least two possible uses for the No Activation option. First, create generic macro templates that can be incorporated into other macros - a set of instructions that may be useful for insertion into other macros. Use the Macro Run command to insert this type macro into other macros.

A second use for this option is to place the macro in a popup or floating menu. A macro activation is not necessary if the macro is part of a popup or floating menu. The only way to invoke the macro would be through the menus.

Macro Nickname

Assign a name to the macro to help remember what the macro does. The nickname, along with the activation, is displayed in the Macro Explorer and is optional.

Build as a Popup Menu

Selecting a Popup Menu activates the Menu Builder. After selecting the activation, check the Build as a Popup Menu box and click OK to open the Menu Builder window. In this window select which macros to include in the popup menu. When activated, the popup menu displays all macros included in the menu. Choose which one to run from the menu.

Capture Macro Introduction

The simplest type macro to create is a recorded macro. The macro records each keystroke, mouse click or mouse movement made. After saving the macro, play back the macro and watch it repeat each of the steps just recorded. If a mistake is made in the recording, either re-record the macro or edit the macro with the Script Editor.

The first step before capturing a macro is to select capture options, such as whether to record mouse movements, playback speed, base the mouse position relative to the position of the window or entire screen, etc. Use the Options > Preferences > General > Capture menu to define the default capture settings.

Each time a capture is started, the options selected are stored as the default values. If a change to one of the settings is necessary for a particular recording, make the change before the capture starts.

There are two different ways to initiate a macro capture. Examples for using both of these methods are outlined in the Tutorial section of the Help file and also in this section for convenience.

Assign Macro Activation First

Click on the Capture Icon or Macro > Capture Macro from the menu to start the capture process. This method requires selecting the macro activation and reviewing or changing the default capture options before starting the capture.

Assign Macro Activation After Capture

The second method uses the System Macro, Ctrl+Alt+Shift+X. While in another application, press the System Macro to begin capturing a macro in that application. This method uses the capture default settings that were defined in the Preferences section. After the capture is complete, follow the prompts to assign a macro activation.

Quick Wizards

The Quick Wizards are a convenient way to step through the macro creation process. The sequence is essentially the same for each of the wizards and are basically self explanatory. We'll briefly discuss the basic steps of the wizard.

Wizard Steps

1. Select the category or type of macro to create and then click on the Next button.
2. The second step prompts to select the specific type of macro to create. Click Next to advance.
3. A window opens listing the functions of the wizard selected as well as giving some examples of macros that may be created with this wizard. Click Next to advance to the next screen.
4. Depending on the wizard selected, a prompt appears asking what to do with the macro or instructions are provided to begin building the macro. Input the necessary information and click next to proceed.
5. After entering the necessary information for the macro, the next step is to select a method for executing the macro - hot key, timed macro, etc. Select an activation method and click Next.
6. Define if the macro should play back in all programs, a single program or only in a specific window. Make a choice and click Next to proceed.
7. Give the macro a nickname and optionally assign an icon with the macro.
8. Click Finish and the macro is complete.

These are the basic steps of the Wizard. Depending on the wizard selected there may be several additional steps, each of which is self explanatory. Each step has a Cancel button to exit the Wizard at any time. Each step includes a Back button to backtrack and make any changes needed.

The Wizards have been divided into categories based on their functionality. Each of the links below lead to brief explanations of each of the wizards and some of the types of macros that can be created. The Tutorial section includes an example of creating a macro with the Wizard.

Text
Internet and Networking
Multimedia
System and Files

- ✓ **Note:** The Quick Wizards are handy for less complicated macros. If planning to use If / Then logic, variables or other advanced features, the macro will need to be created with the Script Editor.

Edit Macro

There are several ways to edit an existing macro. In the Macro Express Pro - Explorer window highlight the macro to be edited. Then take one of the following steps:

1. Click on the Edit Macro icon
2. Click on Macro > Edit Macro from the menu
3. Double click on the macro in the Explorer list
4. Right click on the macro and select Edit Macro from the context menu

The above action opens the Script Editor where changes may be made to the macro script.

Highlight macro in Macro Express - Explorer, then click on Macro > Edit Macro

Copy Macro

Macro Express provides for a simple method to copy an existing macro. From the Macro Express Pro - Explorer window highlight the macro to copy. Then do one of the following:

1. Click on the Copy Macro button
2. Click on Macro > Copy Macro(s) from the menu.
3. Right mouse click on the macro and left click on the Copy Macro menu item.

The Create New Macro window appears prompting to select a new macro activation for the copied macro. Select the macro activation, change the macro nickname if desired and click OK to save. The new macro is now displayed in the Macro Explorer window.

If the macro nickname is not changed it will be displayed as "copy of" followed by the name of the original macro.

Highlight macro in Macro Express - Explorer, then click on Macro > Copy Macro

Enable or Disable Macro

A disabled macro will not execute, but the definition is retained for potential future use. This is helpful for those instances where a particular macro is not needed for a time, but may be needed later. Instead of deleting it and re-creating the macro later, simply disable it and then enable in the future.

Disable a macro

To temporarily disable a macro, highlight the macro from the list in the Macro Express Pro - Explorer and do one of the following:

1. Click on Macro > Disable Macro from the menu
2. Right click on the macro and select Disable Macro from the context menu

Disabled macros appear in the Macro Explorer with italicized red text to distinguish from active macros. A disabled macro will not function again until it has been Enabled.

Enable a macro

To enable a disabled macro, highlight that macro in the Macro Express Pro - Explorer window and do one of the following:

1. Click on Macro > Enable Macro from the menu
2. Right click on the macro and select Disable Macro from the context menu

Floating Menus

When closing a floating menu by clicking on the "X" in the upper right corner of the menu window, the floating menu is disabled. To enable the menu highlight the floating menu from the list of macros in the Macro Explorer and click on Macro > Enable Macro.

Macro Disable and Macro Enable Commands

Also use the Macro Disable and Macro Enable commands in macros themselves to enable and disable other macros in the macro file - such as Floating Menus or Scheduled macros.

Highlight macro in Macro Express - Explorer, then click on Macro > Enable Macro/Disable Macro

Recapture Macro

To overwrite the results of a previous captured macro, highlight the macro from the Macro Express Pro - Explorer list and do one of the following:

1. Click on Macro > Recapture Macro from the menu.
2. Right click on the macro and select Recapture Macro from the context menu.

Either action opens the Capture a Macro window to check the capture settings before recording. Then click the Capture button to start the macro capture.

Stopping the Capture

To stop the capture take one of the following actions:

1. Press the system hot key Ctrl + Alt + Shift + X
2. Right click on the camera icon in the notification area
3. Press the hot key combination of the macro being recaptured

Recapturing removes the contents of the existing macro and replaces it with the newly captured script.

Highlight macro in Macro Express - Explorer, then click on Macro > Recapture Macro

Delete Macro

To remove a macro that is no longer needed, first highlight that macro from the list in the Macro Express Pro - Explorer.

There are several ways to delete the highlighted macro.

1. Click on the Delete Macro button ✗
2. Click on Macro > Delete Macro(s) from the menu.
3. Press the Delete key on the keyboard.
4. Right mouse click on the macro and left click on the Delete Macro menu item.

A dialog window opens to verify that the macro should be deleted. Select Yes to confirm the deletion or No to cancel.

For mass deletions highlight a group of macros and delete them all at the same time.

Deleted macros are sent to the Macro Express Recycle Bin. If necessary a deleted macro can be restored to its original macro file. Or periodically empty all macros from the Recycle Bin.

- ✓ **Note:** To bypass the Macro Recycle Bin highlight the macros to be deleted and press Shift + Delete. This permanently deletes the macros without placing them in the Macro Recycle Bin.

Macro Delete Command

The Macro Delete command deletes other macros during macro playback. Insert the command at the end of a macro script and the macro can delete itself upon completion of the playback.

This feature is handy for use with one-time reminder messages. The macro runs at its appointed schedule, displays the reminder and then is deleted upon completion. The reminder macro doesn't need to remain in the Macro Explorer list after it has run.

Highlight macro in Macro Express - Explorer, then click on Macro > Delete Macro

Run Macro Now

This option does what it's title implies. It runs the macro selected. There are several ways to use this feature.

First Highlight the macro in the Macro Express Pro - Explorer window.

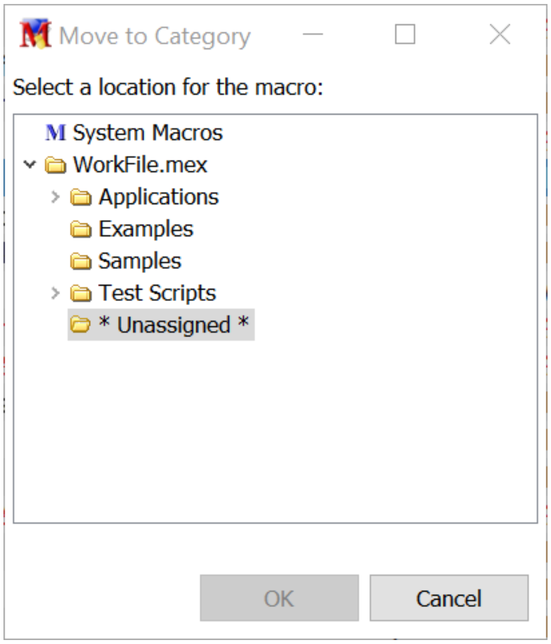
1. Click on the ▶ icon.
2. Click on Macro > Run Macro Now from the menu.
3. Right click on the macro to open the context menu and left click on the Run Macro Now menu item.

Of course, make sure that the macro to "Run Now" is appropriate for the window currently open, etc.

Highlight macro in Macro Express - Explorer, then click on Macro > Run Macro Now

Move to Category

Use this menu command to assign macros to categories already created. Highlight the macro to be assigned and click on Macros > Move to Category to display a list of categories such as in the example below. Highlight the category to move the macro into and press the OK button. The macro is now assigned to the category selected.



Macro Express Explorer > Macro > Move to Category

The other method is to drag the macro from the Macro Express Pro - Explorer list to the desired category. To drag the macro, highlight the macro, hold down the left mouse button and move the pointer to the category listed in the Category column. When the mouse is pointing to the correct category, release the left mouse button. The macro is assigned to the category. Expand the category to view all macros stored in that category.

With either of the above options, also highlight a block of macros and move them all at once to the category of choice.

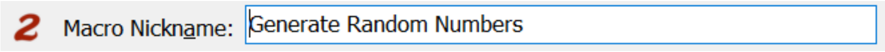
- ✓ **Note:** The categories selected are specific to the macro file that is currently open. When opening a new macro file, the categories created for the new file are displayed. If more than one macro file is open, multiple sets of categories are displayed, one set for each macro file.

Place on Desktop

Use this option to place an icon of any macro onto the desktop. In order to do this, highlight the macro from the list of macros in the Macro Express Pro - Explorer. Then click on Macros > Place on Desktop. The icon previously assigned to the macro is displayed on the desktop.

Selecting a Desktop Icon

Change a macro's icon by double clicking on the macro from the list of macros in the Macro Explorer to open the Script Editor. The icon in the example below, a red number two, is displayed to the left of the macro nickname.



Macro Nickname: Generate Random Numbers

- Click on the icon to open the Select an Icon window.
- Click on the browse button to open a folder with a number of icon options.
- Or browse to another location on the hard drive where other icons may be stored.
- Choose the icon desired.

If an icon was not selected at the time the macro was created, the default Macro Express icon is displayed.

- ✓ **Note 1:** When creating an icon on the desktop, the macro nickname also becomes the file name. Windows does not accept the following characters in a file name: / \ : * < > | ? and *.

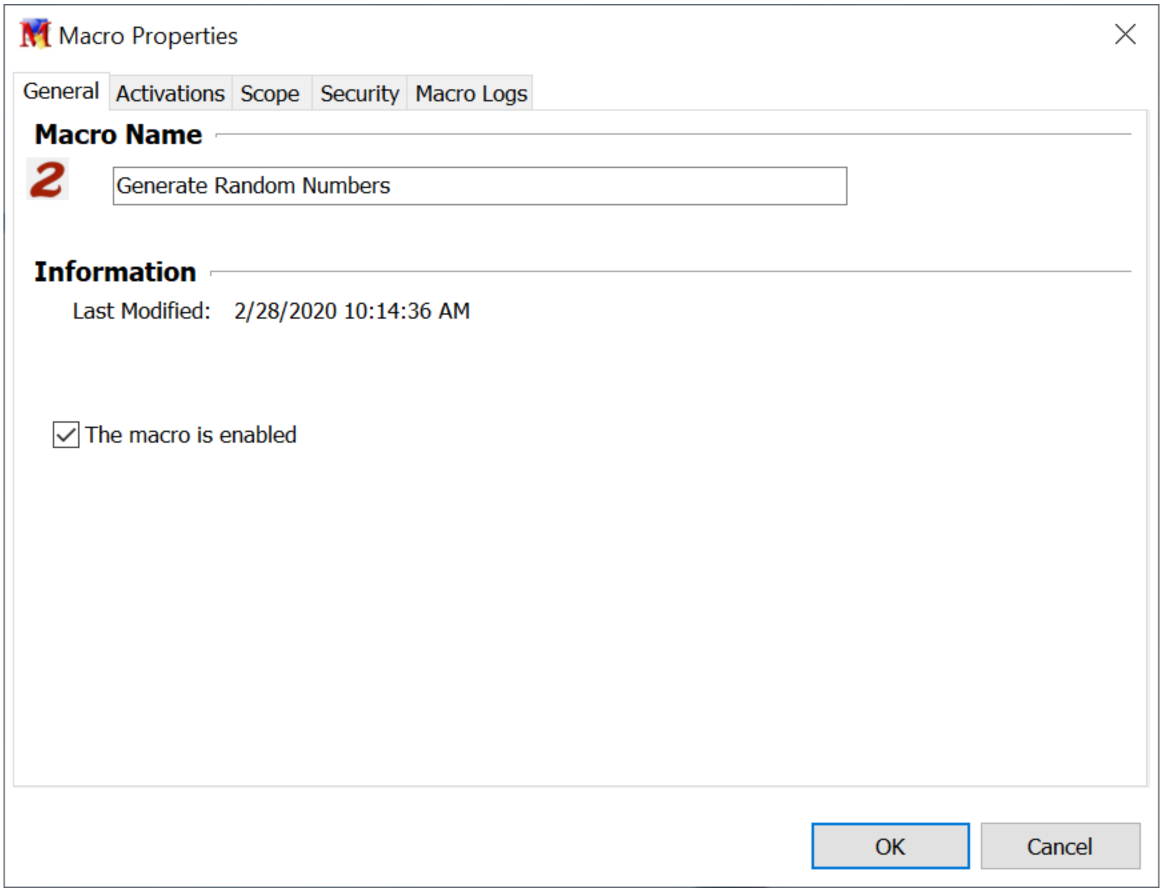
- ✓ **Note 2:** To remove an icon from the desktop, right mouse click on the icon and select delete from the menu that opens.

Highlight macro in Macro Express - Explorer, then click on Macro > Place on Desktop

Macro Properties

This view of the Properties window may be accessed from the Macro Express Pro - Explorer in a couple of ways.

1. Highlight the macro and then click on Macros > Properties from the Macro Explorer menu
2. Right click on the macro and left click on Macro Properties from the context menu.



Macro Name

Generate Random Numbers

Information

Last Modified: 2/28/2020 10:14:36 AM

☑ The macro is enabled

Macro Express Explorer > Right Click on a Macro > Left Click on Macro Properties

The following information is available about the macro selected.

General

This Tab provides some basic information about the macro such as the last modified date or scheduled time if a scheduled macro. Click on the check box to change the status of the macro from Enabled to Disabled or vice versa. This panels also allows editing of the Macro Name.

Activations

This tab displays what activation(s) are in use for the macro. The activation(s) may be modified from this screen.

Scope

The Scope tab displays the macro scope and any programs or windows assigned if using Program or Window Specific scopes. The selections may be modified from this panel.

Security

The Security Tab allows for changes to the existing passwords assigned to the macro or to add a password.

Logging

The Logging Tab displays any error logging that the macro is defined to use. The logging options are set in Preferences > Playback > Miscellaneous and will log errors or will log all commands in the macro, depending on the option selected.

The field in this tab is not editable. Click on the Clear Log button to remove all log entries.

Macro Recycle Bin

The Macro Express Recycle Bin works similar to the Windows Recycle Bin. When a macro is deleted from the Macro Express Pro - Explorer, the macro is removed from the macro file that is currently open. The deleted macro is sent to the Macro Express Recycle Bin and is displayed in the bin along with any other macros previously deleted from the current file.

- ✓ **Note:** Highlight the macro or macros in the Macro Explorer window and press Shift + Delete to permanently delete the macros and bypass the Macro Recycle Bin. This step permanently deletes the macros with no chance for recovery.

Three options are available in the Recycle Bin - Restore Macros, Delete Macros or Empty Recycle Bin.

Restore Macro(s)

The Restore Macros option returns a deleted macro from the recycle bin back to its original file. The macro is again available for use in the Macro Express - Explorer. This is a quick way to restore accidentally deleted macros.

To restore, highlight one or more macros in the Deleted Macro list. To highlight a block of macros, click on the first macro to restore, hold down the Shift Key and click on the last macro in the block. Click on the Restore Macro(s) option to place the deleted macros back into the macro file from which they were deleted.

Delete Macro(s)

Delete Macros permanently removes the selected macros from the Recycle Bin with no chance to retrieve later. To remove, highlight one or more macros in the Deleted Macro list. To highlight a block of macros, click on the first macro to delete, hold down the Shift Key and click on the last macro in the block. Click on the Delete Macro(s) option to remove from the Recycle Bin.

Empty Recycle Bin

Click on the Empty Recycle Bin option to remove all macros from the Recycle Bin. These macros cannot be retrieved once they are removed from the Recycle Bin.

Using Categories

Use this option to assign macros to categories. For example create a Password category to hold any macros containing important passwords or assign any MS Word specific macros to a MS Word category. The categories provide a simple way to find macros after they have been created. A macro may only be assigned to one category.

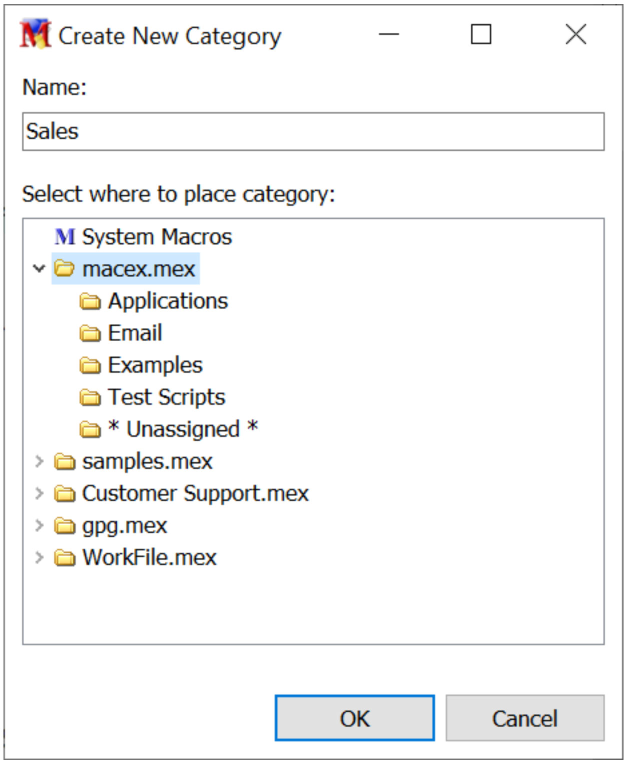
When the program is first installed, there are two default categories - All Categories and Unassigned which may not be changed. But new categories may be added, renamed or deleted.

- All Categories displays all of the macros in the file.
- The Unassigned category displays all macros not assigned to a sub-category of All Categories.

The category icons may be turned on or off via the Macro Express Pro - Explorer menu Options > Preferences > Appearance > Explorer tab.

New Category

Select this menu option to add a new category. A new category folder and edit box appears in the category list. Type in the name to assign the category. Also create sub-categories within the category list. To do this, right click on the main folder and then left click on New Category. A new category sub-folder is added to the list.



Name: Sales

Macro Express Explorer > Category > New Category

Rename Category

To change the name of a category already created, highlight the category and click on the Category > Rename Category menu option. The category name is highlighted so that a new name may be inserted, similar to the way Windows Explorer works.

Enable/Disable Category

This option disables or enables all of the macros in the category selected. If the category contains both enabled and disabled macros, then the status is toggled. The enabled macros are now disabled and the disabled macros in the category are enabled.

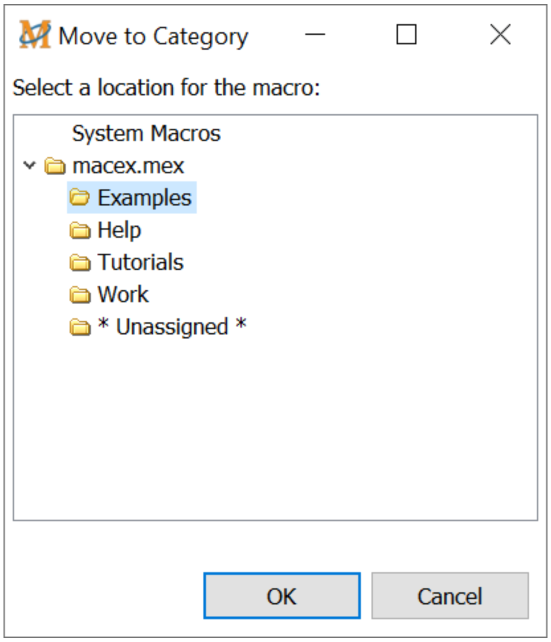
Delete Category

Highlight the name of the category to remove. Then click on Category > Delete Category. A warning message appears verifying that the deletion should continue. Click Yes to delete the category and all of it's sub-categories.

Placing Macros in Categories

Once a category is created, there are two methods for moving macros into a category.

Highlight the macro to be assigned and click on Macros > Move to Category to display a list of categories such as in the example below. Choose the category to move the macro into and press the OK button. The macro is now assigned to the category selected.



Macro Express Explorer > Macro > Move to Category

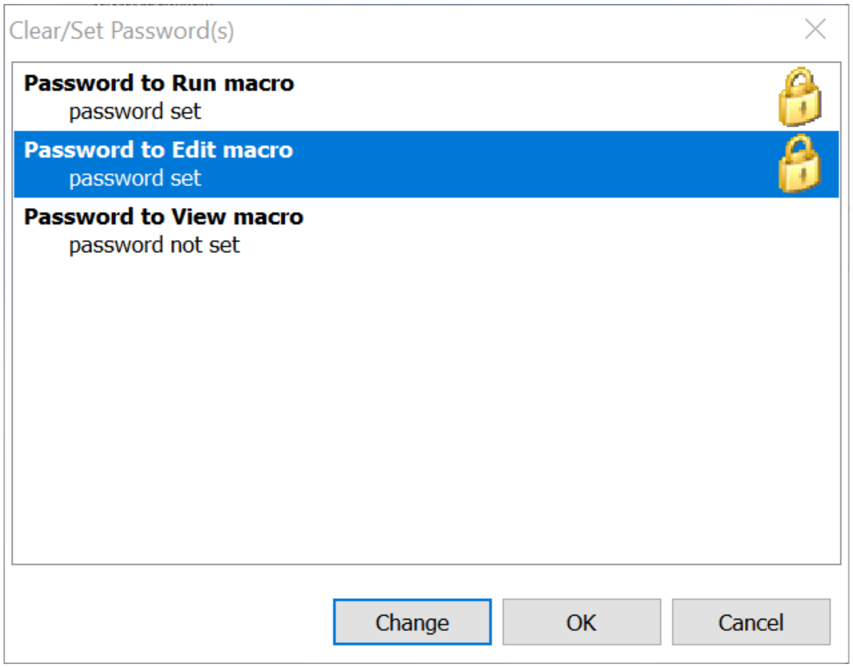
The other method is to drag the macro from the Macro Explorer list to the desired category. To drag the macro, highlight the macro, hold down the left mouse button and move the pointer to the category listed in the Category column. When the mouse is pointing to the correct category, release the left mouse button. The macro is assigned to the category. Expand the category to view all macros stored in that category.

Using either of the above methods, highlight a block of macros and move them all at once to the category of choice.

- ✓ **Note:** The categories selected are specific to the macro file that is currently open. When opening a new macro file, the categories created for the new file are displayed. If more than one macro file is open, multiple sets of categories are displayed, one set for each macro file.

Set/Clear Password(s)

This menu option provides password protection to a category of macros. Select whether to prevent users from running, editing or viewing macros in the selected category. Selecting All Categories effectively password protects the entire macro (.mex) file. The Lock icon indicates that a password is currently in place.



Password to Run macro
password set

Password to Edit macro
password set

Password to View macro
password not set

Macro Express Explorer > Category > Set/Clear Password(s)

Run Password

None of the macros in the password protected category will run without first typing the password. When a macro is activated, a prompt appears requesting the password be input. Once the password is successfully entered, the macro starts running.

Edit Password

The macros in the password protected category are displayed, but may not be edited without first typing the password.

View Password

None of the macros in the password protected category are displayed without first typing the password. Once the password is successfully input, the macros in the category may be viewed, edited, saved or deleted. Once another category is opened, the password must be typed again to reenter the password protected category.

Create a New Password

To create a password for the first time select the Run, Edit or View option and click on the Change button. Leave the Original Password field empty and enter the password in the New Password field. Confirm the password in the Confirm Password field and save.

Change an Existing Password

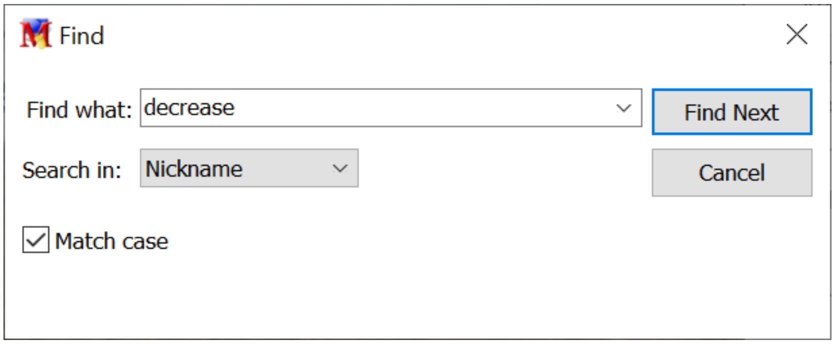
To change an existing password select the Run, Edit or View option and click on the Change button. Enter the current password in the Existing Password field. Insert the new password in the New Password field. Confirm the password in the Confirm Password field and save.

- ✓ **Note:** Leave the New Password and Confirm Password fields blank if removing a password from the category.

Search

Find

Use the Find feature to search through the list of macros in the Macro Express Pro - Explorer window. It performs the search beginning with the macro highlighted and continues through the rest of the list. Pressing CTRL + F or clicking on Search > Find will open the Find window.



Find what: decrease

Search in: Nickname

☑ Match case

Macro Express Explorer > Search > Find

Find what:

Enter the details of the search (what is to be found) in this field.

Search in:

Perform the search on the Nickname, Activations, Scope or Modified columns in the Macro Explorer window. Select which column to search through.

After entering the search criteria press the Find Next button. If a match is found, the macro that matches is highlighted. If no match is found, a message "Reached the end of the list" is displayed.

Match case

This option makes the search case sensitive. Capital and lower case letters need to match exactly in order to find the right match.

Find Again

If the first match is not the macro being sought, press F3 or click on Search > Find

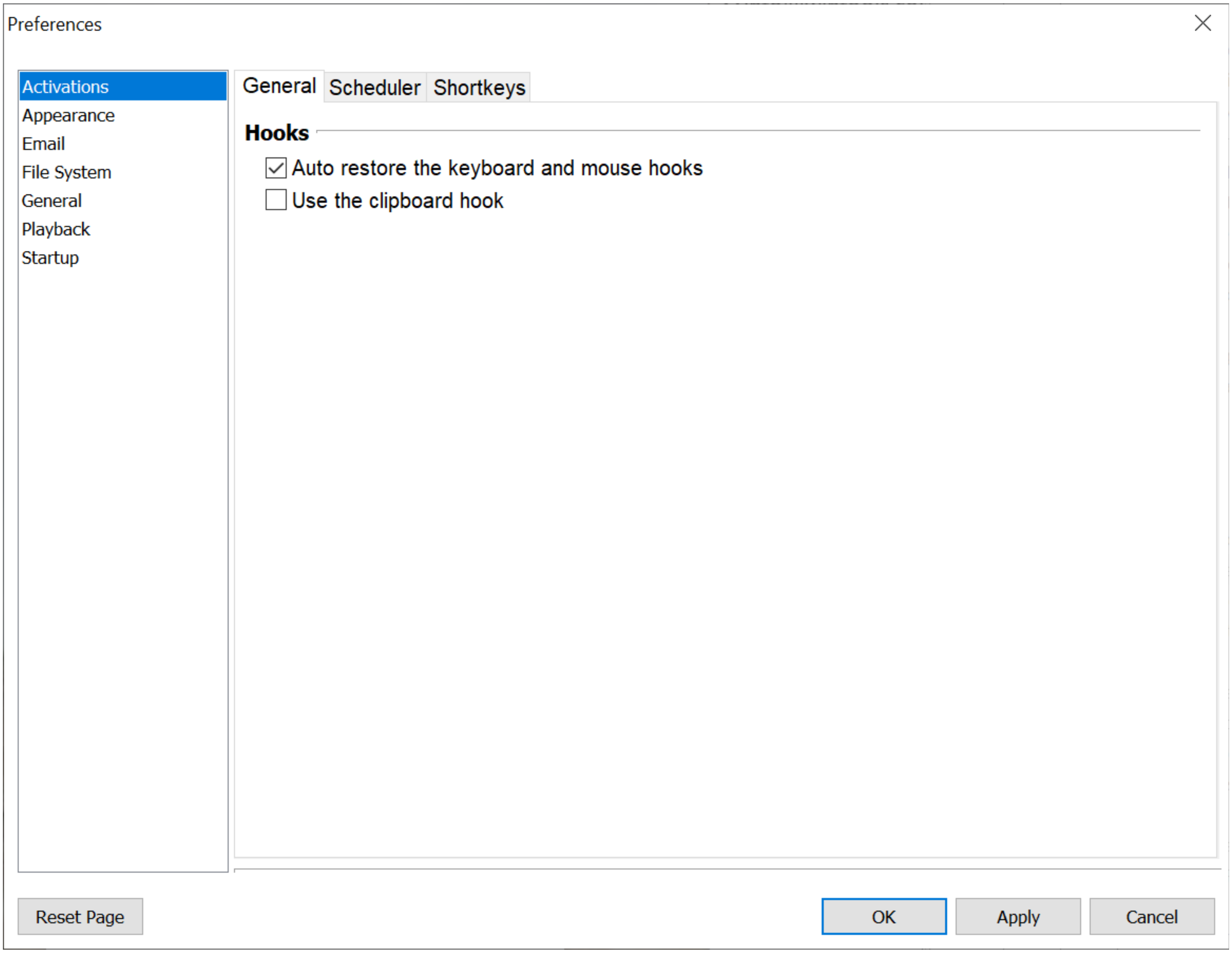
Again to go to the next match.

Options

Preferences

Activations

General Activation



Hooks

☑ Auto restore the keyboard and mouse hooks

☐ Use the clipboard hook

Macro Express Explorer > Options > Preferences > Activation > General Tab

Auto restore the keyboard and mouse hooks

This feature automatically attempts to have Macro Express re-assert itself as the first program to take a look at the keystrokes passed through the Windows system wide keyboard hooks even if other programs are loaded that also use these keyboard hooks.

Typically this is not a problem unless a program is run *after* Macro Express and it uses the keyboard hooks and it places itself there in an improper fashion. When this occurs, keystrokes are no longer passed onto Macro Express and the Macro Express Hot Keys and Shortkeys suddenly fail to trigger macros.

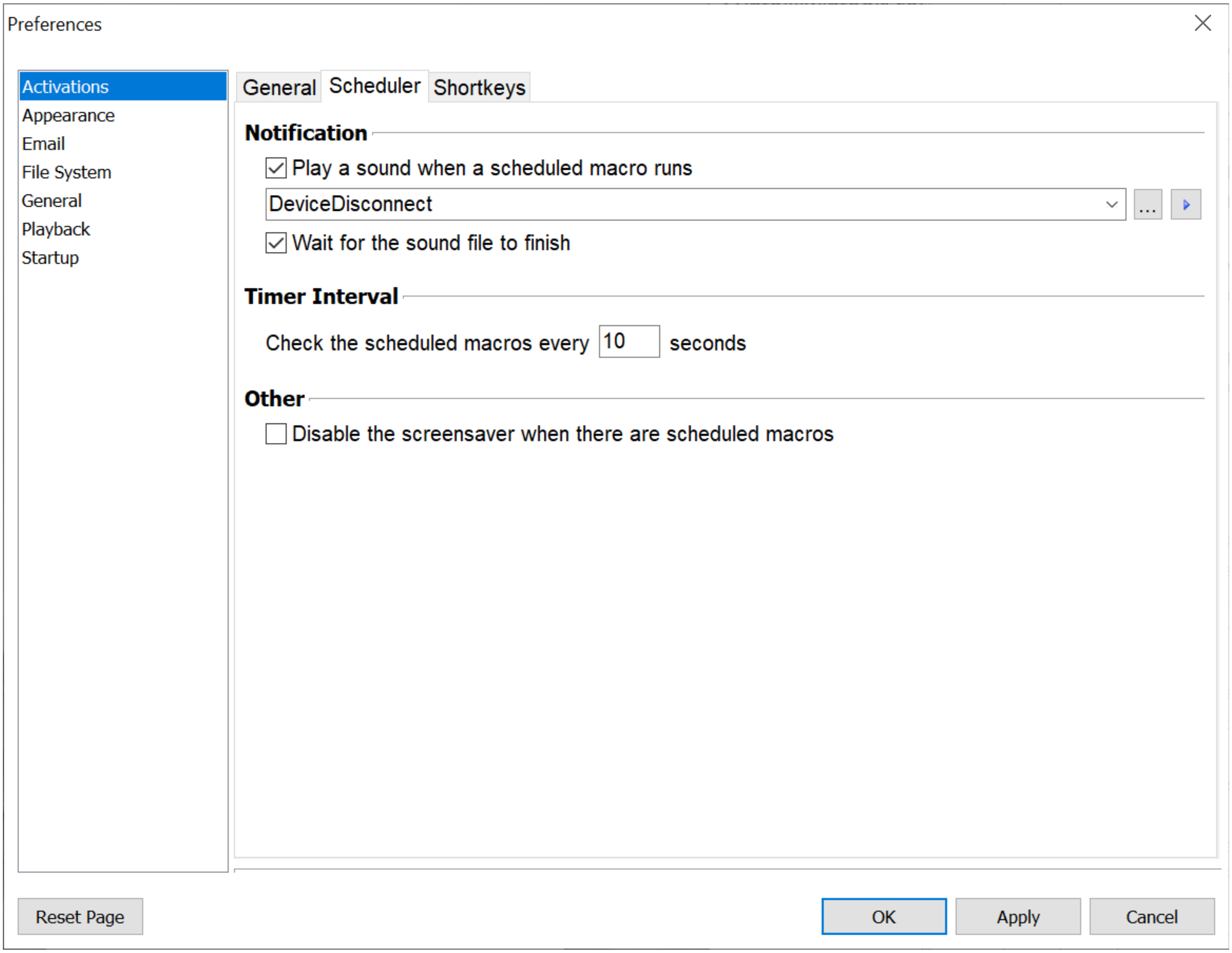
This feature should automatically give Macro Express first looks at the keystrokes thus preserving the use of Macro Express Hot Keys and Shortkeys. In addition, because Macro Express properly passes on keystrokes through the chain of keyboard hooks, other programs will still function correctly.

The above also applies to the Windows mouse hooks.

Use the clipboard hook

This option turns on or off the ability to run clipboard activated macros. Select this option to allow clipboard activated macros to run. If conflicts arise with other programs using the clipboard, turn off this option.

Scheduler



Notification

☑ Play a sound when a scheduled macro runs

DeviceDisconnect

☑ Wait for the sound file to finish

Timer Interval

Check the scheduled macros every 10 seconds

Other

☐ Disable the screensaver when there are scheduled macros

Macro Express Explorer > Options > Preferences > Activation > Scheduler Tab

Notification

Select this option to play a sound (.wav) file when a scheduled macro is about to execute. The sound file will not play back when using the When the program starts and When the macro file is loaded schedule options. The default option is to not play any sound.

Sound File

Enter the file name and path of the sound file into the edit field. Or click on the Browse button to locate the sound file. You may also select one of the system sounds available on your computer by clicking on the drop down arrow. Click on the Test button to hear the sound selected.

A system sound is a sound assigned to specific Program Events within Windows such as Asterisk, Device Connect or Windows Logoff. The sounds assigned to these events are selected in the Windows Control Panel either individually or as part of a theme. Not all system sounds listed in the Control Panel are available in this command.

Wait for the sound file to finish

Placing a check in the box causes the playback of the macro to stop until the sound file has finished playing. The macro resumes playback once the sound file has finished.

Check the scheduled macros every ___ seconds

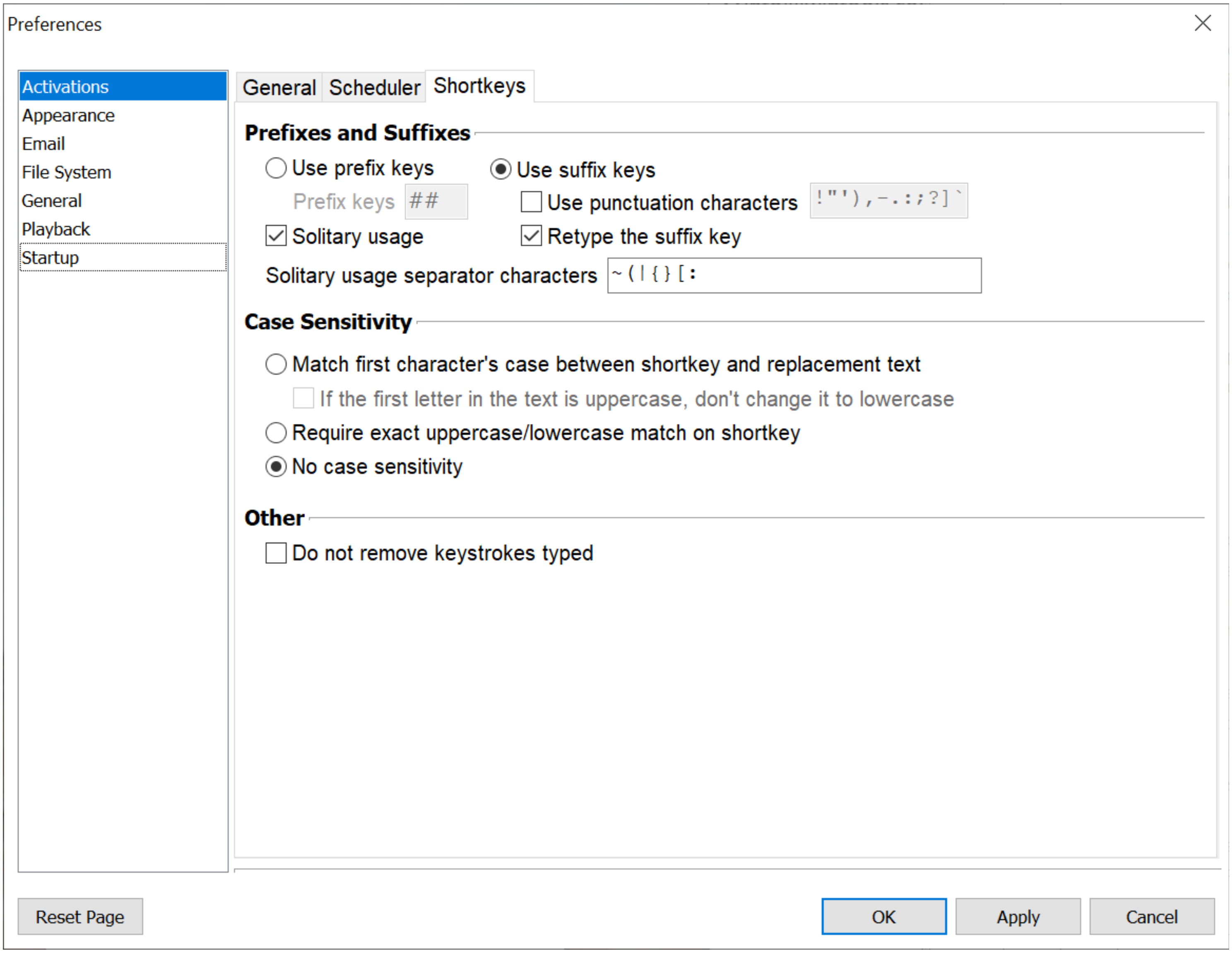
Specify the interval, in seconds, for how often Macro Express checks to see if a scheduled macro is ready to execute. Macro Express will shut off the timer if no scheduled macros exist in the current file or none are active. The default setting is to check for scheduled macros every 10 seconds.

Suppose a scheduled macro plays back every 3 seconds. For the macro to work correctly, the Timer Interval needs to be set to check at least every 3 seconds. Otherwise some of the scheduled times will not run.

Disable the screensaver when there are scheduled macros

Screen Savers prevent a scheduled macro from running. This option prevents a screen saver from starting if Macro Express is running and if there are any active scheduled macros in the macro file currently in use. It stops the screen saver from starting, allowing the macro to run.

Shortkeys



Prefixes and Suffixes

○ Use prefix keys ● Use suffix keys

Prefix keys ## □ Use punctuation characters !"'),-.:;?]`

☑ Solitary usage ☑ Retype the suffix key

Solitary usage separator characters ~(|{}[:

Case Sensitivity

○ Match first character's case between shortkey and replacement text

□ If the first letter in the text is uppercase, don't change it to lowercase

○ Require exact uppercase/lowercase match on shortkey

● No case sensitivity

Other

□ Do not remove keystrokes typed

Macro Express Explorer > Options > Preferences > Activation > Shortkeys Tab

A shortkey is a set of characters such as abc, myname, etc. The macro runs when the characters are typed into an application. Click on Options > Preferences > Activations and the Shortkeys tab to define the settings. Below are descriptions of the default settings that can be assigned to all shortkeys.

Prefixes and Suffixes

The Prefix Key and Suffix Key options cannot be overridden elsewhere in the program.

Use Prefix Keys

The Prefix Keys are used to designate up to three characters that may be added to the beginning of every shortkey. A prefix key is a method to prevent the accidental triggering of a macro during everyday typing.

This option places a prefix key or keys, such as z, ##, qq, etc., in front of each shortkey. So, if the Prefix Key was set to ## and the shortkey was set to "test", typing ##test starts the macro. Typing ## before any text would make it very unlikely for a macro to start accidentally.

The other option is to leave the Prefix Keys field blank. In the above example, typing the shortkey "test" would start the macro. No other keys are required.

Below is a list of the Prefix characters that are recognized by Macro Express.

- The letters A through Z (both upper and lower case)
- The numbers 0 through 9
- The following characters: ! @ # $ % ^ & _ = < > ? . ; , * - / + ` ' ")] and space bar

Shortkey Detection

The program provides two methods of "shortkey" detection, Solitary Usage and non Solitary Usage. Solitary usage is the default setting.

Solitary Usage

The solitary usage option indicates that a shortkey is recognized only when the shortkey is preceded by a character or key that is not valid as part of the shortkey name. This is best illustrated by the example below.

Assume the shortkey was named red and the macro text was a bright color. The replacement would not take place if the word Fred were typed. Why? Because "F" is a valid character in a short key name. We really didn't want the replacement to take place because we were typing Fred and not red.

However, the replacement would take place if typing the word red preceded by a blank space. Why? Because a blank space is not valid as part of a shortkey.

Non Solitary Usage

The non solitary usage option (leaving the Solitary Usage option unchecked) indicates that a shortkey will be recognized whenever the short key text is typed, even if it is part of another word. The following example illustrates this point.

Assume the shortkey was named red and the macro text was a bright color. Typing the word Fred would cause the macro to start and the text typed would be Fa bright color.

Typically the "non solitary usage" mode is used with a single prefix key or two to ensure that a macro does not play back unintentionally.

Solitary Usage Separator Characters

One or two Prefix Keys may optionally be input that are applied to all original shortkeys created. The following Prefix characters may be entered and used:

- The letters A through Z (both upper and lower case)
- The numbers 0 through 9
- Others: Essentially any other valid character or symbol such as ! @ # $ % ^ & _ = < > ? . ; , * - / + \.

This list of Separator Characters provides a method to remove characters for use as a prefix key. Attempting to assign an invalid character as a prefix would not be allowed.

Use Suffix Keys

Another alternative is to use the Suffix Keys option. With this option activated, type the shortkey and then press down either the Space Bar or the Enter Key to activate the macro.

Use Punctuation Characters

Punctuation keys such as . , : ; ? !) '] " ` and - may also be used as a suffix key for a shortkey when the Include Punctuation Characters box is selected. If the shortkey was set to test, type the word test while in another Windows application and press either the Space Bar, Enter Key or one of the punctuation keys listed above to start the macro. Punctuation characters may be added or removed from this list.

Retype Suffix Key

The Retype Suffix Key option, when checked, types the suffix key (Enter, Space or punctuation keys) to the end of the macro text. This option is mainly for those who use shortkeys as a simple text replacement in writing documents.

Assume a shortkey of test with a replacement text of This is a test. Typing the word test into notepad or other text editor and using the space bar as the trigger causes the macro to start. The replacement text is typed into the application and the cursor moves one space to the right of the text.

Or typing test and using the ? to activate the shortkey causes the replacement text to appear with a ? at the end of the text. If the Replay Suffix key is not selected, the suffix key is not placed at the end of the replacement text.

Complete list of characters that can be used with shortkeys

Up to 32 characters may be entered as an original short key (which are to be replaced). In addition, up to 3 Prefix Keys may optionally be entered in that are applied to all Original Shortkeys created. The following Prefix characters may be entered and used:

The letters A through Z (both upper and lower case)
The numbers 0 through 9
The following characters: ! @ # $ % ^ & _ = < > ?
. ; , * - / + ` ' ")] and space bar

If not using a prefix key, use the suffix key and press the space bar, the enter key or a suffix character of . , : ; ? ! or - after typing the shortkey to activate the macro.

✓ **Note:** Pressing down the CTRL or ALT key is detected by the program when using the Shortkeys option and is interpreted as a key being pressed. Therefore, if the ALT or CTRL key is pressed in the middle of a shortkey being typed, the shortkey will not be recognized and the macro will not start.

Case Sensitivity

The following options are the default options used for each new shortkey created. These options can be overridden on an individual shortkey basis.

Match First Character's Case Between Shortkey & Replacement Text

This option causes the upper case setting of the first letter in the shortkey to dictate that the first character in the macro will be upper case (if the first thing in the macro is a character). This option is mainly for those doing simple text replacement in documents and facilitates an easy way to capitalize the replacement text in instances where it needs to be, such as at the beginning of a sentence.

Require Exact Upper Case/Lower Case Match on Shortkey

With this setting selected, the shortkey must be typed exactly as it appears (upper and lower case letters) in order for the replacement to work.

No Case Sensitivity

Neither of the above cases apply with this option checked.

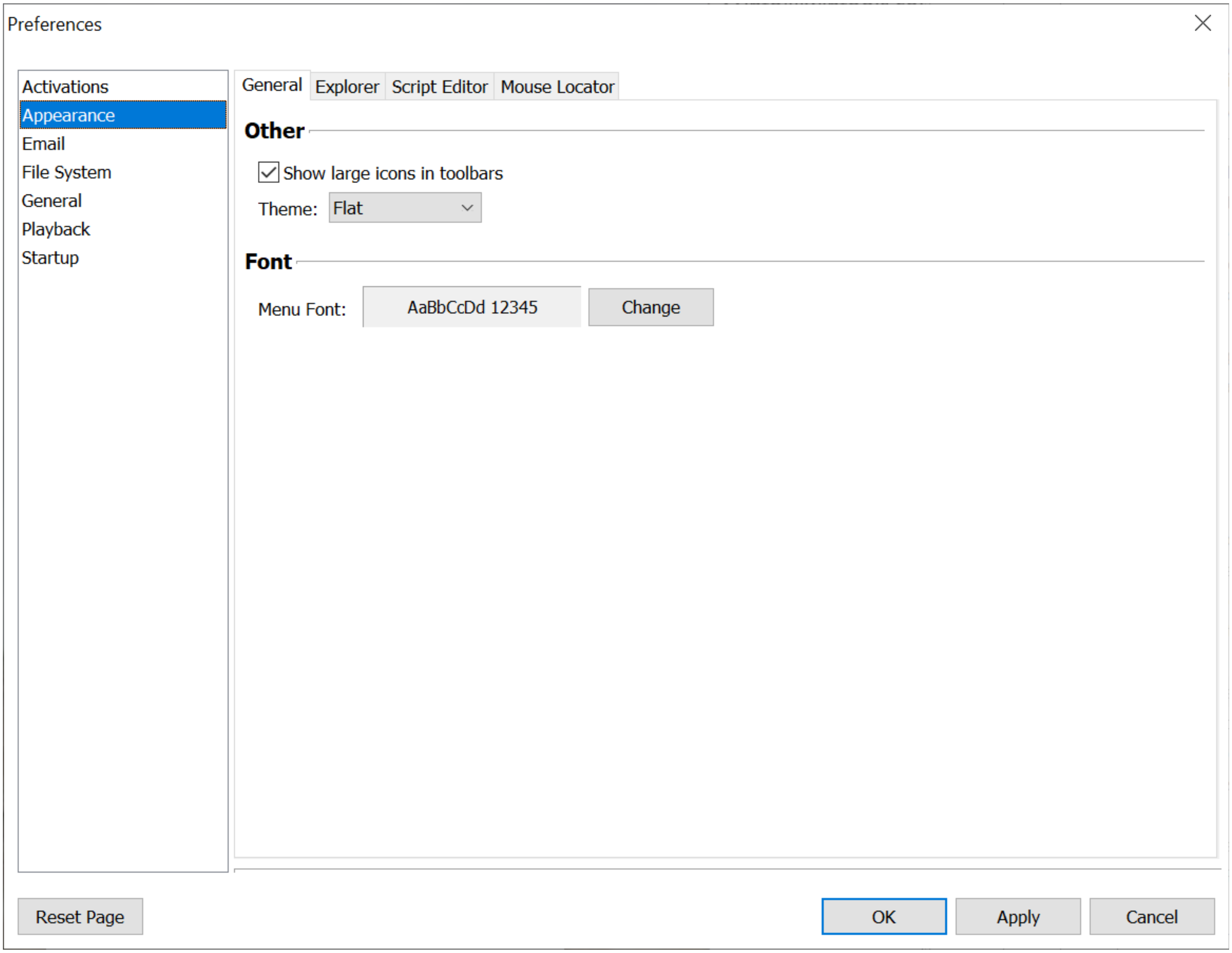
Other

Do Not Remove Keystrokes Typed

This option instructs the shortkey to not erase itself after being typed. Normally, backspaces are issued to remove the shortkey from the screen before the macro is executed.

When using shortkeys in non-text usage programs, it may not be desirable to have "backspaces" issued. For example, using a shortkey in Internet Explorer where focus is not on an edit box, will cause the "back" button to be pressed numerous times due to the backspaces being issued. The Do Not Remove Keystrokes Typed option will prevent this from happening.

General Appearance



Preferences

Other

☑ Show large icons in toolbars

Theme: Flat

Font

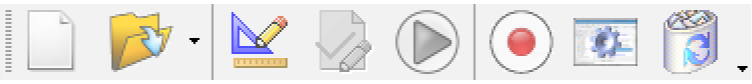
Menu Font: AaBbCcDd 12345

Macro Express Explorer > Options > Preferences > Appearance > General Tab

Other

Show large icons in toolbars

Use this option to display the larger icons in the Macro Express - Explorer toolbar. If this option is turned off, the smaller icons will be displayed. The Macro Express - Explorer must be restarted before the change takes effect.



Theme

Choose a theme from the drop down list to change the look of the Explorer window. Choose from Blue Gradient, Retro, Flat, Standard or Custom options.

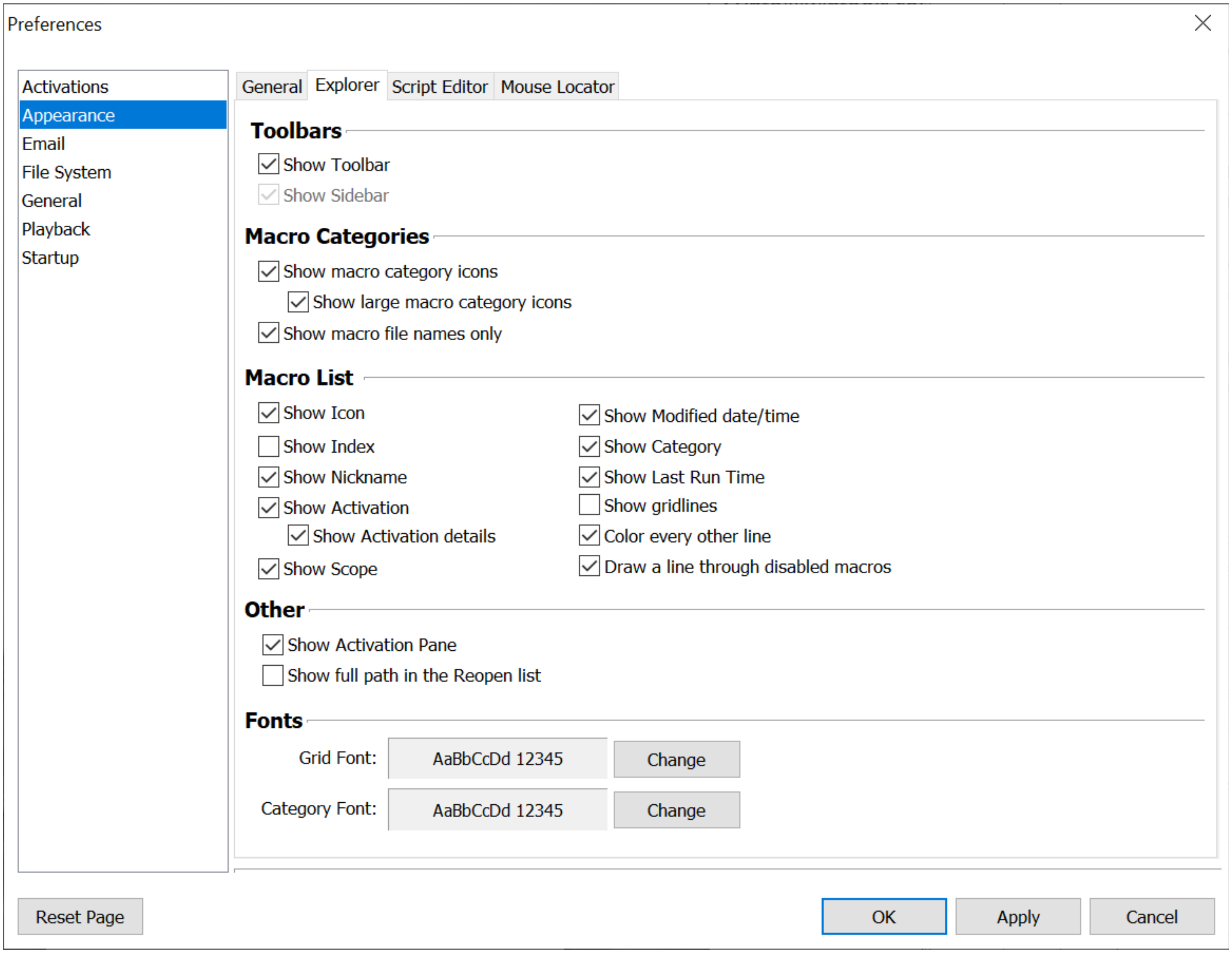
The **Custom theme** allows the user to select colors for various window components, choose from several menu styles and choose the button alignment and spacing in the program's windows and dialogs.

Font

Menu Font

Change the font type and style of the menus in the Macro Express Explorer, Script Editor and Menu Builder windows.

Explorer



Preferences

Toolbars

☑ Show Toolbar

☑ Show Sidebar

Macro Categories

☑ Show macro category icons

☑ Show large macro category icons

☑ Show macro file names only

Macro List

☑ Show Icon

□ Show Index

☑ Show Nickname

☑ Show Activation

☑ Show Activation details

☑ Show Scope

☑ Show Modified date/time

☑ Show Category

☑ Show Last Run Time

□ Show gridlines

☑ Color every other line

☑ Draw a line through disabled macros

Other

☑ Show Activation Pane

□ Show full path in the Reopen list

Fonts

Grid Font: AaBbCcDd 12345

Category Font: AaBbCcDd 12345

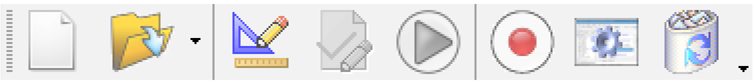
Macro Express Explorer > Options > Preferences > Appearance > Explorer Tab

Use these options to adjust the look of the Macro Express Pro - Explorer window.

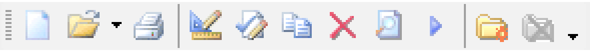
Toolbars

Show Toolbar

Select to display the toolbar (similar to what is shown below). The toolbar is not displayed in the Macro Explorer window when the option is not selected.

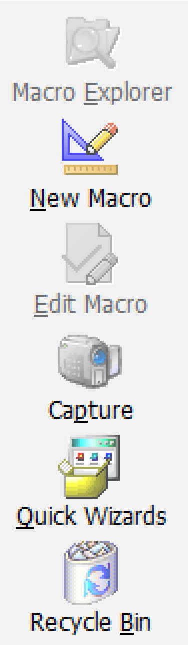


Or if the Show large icons in toolbars option is turned off, the smaller icons will be displayed.



Sidebar

Choose to display the sidebar (similar to what is shown below). The sidebar is not displayed in the Macro Explorer window when the option is not selected. If the large icon option has been selected, then the Sidebar option is not available.



Show macro category icons

This option displays folder icons next to each category in the Category field of the Macro Explorer.

Show large macro category icons

Choose to show a larger size folder icon in the Category field.

Show macro file names only

Choose to display just the name of the macro file, rather than showing All Categories with the macro file name in parentheses.

Macro List

Below is a chart containing the options to display in the Macro Express Explorer window.

Display Options	Description
Show Icon	Displays icons next to each macro listed in the Macro Express has not been assigned to the macro, the default Macro Expres
Show Index	Each macro is assigned an index number. Display the index nu
Show Nickname	Display the nickname assigned to each macro created.
Show Activation	Show the activation (hot key, shortkey, etc.) that is assigned
Show Activation Details	Show additional details about the activation displayed, such as in use. If a macro has more than one activation assigned, eacl rather than showing "Multiple" in the Activations column.
Show Scope	Display the scope that is assigned to each macro.
Show Modified date/time	Display the date and time that the macros were last modified
Show Category	Show the name of the category (if any) that the macro is assi
Show Last Run Time	Show the date and time that the macro was last run.
Show gridlines	Show grid lines between each macro listed in the Macro Expre
Show preview pane	Display a panel at the bottom of the Macro Explorer window p details about the highlighted macro.
Color every other line	Show every other macro in the Macro Express - Explorer list w background. This helps to read across the line easier.
Draw a line through disabled macros	This option draws a line through any disabled macros displaye - Explorer window to make it more readily apparent that the r All disabled macros are displayed with italicized red font.

Other

Show full path in the Reopen list

In the Macro Express Pro - Explorer window click on File > Reopen to view a list of the most recently used macro (.mex) files. By default the list only displays the file name. Select this option to show the file name and the path where the file is saved on the computer.

Fonts

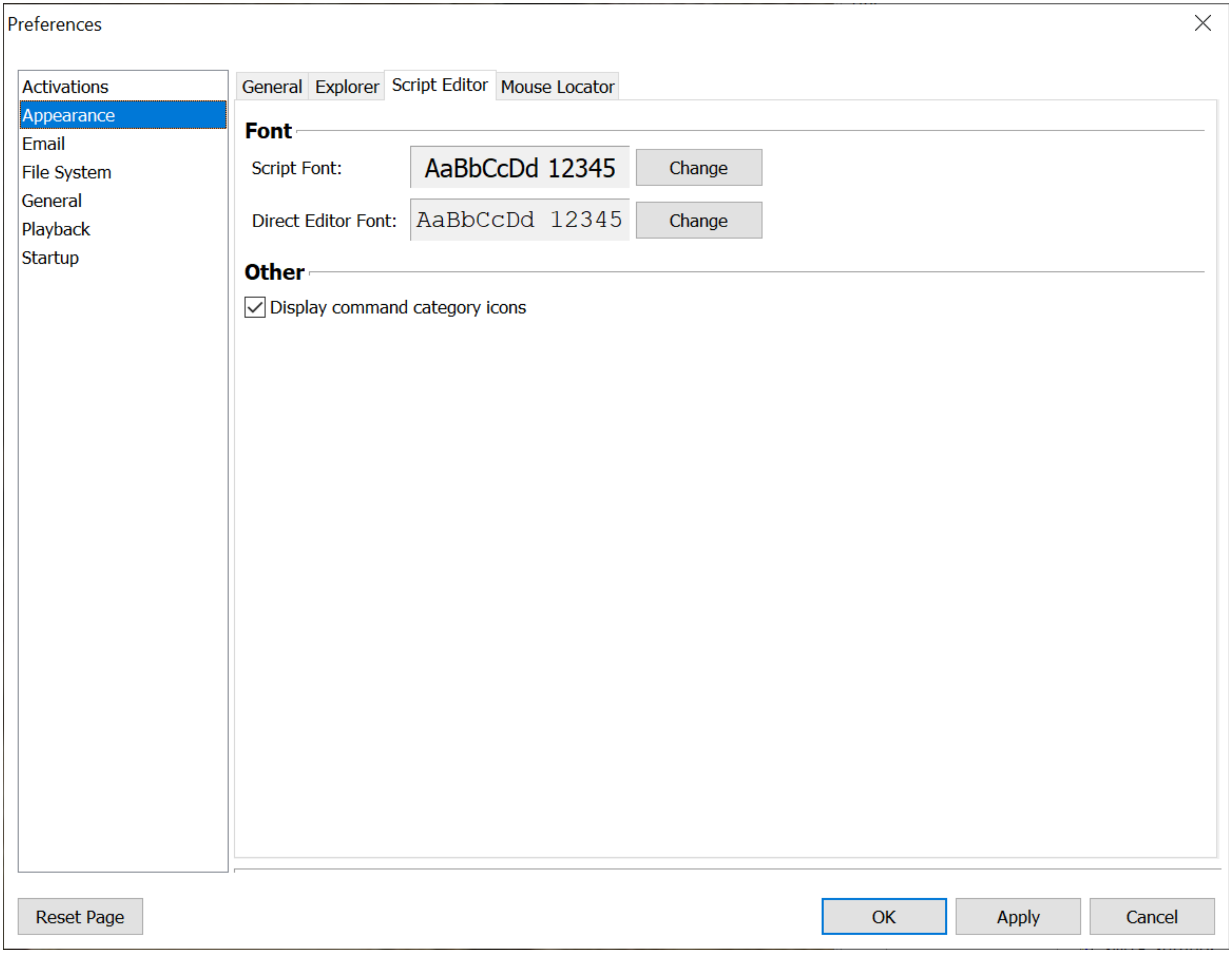
Grid Font

Change the font type, style, size and color of the text displayed in the Macro Express Explorer grid.

Category Font

Change the font type, style, size and color of the text displayed in the Macro Express Explorer categories column.

Script Editor



Macro Express Explorer > Options > Preferences > Appearance > Explorer Tab

Script Font

Use this option to select the font type, color, size, etc. to display in the Script Editor.

Direct Editor Font

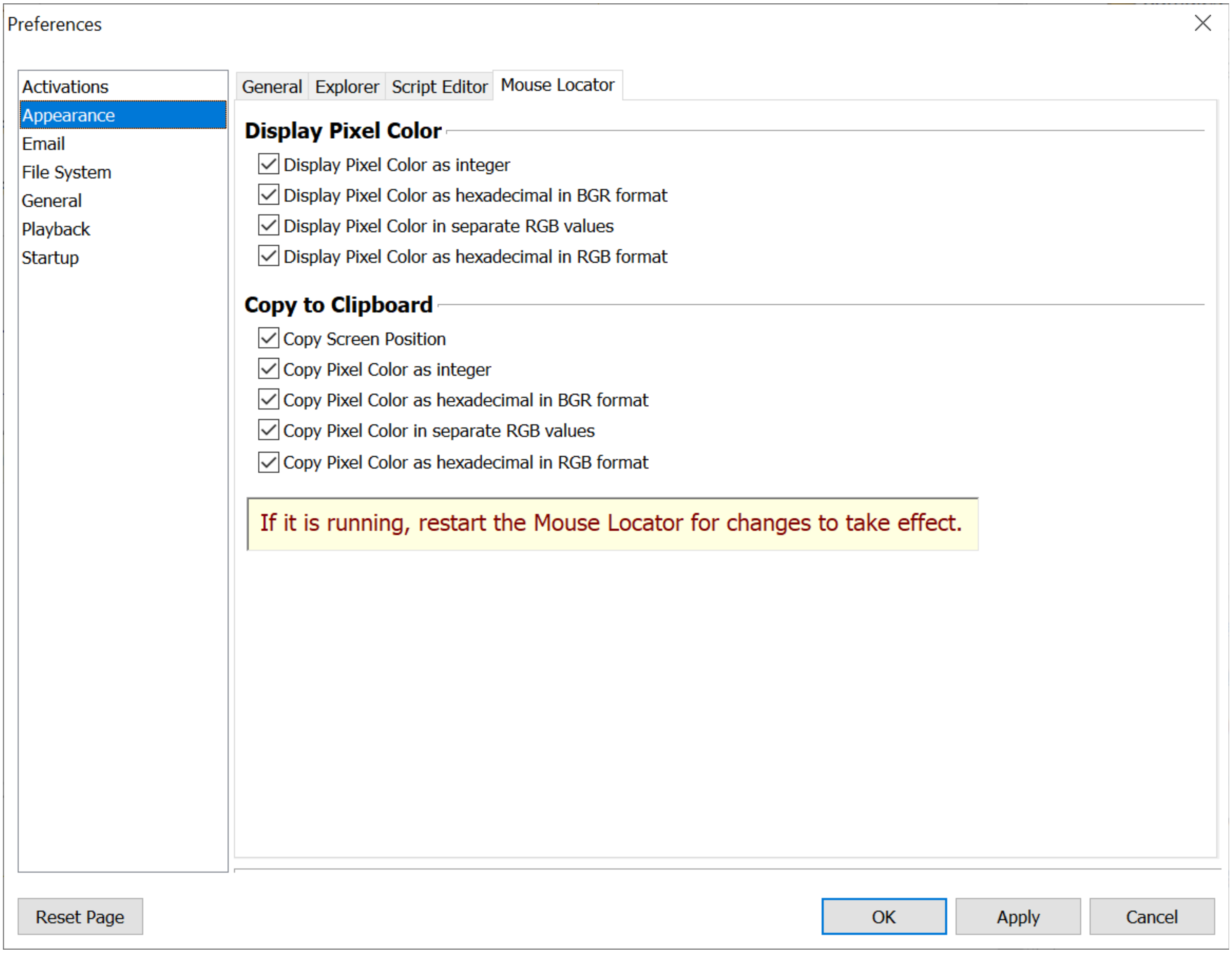
Use this option to select the font type, color, size, etc. to display in the Direct Editor.

Display command category icons

Each group of macro commands in the Script Editor, such as CD-ROM, Logic, etc., has a distinct icon associated with it. Each time a command from that group is used and added to the macro script, the associated icon is also displayed in the script.

If not selected, the icons are not displayed in the category list or next to each command in the macro script.

Mouse Locator Settings



Preferences

Display Pixel Color

☑ Display Pixel Color as integer

☑ Display Pixel Color as hexadecimal in BGR format

☑ Display Pixel Color in separate RGB values

☑ Display Pixel Color as hexadecimal in RGB format

Copy to Clipboard

☑ Copy Screen Position

☑ Copy Pixel Color as integer

☑ Copy Pixel Color as hexadecimal in BGR format

☑ Copy Pixel Color in separate RGB values

☑ Copy Pixel Color as hexadecimal in RGB format

If it is running, restart the Mouse Locator for changes to take effect.

Macro Express Explorer > Options > Preferences > Appearance > Mouse Locator tab

Style

The Translucent style is the default for the Mouse Locator. However, depending on what is displayed underneath the Mouse Locator it may be difficult to read the values for the mouse position and colors. Turn off the Translucent style if you experience difficulties reading the text.

- ✓ The Translucent Style is only available in Windows 7 and prior.

Display Pixel Color

To display colors, computers use the RGB Additive Color System. The letters R, G and B represent Red, Green and Blue respectively. Separate values are used for the amount of Red, Green and Blue. On a Windows computer, each color ranges from 0, all the way off, up to 255 (or FF hexadecimal) all the way on. Combining different values for each basic color allows millions of different colors to be displayed. For example, White has a value of 000000, Green has a value of 00FF00 and black has a value of FFFFFF.

The hexadecimal number that represents a color can also be represented as an integer value. For example, one orange color has a hexadecimal value of 2D42D4. When displayed as an integer this value is 2966228. The Get Pixel command returns the integer value of the color for a given pixel. This allows the integer variable commands to test for specific colors in your macros.

The Mouse Locator displays colors in a variety of formats to make it convenient to enter colors into other programs. For example, the Get Pixel Color uses an integer color value. When entering a color for a web page the color value is in hexadecimal RGB format. To enter a color in Word requires separate integer values for Red, Green and Blue. And some programming languages require a hexadecimal BGR (Blue, Green, Red) format.

The Display Pixel Color options allow you to choose which Pixel Color values to display on the Mouse Locator.

Copy to Clipboard

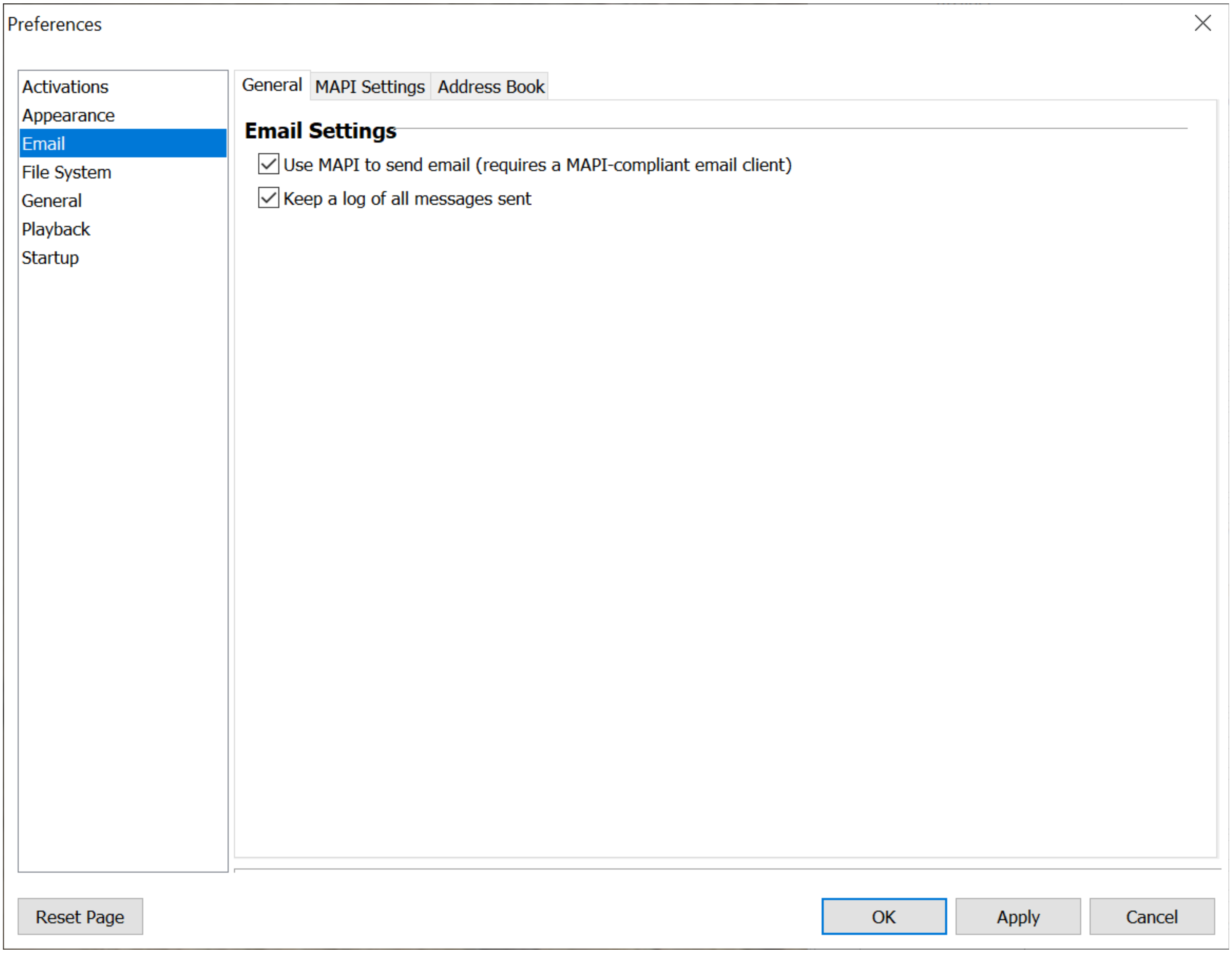
You can use the Mouse Locator to get the value of a color on your computer monitor. Make sure the Mouse Locator is the active window, then move your mouse over the color you want and press Ctrl + c. The color will be copied to the clipboard so you can paste it into another program.

By default, several values are copied to the clipboard including the Screen Position of the mouse and the Pixel Color in a variety of formats. Here is an example:

```
990,385,2966228,2D42D4,212,66,45,#D4422D
```

The Copy to Clipboard options allow you to choose which values to include when copying to the clipboard.

General Email Settings



Email Settings

☑ Use MAPI to send email (requires a MAPI-compliant email client)

☑ Keep a log of all messages sent

Macro Express Explorer > Options > Preferences > Email > General Tab

Use MAPI to send Email

Select this option to send messages through the default email program rather than using the Macro Express email client. Macro Express will only be able to send email using the default email client if it is compatible with the Messaging Application Programming Interface (MAPI). Mainstream email programs such as Outlook, Eudora, Thunderbird, etc. are compatible.

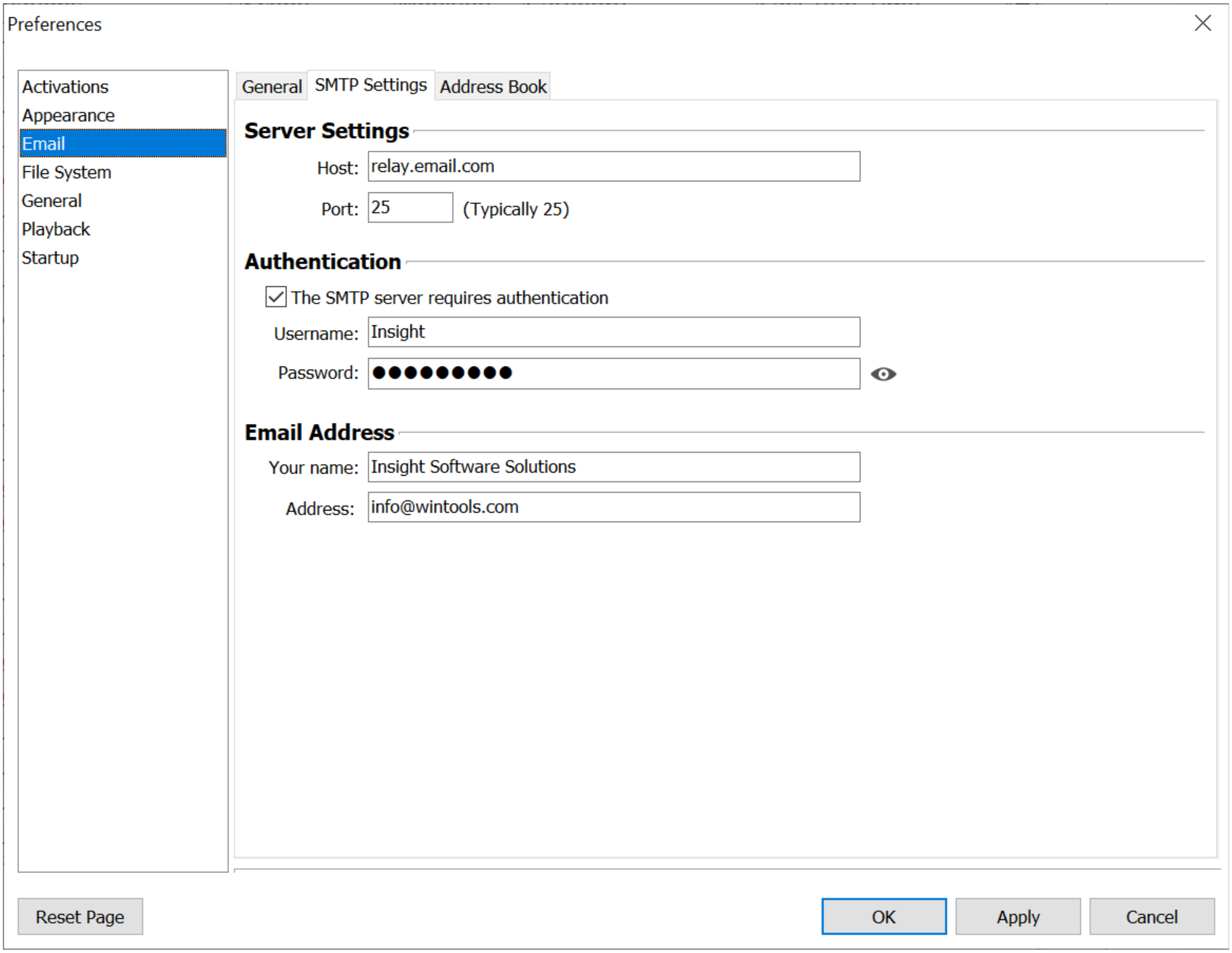
If using this option, the SMTP Settings tab changes to a MAPI Settings tab so the options can be set.

Keep a Log of All Messages Sent

This option logs all email messages sent through Macro Express to an email log file. From the Macro Explorer window click on Tools > Sent Email Log to review the mails sent via Macro Express.

SMTP Settings

To successfully use the email commands via SMTP, the default email settings need to be entered into the SMTP settings panel. Click on Options > Preferences > Email to enable the email functions of Macro Express.



Server Settings

Host: relay.email.com

Port: 25 (Typically 25)

Authentication

☑ The SMTP server requires authentication

Username: Insight

Password: ●●●●●●●●●

Email Address

Your name: Insight Software Solutions

Address: info@wintools.com

Macro Express Explorer > Options > Preferences > Email > SMTP Settings

Server Settings

The SMTP Server is what allows email to be sent over the Internet. The server settings may be obtained from the Internet Email provider or from the configuration section of the default email program on the computer.

Host - The "Host" value may appear as a series of numbers and periods such as 192.168.0.1 or as a set of characters looking something like a web site name such as mail.myisp.com.

Port - The port number is almost always 25. When in doubt, set this value to 25.

Authentication

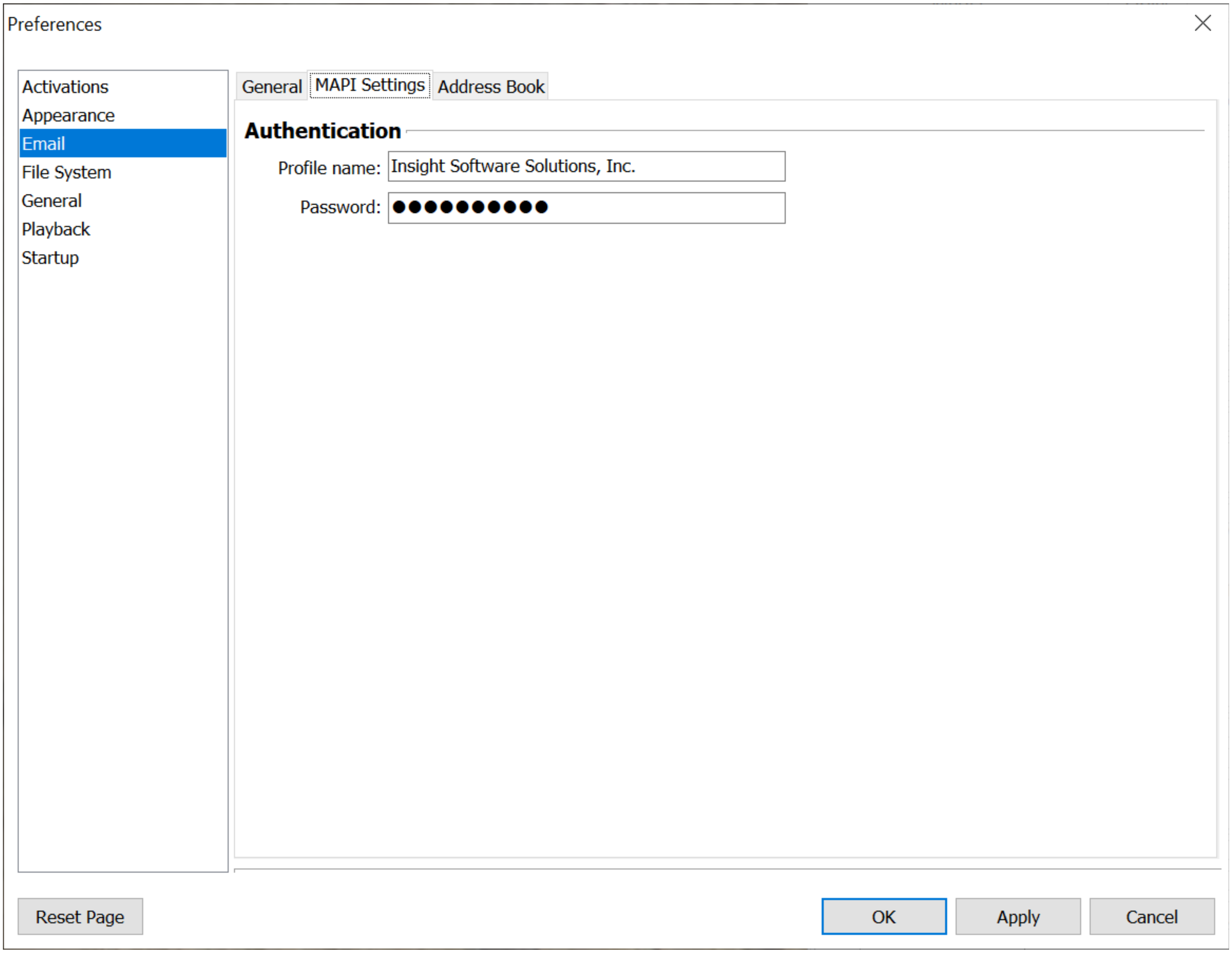
The SMTP Server Requires Authentication

The email provider may require use of SMTP Server Authentication. If so, select this option and insert the user name and password in the appropriate fields. If authentication is not required, then leave this option unchecked.

Email Address

This is the name and email address that display in the return address of any email sent through Macro Express. Please note that if the email address entered is incorrect, the recipients of the email may not know how to reply to the message.

MAPI Settings



Authentication

Profile name: Insight Software Solutions, Inc.

Password: ●●●●●●●●●●

Macro Express Explorer > Options > Preferences > Email > MAPI Settings

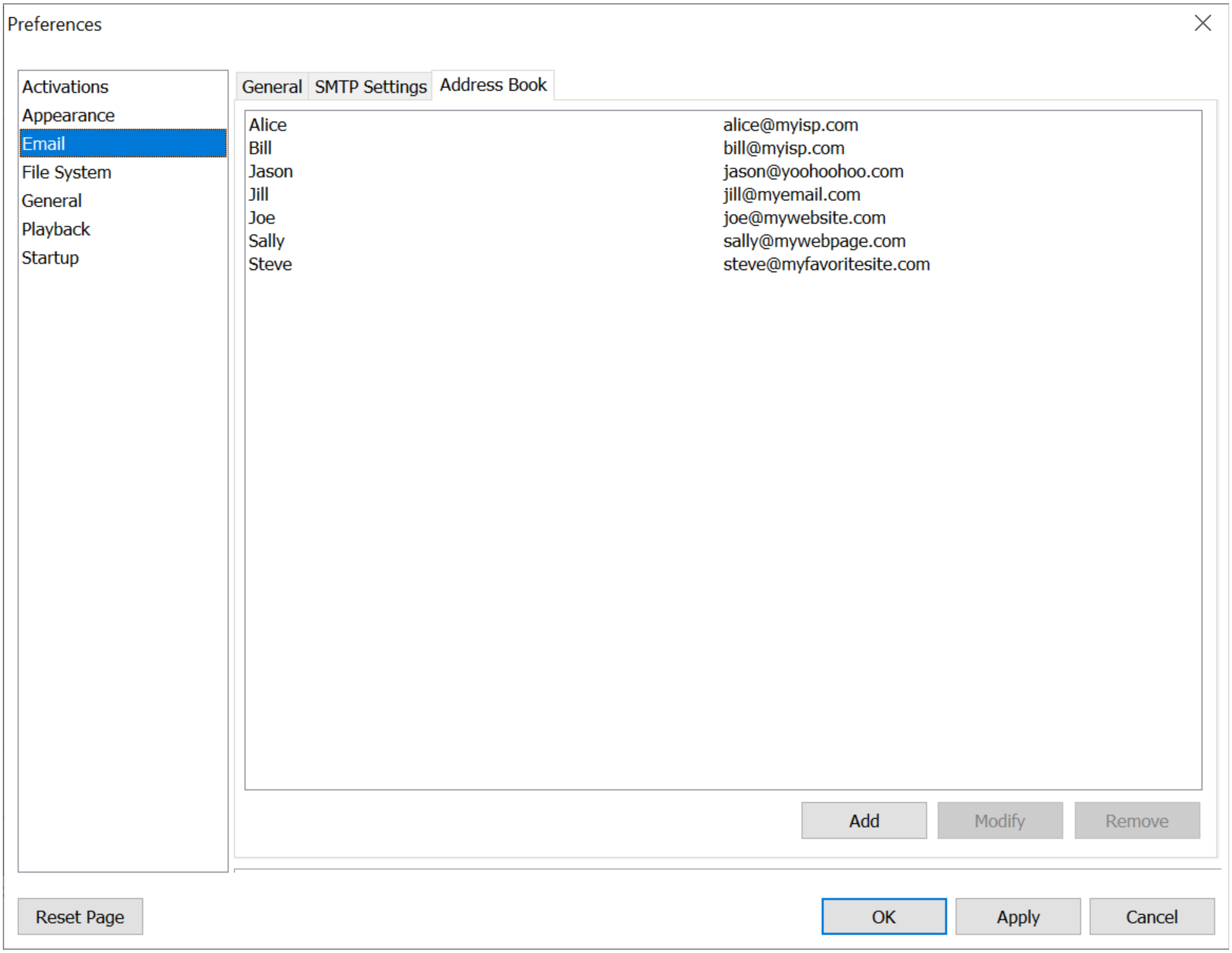
This option is only available if MAPI is selected in the Email > General tab.

The MAPI information - Profile Name and Password - is only required if using multiple email profiles. Enter the email Profile to use with Macro Express and the password.

If not using multiple email profiles then this information should be left blank.

Address Book

The Address Book is a list of names and email addresses to choose from when creating email macros. Use this function to add names, make changes to existing names or delete names from the list. Click on Options > Preferences > Email to access this option. The Email Send command also has an option for editing the list of email recipients.

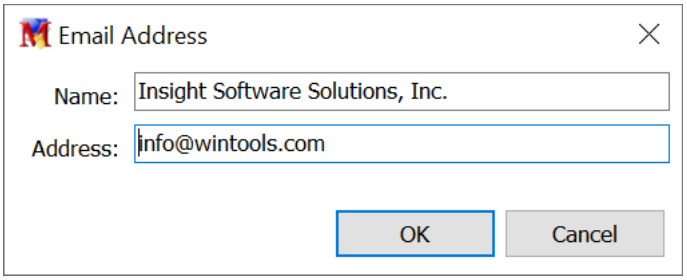


Alice	alice@myisp.com
Bill	bill@myisp.com
Jason	jason@yoohoohoo.com
Jill	jill@myemail.com
Joe	joe@mywebsite.com
Sally	sally@mywebpage.com
Steve	steve@myfavoritesite.com

Macro Express Explorer > Options > Preferences > Email > Address Book Tab

Add

Click on the Add button to open the Email Recipient Maintenance window shown below. The two fields will be blank. Just enter the person's name and email address and click OK to add that person to the list.



Name: Insight Software Solutions, Inc.

Address: info@wintools.com

Address Book Window > Add or Modify Buttons

Modify

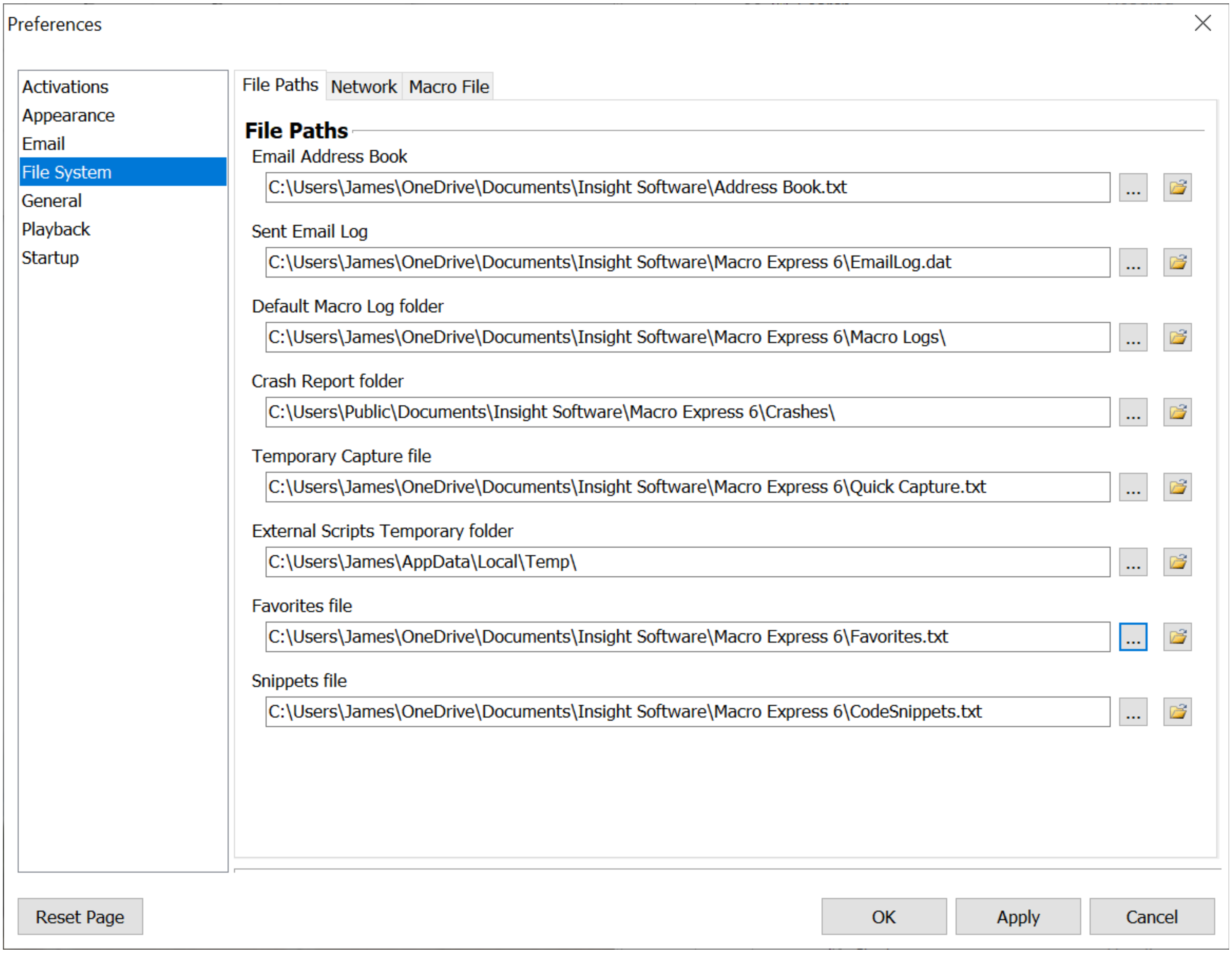
If the email address has changed for someone on the list or other information needs to be changed, highlight that person's name from the list and click on the Modify button. The name and email address are displayed in the Email Address window as shown above. Edit this information and click OK to save the changes.

Remove

To remove a person from the email recipient list, highlight that person's name and click on the Remove button. The name will be removed from the list.

File Paths

Use this option to set the default file locations for a number of files used by Macro Express. Descriptions of the files are provided below.



File Paths

Email Address Book: C:\Users\James\OneDrive\Documents\Insight Software\Address Book.txt

Sent Email Log: C:\Users\James\OneDrive\Documents\Insight Software\Macro Express 6\EmailLog.dat

Default Macro Log folder: C:\Users\James\OneDrive\Documents\Insight Software\Macro Express 6\Macro Logs\

Crash Report folder: C:\Users\Public\Documents\Insight Software\Macro Express 6\Crashes\

Temporary Capture file: C:\Users\James\OneDrive\Documents\Insight Software\Macro Express 6\Quick Capture.txt

External Scripts Temporary folder: C:\Users\James\AppData\Local\Temp\

Favorites file: C:\Users\James\OneDrive\Documents\Insight Software\Macro Express 6\Favorites.txt

Snippets file: C:\Users\James\OneDrive\Documents\Insight Software\Macro Express 6\CodeSnippets.txt

Macro Express Explorer > Options > Preferences > File System > File Paths

Email Address Book

This specifies the path and name of the file that holds any email addresses to be stored. This option is provided so that network users can each have their own individual list of email addresses. If email addresses are common to all users, simply point each user to the same file.

Sent Email Log

The Sent Email Log field displays the path and name of the file that logs the mails sent via Macro Express. The email information will only be saved to this file if the Email Logging option has been set. Start a new log file at any time by changing the file name.

Default Macro Log folder

This specifies the path and name of the file that holds the error logs created when using the Log Errors command. Start a new log file anytime by changing the file name. This option is provided so that network users can each have their own individual list of error messages. If error messages are common to all users, simply point each user to the same file.

Crash Report folder

This specifies the path and folder where crash reports are saved. Crash reports are generated by an unexpected closing of the program. The details of the problem are saved to a file that begins with 'error' and ends with '.TX' in the folder specified here. The default path is the folder where the Macro Express program is installed.

Temporary Capture File

Use the System Macro - Ctrl + Alt + Shift + B to capture a temporary macro. The temporary macro is stored in the file located in the path defined in this field.

External Scripts Temporary folder

Macro Express allows for the creation of External Scripts in JScript, VBScript or HTA/HTML. When the macro plays back the contents of an External Script, the script is copied from the macro and temporarily saved to a file in the folder location selected. The macro reads the contents of the file and plays them back. When the macro is finished the temporary file is deleted.

Favorites File

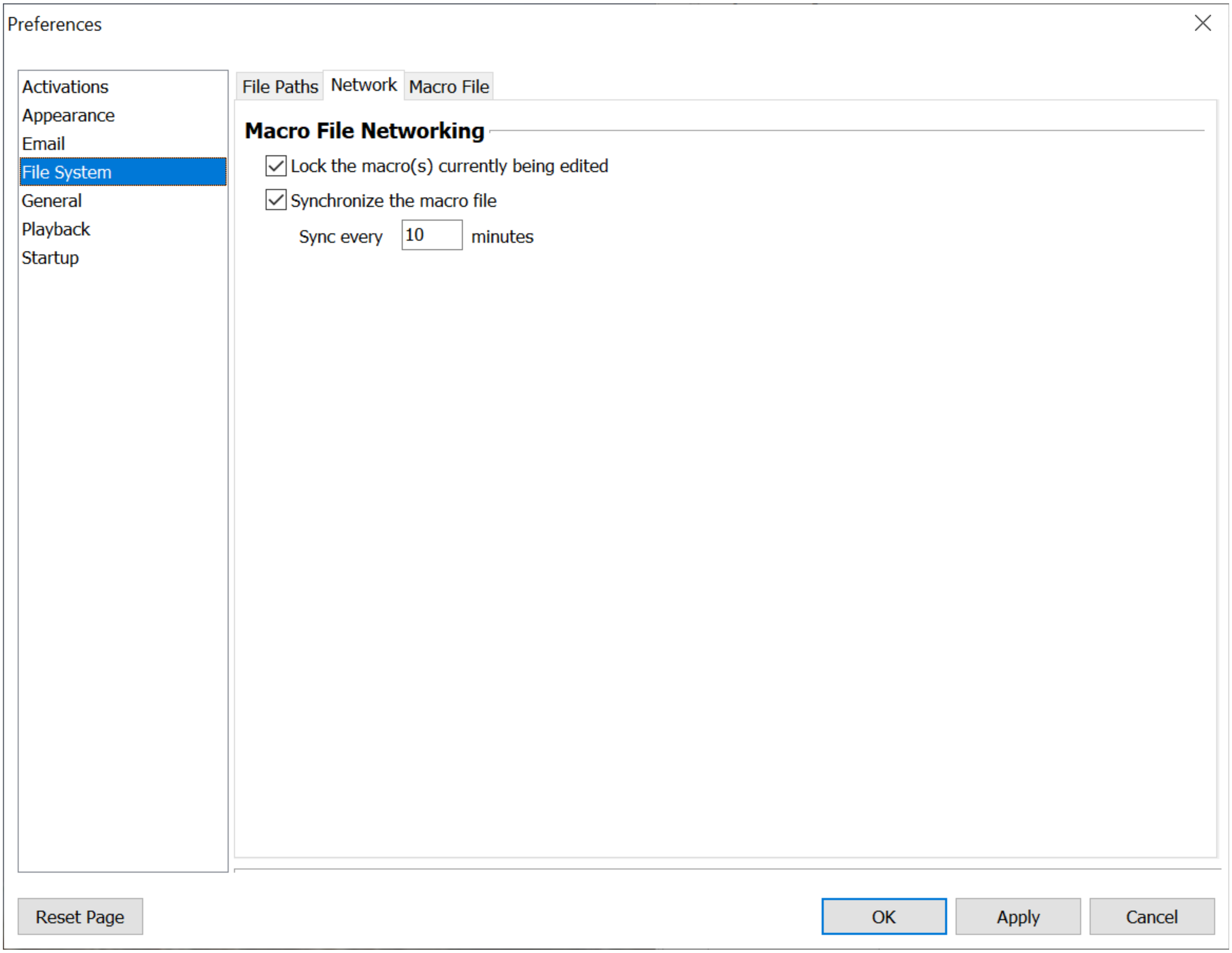
Instead of displaying all of the commands available in the Script Editor, Macro Express includes an option to create a favorites list, making it easier to find the most frequently used commands. From the Script Editor menu click on Options > Customize Favorites to assign the favorites. The information is stored in the file listed in this file path.

Snippets File

Use the Code Snippets function to create snippets of a macro script, such as routines for use in multiple macros. Save the code snippet to later insert into a macro at any time. Select Code Snippets from the drop down list as shown below to view, create or insert snippets. The code snippet information is stored in this file path.

Click on the Open button to open the folder and view the contents associated with the file path selected.

Network



Macro File Networking

☑ Lock the macro(s) currently being edited

☑ Synchronize the macro file

Sync every 10 minutes

Macro Express Explorer > Options > Preferences > File System > Network

Macro File Networking

Macro Express can be set to run over a network using the options described below.

Lock the Macro(s) Currently Being Edited

This option prevents two or more users from editing a macro at the same time by locking the macro as it is being edited. If this option is disabled the lock is neither set nor checked when a macro is edited. If the Lock the Macro(s) Currently Being Edited option is disabled and two or more users edit a macro at the same time, the macro may become corrupted. We recommend leaving this option enabled.

Synchronize the Macro File

This option is available to allow automatic synchronization of data files between users on a network. This updating can be set to occur every minute, every two minutes on up to every 999 minutes.

For example, if the interval is set to 5 minutes and another user makes a change to the same macro file currently open, then within 5 minutes the macro data will automatically be synchronized between the networked computers.

When running Macro Express over a network there are options to prevent certain users from changing any of the Macro Express data. This may be accomplished via the Command Line Parameters.

Sync Every ___ Minutes

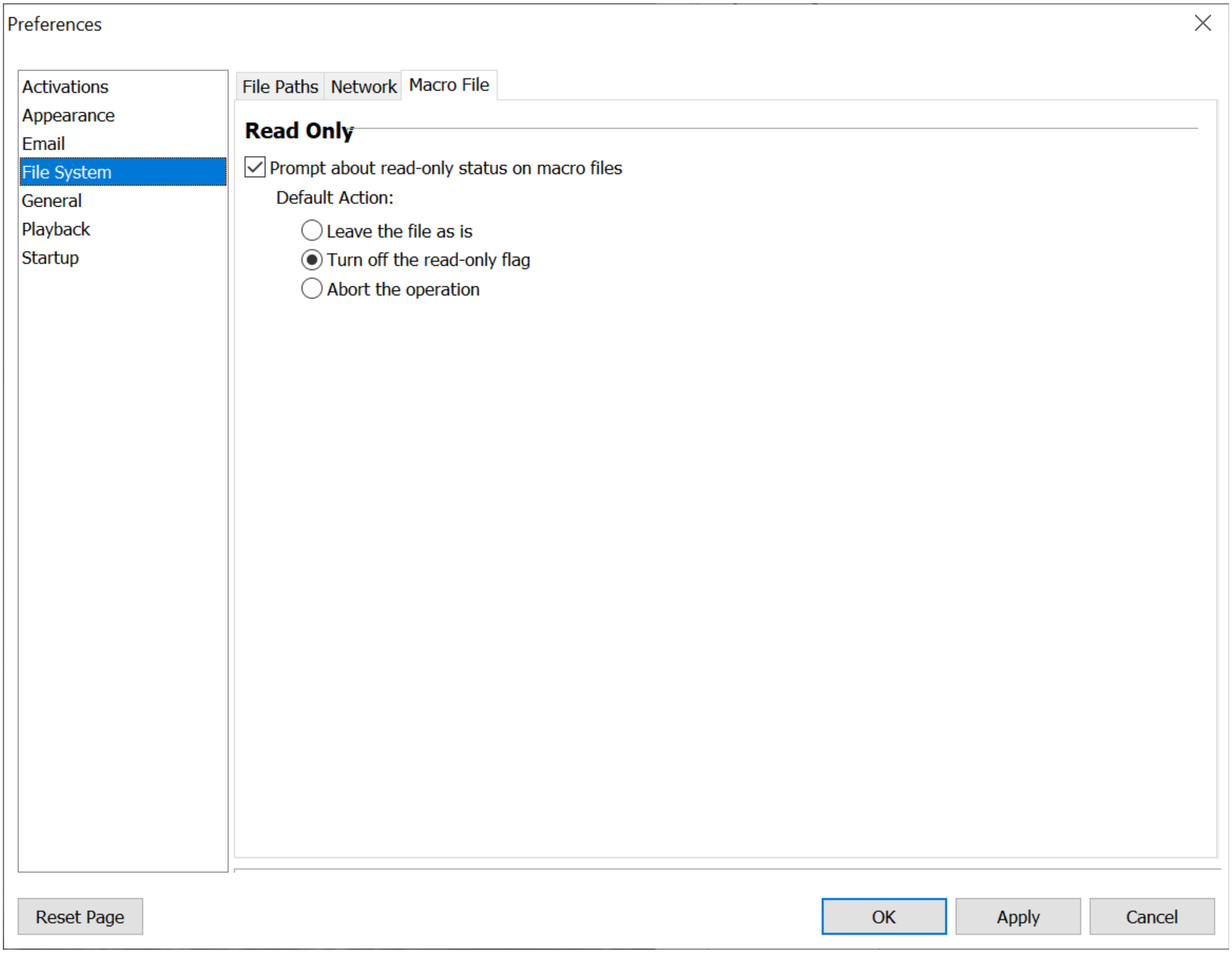
Enter the time interval in minutes of the synchronization schedule.

✓ **Note:** Multi-user networking of this product requires the purchase of a multi-user license.

Read Only Prompt

This preference setting refers to Macro Express (.mex) files that have been set to a Read Only status. It may refer to opening a file through the File > Open File menu, loading a macro file during playback of a macro, or other times the .mex file is being accessed.

- ✓ **Note:** If using the Load New Macro File command and the file to be loaded is a Read Only file, then the Default Action selected in this Preference panel will be used.



Read Only

☑ Prompt about read-only status on macro files

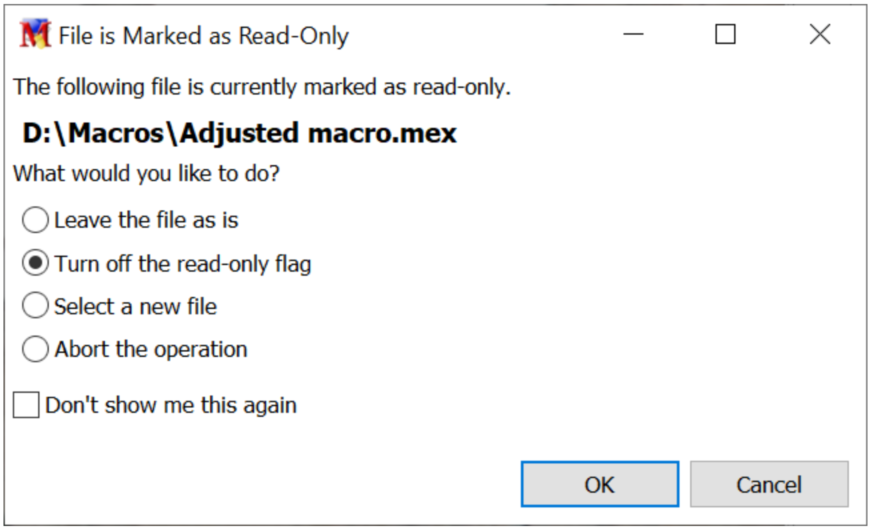
Default Action:

○ Leave the file as is
● Turn off the read-only flag
○ Abort the operation

Macro Express Explorer > Options > Preferences > File System > Read Only Prompt

Prompt about read-only status on macro files

Select this option to have the program prompt for action when a read only .mex macro file is encountered. The following dialog opens allowing the user to select a course of action.



File is Marked as Read-Only

The following file is currently marked as read-only.

D:\Macros\Adjusted macro.mex

What would you like to do?

○ Leave the file as is

● Turn off the read-only flag

○ Select a new file

○ Abort the operation

□ Don't show me this again

Default Action

If the Prompt option is not selected, then choose from one of the three default options discussed below. The action selected will occur each time a read only .mex macro file is encountered.

Leave the file as is

The read only status will be left as is. Macro Express will attempt to perform the requested action with the read only file.

Turn off the read-only flag

The read only status of the macro file is turned off, which will allow for editing of the macro file, etc.

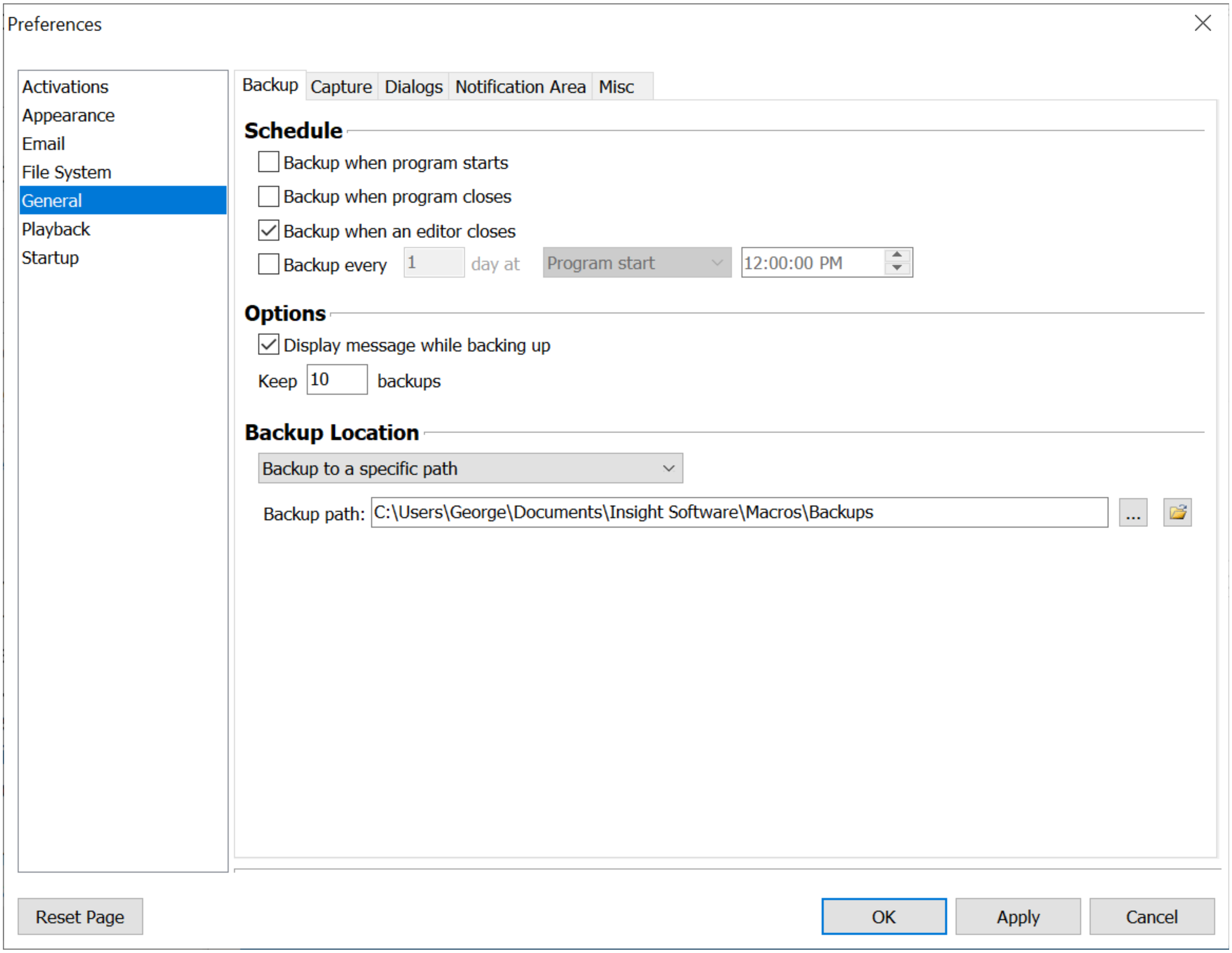
Abort the operation

The function that Macro Express is attempting to do with the file will be aborted.

General

Backup

Use this option to create automatic backups of the macro data file currently in use. This option allows for selecting the frequency of the backups, the number of backups to store and where to save the backup files.



Schedule

□ Backup when program starts

□ Backup when program closes

☑ Backup when an editor closes

□ Backup every 1 day at Program start 12:00:00 PM

Options

☑ Display message while backing up

Keep 10 backups

Backup Location

Backup to a specific path

Backup path: C:\Users\George\Documents\Insight Software\Macros\Backups

Macro Express Explorer > Options > Preferences >General > Backup

- ✓ **Note:** Automatic backups will only occur after the macro (.mex) file has been modified. In the example from the image above, if the file has not been modified, then starting the program or closing the editor will not trigger a backup. Only if the file as been modified, does the automatic backup occur. The date/time stamp of the backup file will match the date/time when the original (.mex) macro file was last modified.

- ✓ Disabling, enabling and deleting macros, as well as creating new or editing existing macros, are considered modifications to the file and will initiate a backup.

Schedule

Determine the frequency of backups.

Backup when program starts

A backup of the data is created when the Macro Express Player is started.

Backup when program closes

This creates a backup of the data every time that the Macro Express Player is closed.

Backup when an editor closes

A data backup file is created each time one of the Macro Express editor windows closes. This includes the Script Editor, the Menu Builder, the Quick Wizard and the Macro Express - Explorer window.

Backup Every ____ Day at _______________

This options provides flexibility for creating a backup schedule. Select the number of days between backups. If the number 2 is entered, then a backup will run every two days. Choose to have the backups run when the program starts, closes or select a specific time of the day.

No Backup

To not schedule any backups, leave the four options unchecked. Manually back up the macro file by opening the Macro Express - Explorer window and clicking on File > Backup. See the Backing Up Macros topic for more information on this.

Options

Choose to display a message notifying of a backup and decide how many backup files to keep.

Display message while backing up

Choose this option to display a message on screen when the backup occurs. If the file is small, this may happen almost instantaneously, so that the display message may not be visible. If the backups are scheduled intermittently, then this may serve as a reminder that a backup is in progress.

Keep __ Backups

Specify the number of backup copies to store on the computer. The default number of backup files is 10.

Backup Location

Specify the location of the backup files from the drop down list. Select the number of backup copies to keep.

Backup to the Same Folder as the Macro File

The backup file is created and saved into the same folder where the original macro file is currently located.

Backup to a Specific Path

Use this option to store the backups in a folder other than the one where the original file is located.

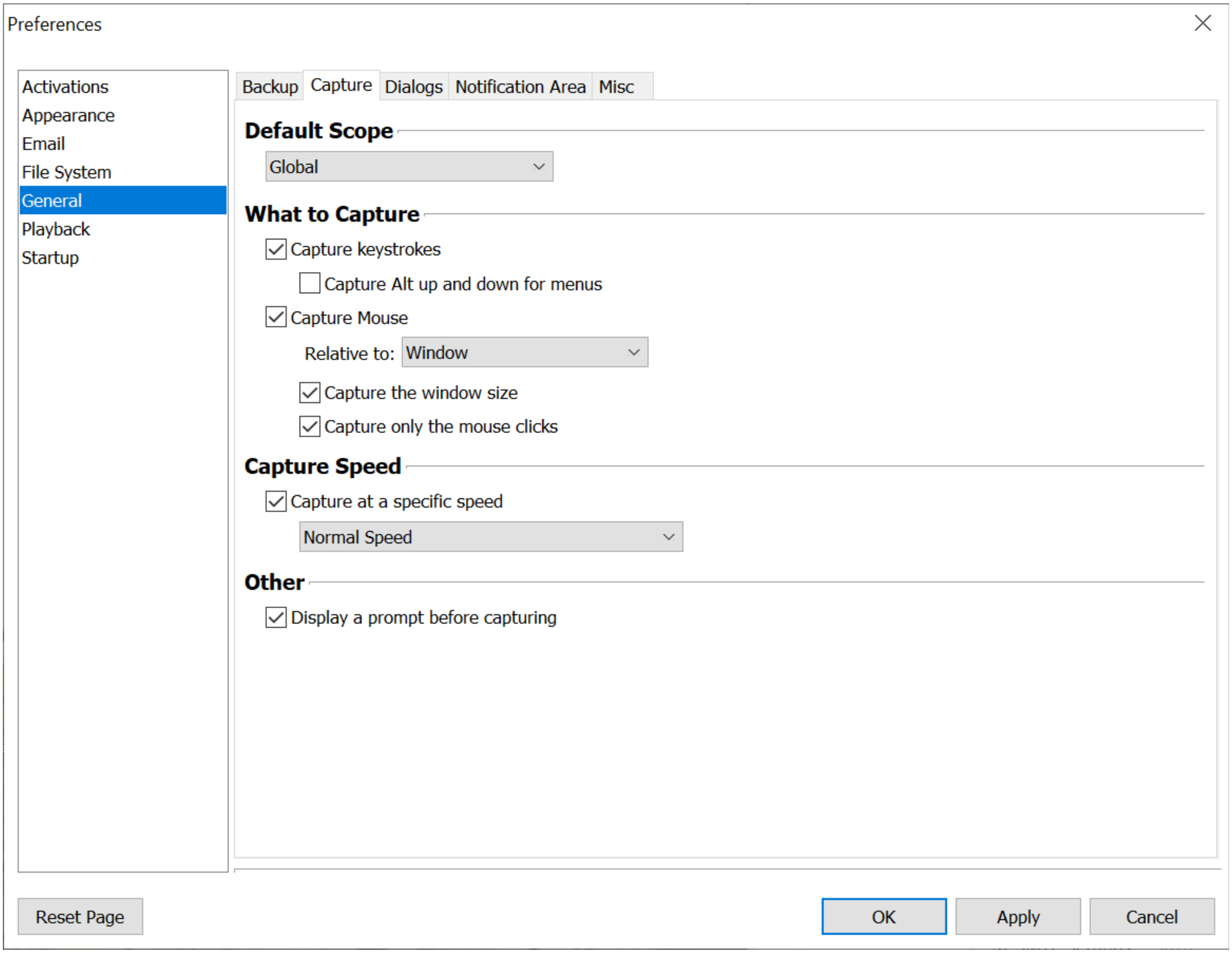
Backup Path

If using the Backup to a Specific Path option, click on the Browse button to select the location where the backup file will be stored. Or type in the full file path in the edit field.

Click on the Folder button to open the folder containing the backup files.

✓ **Note:** To manually back up the macro file at any time, open the Macro Express Explorer window and click on File > Backup.

Capture



Default Scope

Global

What to Capture

☑ Capture keystrokes

□ Capture Alt up and down for menus

☑ Capture Mouse

Relative to: Window

☑ Capture the window size

☑ Capture only the mouse clicks

Capture Speed

☑ Capture at a specific speed

Normal Speed

Other

☑ Display a prompt before capturing

Macro Express Explorer > Options > Preferences > General > Capture

Macro Express records macros by capturing mouse and keystrokes and saving this information for later playback as a macro.

Before starting the first recording, click on Options > Preferences > General - Capture tab from the Macro Express Pro - Explorer window to define the capture settings as explained below.

Start the capture

There are two methods for starting a macro recording.

1. Click on the Capture icon or Macro > Capture Macro from the menu

-- This method prompts to assign the macro activation before the recording.

2. Press the System Hot Key of Ctrl + Alt + Shift + X

-- This method prompts to assign the macro activation after the recording.

Stop the capture

There are three ways to stop the recording, depending on which method chosen to start the capture.

1. Press the System Hot Key of Ctrl + Alt + Shift + X
2. If using the Capture icon press the hot key assigned to the macro
3. Right click on the Camera icon in the Notification Area that displays while recording

---

Default Scope

The Global scope allows for the playback of the macro into any Windows program. The Program/Window scope indicates that the recorded macro will only start in the program or window selected. Or allow the macro to play back in all programs except the ones defined.

Capture Keystrokes

Select this option to record all keystrokes made during the capture process.

Capture Alt Up and Down for Menus

Some window menus do not always respond to a single press of the Alt key in a macro. This option records the Alt key twice to help ensure proper activation of the menu.

For example, to open the File menu in an application, press Alt + f during the capture. The macro script displays <ALT><ALT>f, even though the Alt key was only pressed once during the capture.

Capture Mouse Relative To

A check placed in this option records all mouse clicks and movements made during the capture process. There are several mouse options to choose from.

Relative to Screen - This option captures all mouse clicks and movements in relation to the cursor's position on the monitor.

Relative to Window - This option captures all mouse clicks and movements in relation to the application window that is open and has focus.

Capture Window Size is only available if the Relative to Window option is selected.
This option remembers the size of the window in which the capture occurs. When the macro plays back, the window is resized, if necessary, to the original size when the recording was made. This helps ensure that the mouse clicks and movements are accurate.

Relative to Last Position - This option captures the mouse clicks or movements based on the last position of the mouse cursor.

Capture Only the Mouse Clicks

Select this option to only capture the mouse clicks, including the position of the mouse click. This option captures the position where the mouse is to click and also the mouse click itself. No other movements of the mouse are recorded.

Unless drawing or recording the mouse dragging an item, it is best to turn on this option. It reduces the size of the macro considerably and makes it much easier to edit if necessary. It also helps improve the playback speed of the macro, since all of the interim mouse move commands and associated delays are eliminated from the macro.

Capture Speed

This option refers to how fast the macro plays back in relation to the recording speed.

Speed	Description
Actual Speed	Plays the macro back at the same speed as the macro was recorded. Any hesitations, etc. play back at the same speed as recorded.
Half Speed	Slows down the macro playback to half of the recording speed.

Double Speed	Speeds up the macro playback to twice the actual recording speed.
Quadruple Speed	Speeds up the macro playback to four times the actual recording speed.
Fastest Speed (no delays)	Plays back the macro with no hesitations, delays, etc. Remove the check next to the Capture at a Specific Speed box to eliminate delays from the macro. **Caution:** Selecting this option may cause the macro to play back too fast and throw off the timing of the macro.

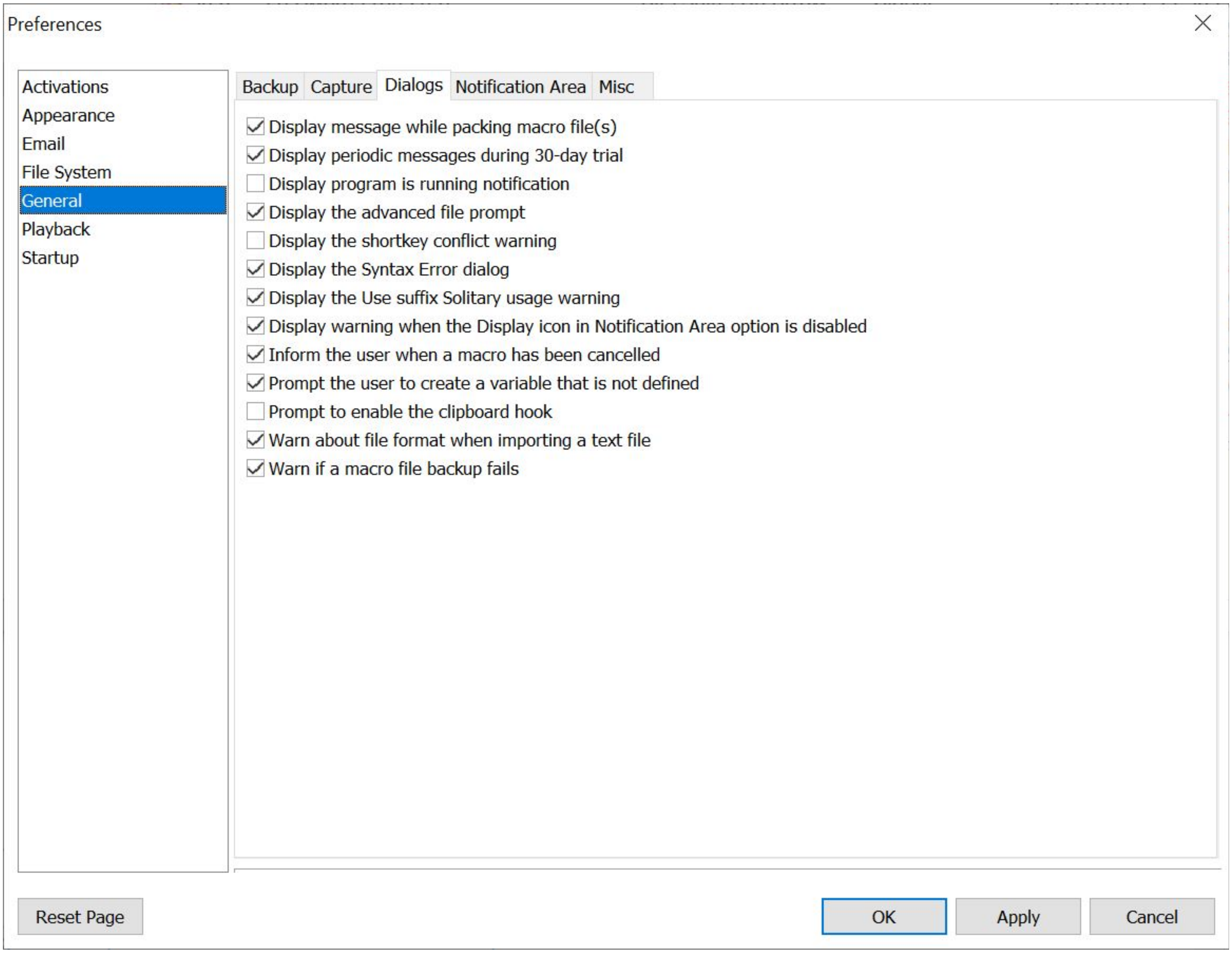
After the capture is complete, optionally modify the playback speed of the macro in the Script Editor or Direct Editor. Use the Macro Playback Speed command to do this.

Other

Display a Prompt before Capturing

With this option selected, a message is displayed indicating that the capture is starting.

Dialogs



☑ Display message while packing macro file(s)
☑ Display periodic messages during 30-day trial
☐ Display program is running notification
☑ Display the advanced file prompt
☐ Display the shortkey conflict warning
☑ Display the Syntax Error dialog
☑ Display the Use suffix Solitary usage warning
☑ Display warning when the Display icon in Notification Area option is disabled
☑ Inform the user when a macro has been cancelled
☑ Prompt the user to create a variable that is not defined
☐ Prompt to enable the clipboard hook
☑ Warn about file format when importing a text file
☑ Warn if a macro file backup fails

Macro Express Explorer > Options > Preferences > General > Dialogs

Display message while packing macro file(s)

Select this option to display the progress of the file packing. From the Macro Express Pro - Explorer window click on Tools > Pack and choose to pack the current file or all open .mex files.

Display periodic messages during 30-day trial

This option displays 5 reminder messages during the 30 day evaluation period.

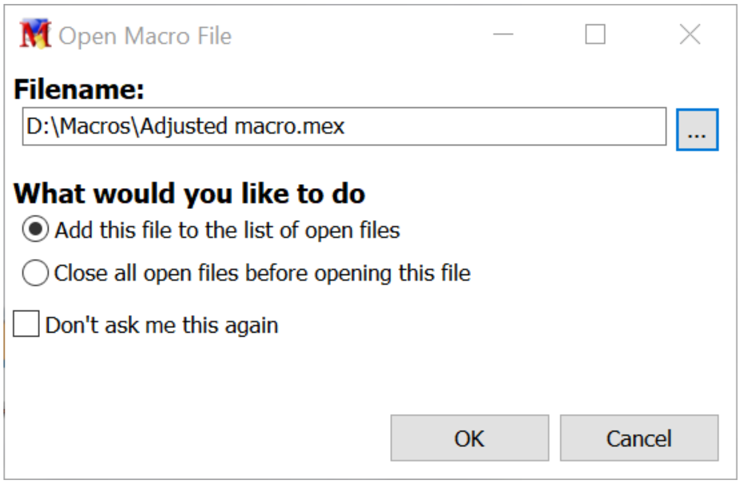
Display program is running notification

For first time users of the program a message appears above the Macro Express icon in the notification area. The message notifies users that the program is running and to click on the icon to open the Macro Express Explorer window. Clicking on the message causes it to not appear anymore.

Choose this option to have the message appear in the future.

Display the advanced file prompt

Select this option to open a dialog similar to the one below when opening an existing macro file or creating a new one. Since Macro Express allows more than one file to be open at a time, this dialog provides an option to decide whether to leave existing files open or close all open files before loading in the new one.



Filename:
D:\Macros\Adjusted macro.mex

What would you like to do

● Add this file to the list of open files

○ Close all open files before opening this file

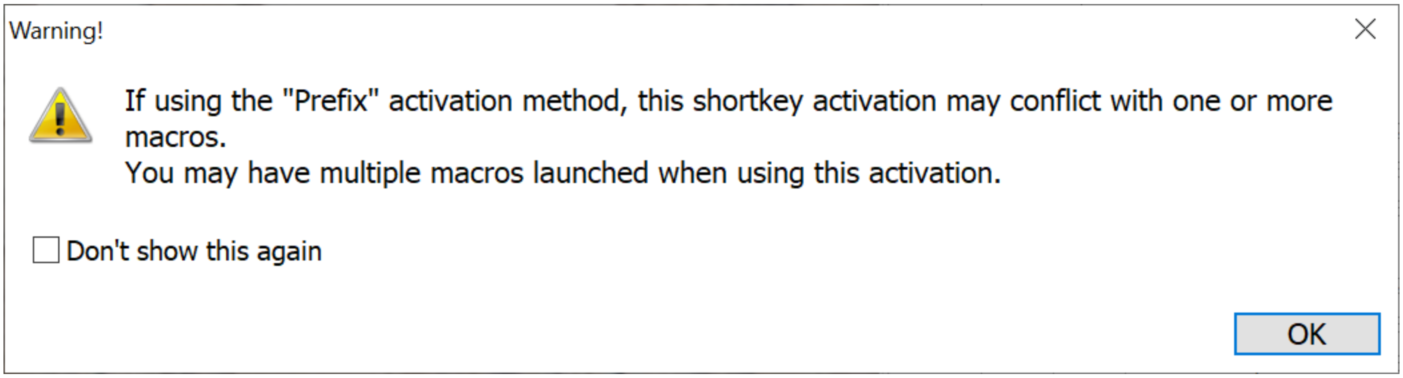
□ Don't ask me this again

Display the shortkey conflict warning

When using the Prefix key activation method for shortkey activated macros, certain shortkey combinations may conflict with each other. For example, if a shortkey activation is set to 'be' and a second activation is set to 'bet', 'bet' will never play back because 'be' will always play back before typing the last "t" in "bet". The suffix key option does not run into this potential conflict.

The warning message is just a warning of potential conflict. It will not prevent creation of the macro or use of the shortkey.

Select the Don't show this again option to not have the warning display in the future. The warning may be turned back on from the Dialogs preference panel as shown above.



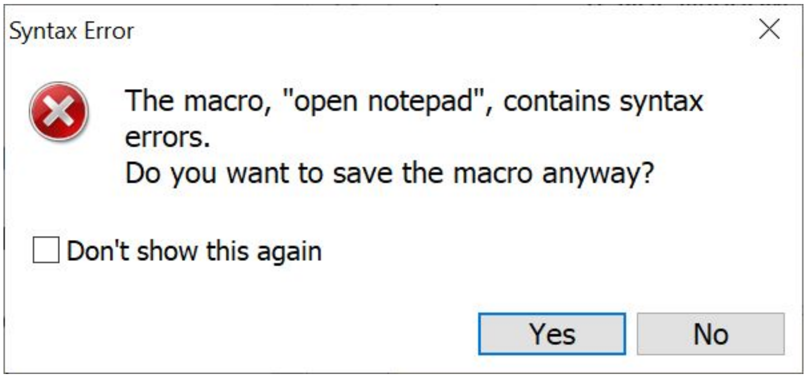
Warning!

If using the "Prefix" activation method, this shortkey activation may conflict with one or more macros.
You may have multiple macros launched when using this activation.

□ Don't show this again

Display the Syntax Error dialog

If you save your macro with a syntax error, such as a Repeat without an End Repeat or an If command without an End If, an error message will display warning you of the error.



Syntax Error

The macro, "open notepad", contains syntax errors.
Do you want to save the macro anyway?

□ Don't show this again

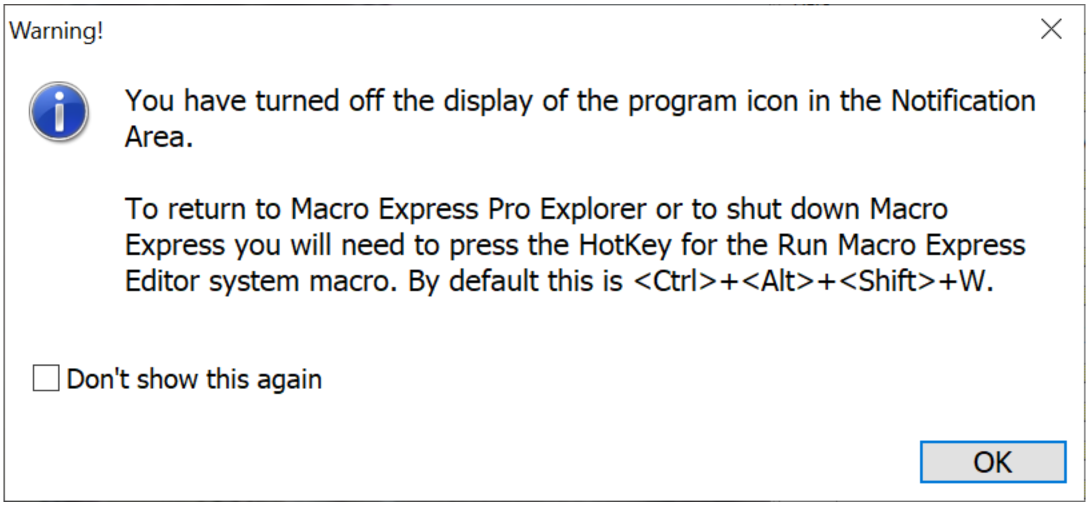
The warning is also displayed in the Messages tab of the Script Editor.

Display the Use suffix Solitary usage warning

If the Suffix Key option is selected for use with shortkey macros, this message is a reminder that you may want to turn on the Solitary usage option, if it is not already enabled.

Display warning when the Display icon in Notification Area option is disabled

If you elect to not have the Macro Express icon display in the notification area by going to the Options > Preferences > General - Notification Area tab, you will be shown the message below providing instructions for opening the Macro Express Pro Explorer window.


Warning!

You have turned off the display of the program icon in the Notification Area.

To return to Macro Express Pro Explorer or to shut down Macro Express you will need to press the HotKey for the Run Macro Express Editor system macro. By default this is <Ctrl>+<Alt>+<Shift>+W.

□ Don't show this again

Inform the user when a macro has been cancelled

Select this option to have a dialog box display when a running macro is cancelled.

Prompt the user to create a variable that is not defined

This option prompts a user to define a variable to be used in the macro, but no value has been assigned.

Prompt to enable the clipboard hook

The clipboard hook option found in the Activations > General tab of the preferences is required if using clipboard activated macros. This message alerts the user that the macro file contains macros that use the clipboard activation and the clipboard hook option is not enabled.

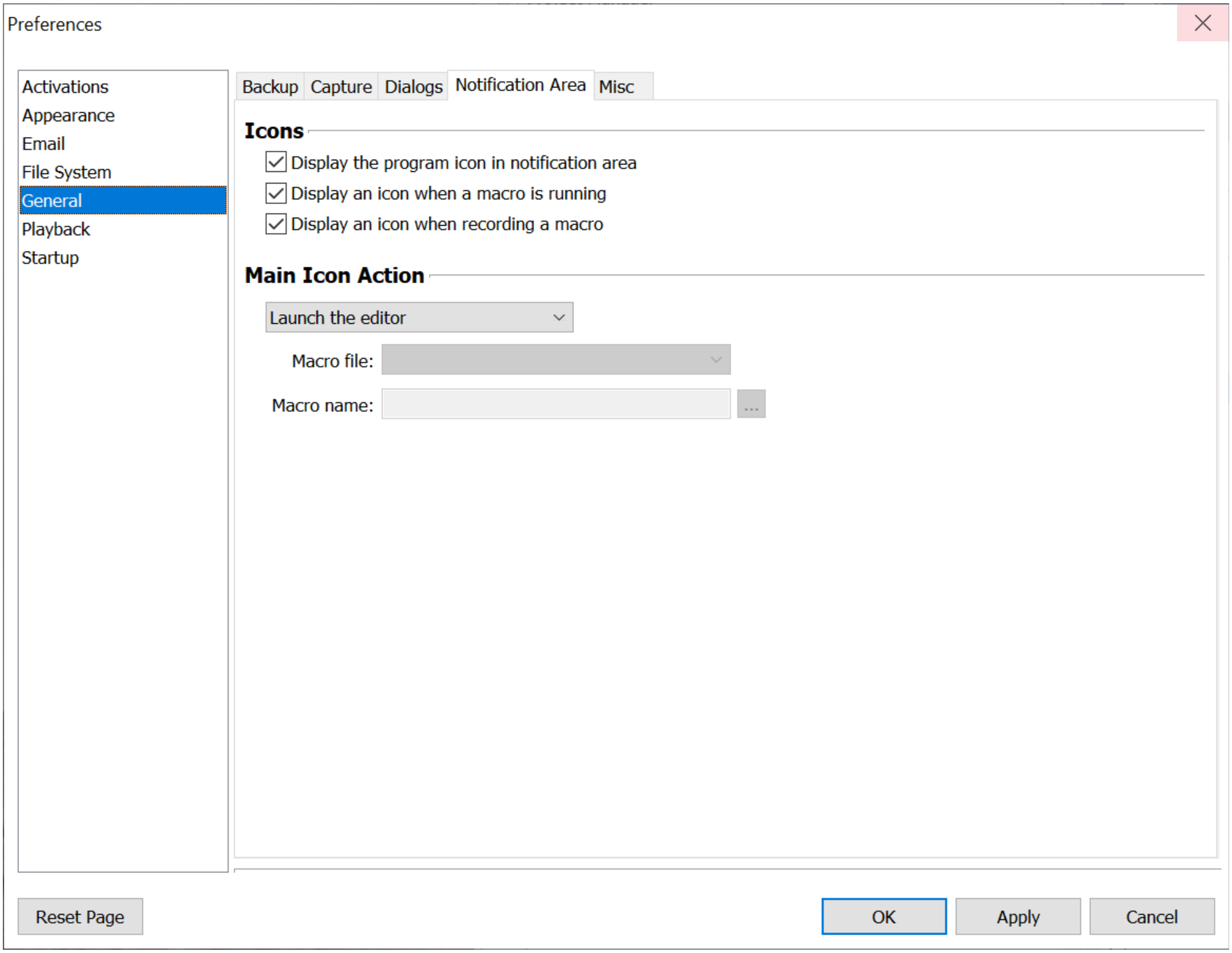
Warn about file format when importing a text file

Receive a prompt when importing a text file (File > Import > Import Text File) to be notified that the file must be in a specific format or the import process may not be successful.

Warn if a macro file backup fails

Displays a message to let the user know that the .mex file backup has failed.

Notification Area


Preferences

Icons

☑ Display the program icon in notification area

☑ Display an icon when a macro is running

☑ Display an icon when recording a macro

Main Icon Action

Launch the editor

Macro file:

Macro name:

Macro Express Explorer > Options > Preferences > General > Notification Area

This topic discusses items placed in the notification area and notification area icon functionality. The notification area refers to the small icons in the lower right hand corner of a conventionally configured screen (next to the clock).

Display icon in notification area

This option, when selected, places a Macro Express icon into the Windows notification area when the Macro Express Player is active. The icon in the notification area serves two primary purposes.

1. It readily identifies that Macro Express is running in the background.
2. It provides quick access to Macro Express.

--Left click on the icon to open the Editor.

--Right click on the icon to open a menu with options to open the editor, access the help file or tools, suspend the program, shut down the program, etc.

When not selected, Macro Express runs in hidden mode. This mode completely hides the macro editor. In the absence of a notification area icon, use the system macros that are defined for activating the editor and terminating the player. Ctrl Alt Shift W is the default hot key to open the editor.

Show macro running status

This option displays the "Running Man" icon in the notification area during the execution of a macro. It signals that a macro playback is in progress.

Show recording icon

This option displays the camera icon in the notification area during the recording (capturing) of a macro. It signals that a macro recording is in progress.

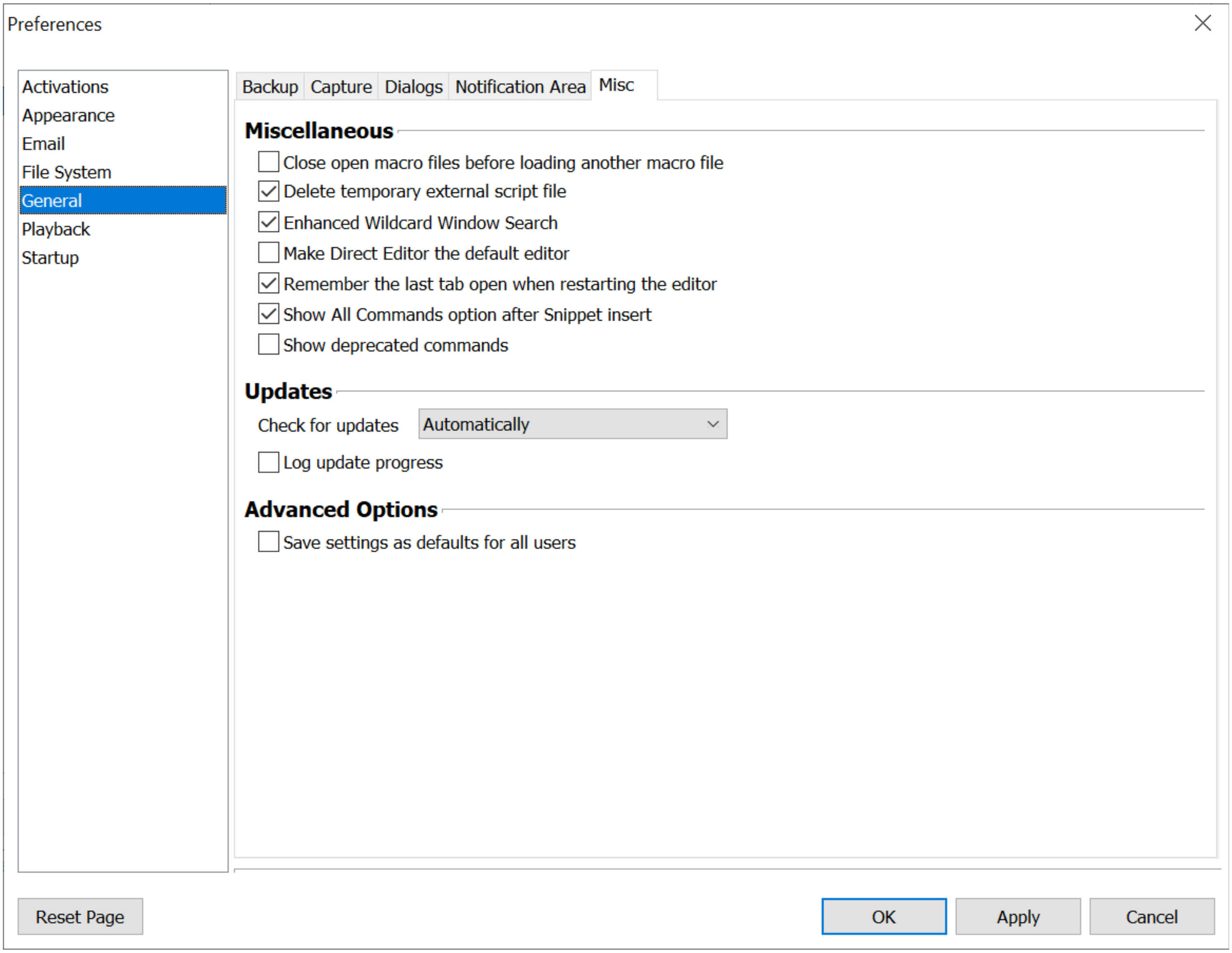
- ✓ **Note:** Windows 7 and later hides many of the notification area icons by default. Click on the Show Hidden Icons icon to display those hidden by Windows. The icon looks like a small triangle ▲ in Windows 7 and like this ^ in Windows 10. Other operating systems should be similar.

Main Icon Action

Use this option to change what happens when left clicking on the Macro Express icon in the notification area. Select from three options.

1. The default setting opens the Macro Express Explorer window.
2. Let the program do nothing when left clicking on this icon.
3. Run a macro. Specify the macro file and which macro to run. The macro runs each time the Macro Express icon in the notification area is clicked. If the specified macro file is no longer loaded when the icon is clicked, the macro file will be loaded. If the "Close open macro files before loading another macro file" option is enabled, then all other macro files will be closed.

Misc



Miscellaneous

☐ Close open macro files before loading another macro file

☑ Delete temporary external script file

☑ Enhanced Wildcard Window Search

☐ Make Direct Editor the default editor

☑ Remember the last tab open when restarting the editor

☑ Show All Commands option after Snippet insert

☐ Show deprecated commands

Updates

Check for updates: Automatically

☐ Log update progress

Advanced Options

☐ Save settings as defaults for all users

Macro Express Explorer > Options > Preferences > General > Misc

Miscellaneous

Remember the last tab open when restarting the editor

Select this option to have the Script Editor window remember which tab was last opened when the editor is closed. When reopening the editor, that tab will still be open. This function refers to the Script, Activations, Variables, Miscellaneous, Scope, Security or Notes tabs.

Close open macro files before loading another macro file

This setting instructs the program to close any macro (.mex) files currently open in the program when another macro file is loaded. The setting chosen will be the default when using the File > Open File menu from the Macro Express Explorer window. This setting also affects double clicking on a .mex macro file to load it into the program, loading a file from a command line, or the File > Reopen menu option.

Delete temporary external script file

When the External Script command runs, a unique file containing the script is created in the Temp folder. By default this file is deleted after the External Script command runs. In order to debug issues with the External Script command you may want see what was sent to the script processor. To do this, turn off this option to save the temporary files created by the External Script command.

- **Warning:** Selecting this option will leave a file in the Temp folder each time the External Script command runs.

Enhanced Wildcard Window Search

This option enhances the use of wildcards in window names when using the partial match option. It provides some additional functionality when the command is looking for wildcards.

Make Direct Editor the default editor

When creating or editing a macro, the Direct Editor will open by default rather than the Script Editor.

Show All Commands option after Snippet insert

After inserting a code Snippet into the macro script, the All Commands view is displayed in the Script Editor. With the option turned off, the Snippets view will remain displayed.

Show deprecated commands

Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future. Where possible we recommend that you avoid the use of deprecated macro commands.

By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option.

Below are the commands that have been deprecated.

Play CD-ROM	Stop CD-ROM	Pause CD-ROM

CD-ROM Next Track	CD-ROM Previous Track	CD-ROM Goto Track
CD-ROM Step	CD-ROM Back	Eject CD-ROM
Close CD-ROM Tray	Cascade Desktop Windows	Tile Desktop Windows Horizontally
Tile Desktop Windows Vertically	Windows Sizing Border	Dial-Up Networking
If Dial-Up Successful	Hangup	Suspend Computer
Repeat Counter		

Updates

Check for Updates

Use this option to have the program check to see if any applicable updates are available to download and install.

Automatically

Each time Macro Express is loaded the program will check to see if an update is available. If so, a notification will be displayed in the bottom panel of the Macro Express Pro Explorer window. Click on the notification to see a list of the changes and a link to start the upgrade process.

Manually

Open the Macro Express About box and it will check to see if any updates are available. If so, click on the update link to see a list of the changes and a link to start the upgrade process.

Disabled

The progress of the update is stored in a log file, easily accessible from the Macro Express Explorer window by clicking on Tools, Macro Logs.

Log Update Progress

Save the progress details of the update to a log file. From the Macro Express Pro Explorer window click on Tools and then Macro Logs to view the folder containing the log file.

Advanced Options

Save settings as defaults for all users

The preferences for Macro Express Pro are stored in each user's profile. An option set for one user will not be available for other users. When a new user begins to use Macro Express Pro the program's default preference settings are assigned to that user.

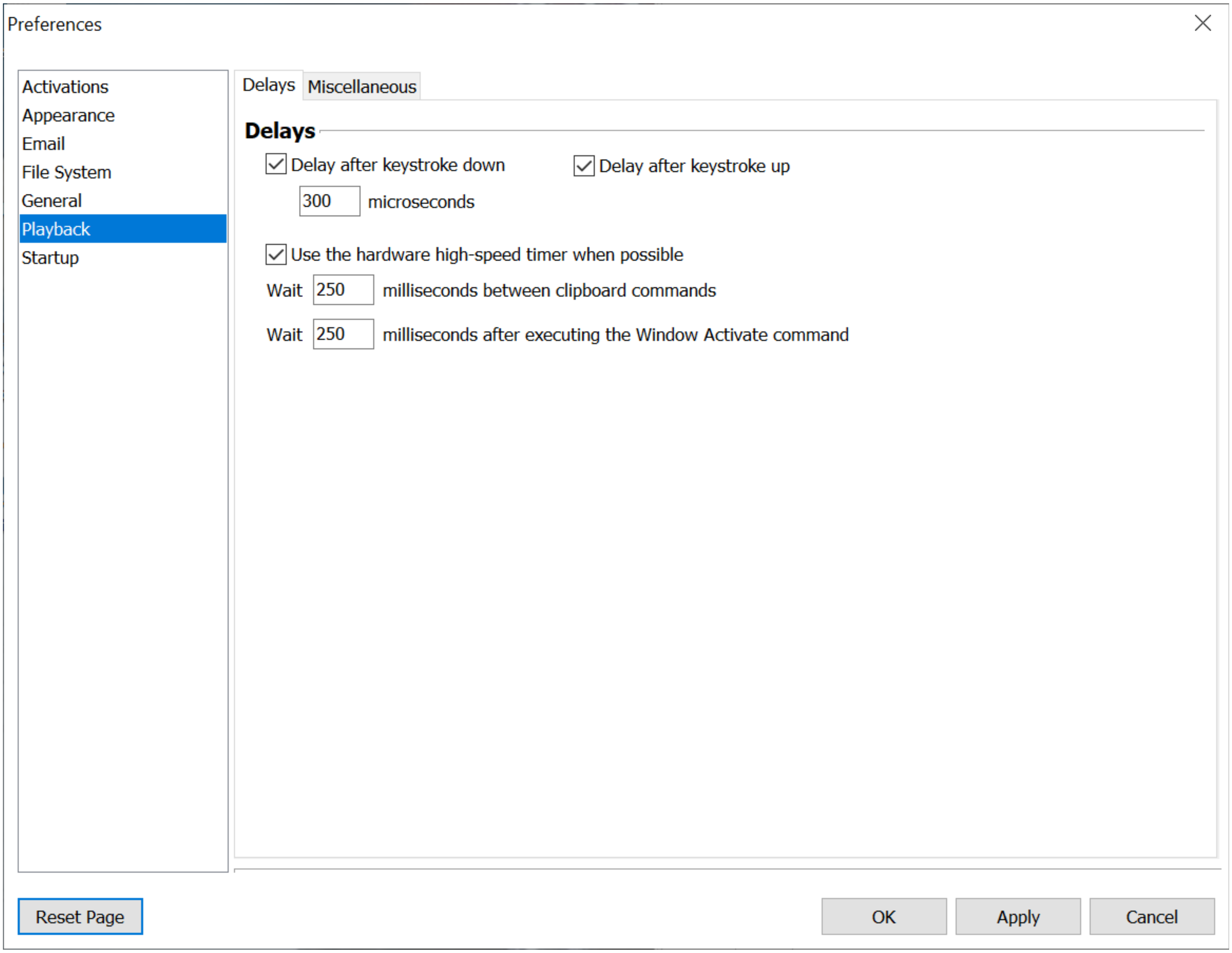
The "Save settings as defaults for all users" can be used to assign preference settings for other users of that computer. To do this, adjust the preference settings as desired. Then select the "Save settings as defaults for all users" check box and click the OK or Apply button. From that point on, other users of the computer will use the same default preference settings.

Unlike all other settings in the preferences, the “Save settings as defaults for all users” option is meant for a one time use. If this option is selected, when either the OK or Apply button is clicked, Macro Express Pro copies all of the settings from the HKCU area of the registry to the HKLM area. This is useful when preparing to Export Program Configuration to use later to import the preferences or to set the preferences when installing the program. The “Save settings as defaults for all users” option is always unselected when the Preferences dialog is opened.

Each user may change the preferences to suit their needs. Once changed, the default values will not apply when they are using Macro Express.

**The "Save settings as defaults for all users" option is not available in the portable version.

Delays



Delays

☑ Delay after keystroke down ☑ Delay after keystroke up

300 microseconds

☑ Use the hardware high-speed timer when possible

Wait 250 milliseconds between clipboard commands

Wait 250 milliseconds after executing the Window Activate command

Macro Express Explorer > Options > Preferences > Playback > Delays

Delays

When typing text on the keyboard each key is pressed and then released. Macro Express simulates this action for each keystroke played back in a macro. A delay between keystrokes may be required for longer strings of text, so that the text does not play back too fast and cause a Keyboard Buffer overflow. This may result in some of the text missing or scrambled.

These two delay options are global settings for all macros that play keystrokes. If you do find the text playing back too fast, increase the microsecond (millionth of a second) delay and test. Adjust as necessary.

Delay after keystroke down

A delay in the number of microseconds selected is applied after each keystroke is pressed down.

Delay after keystroke up (previously named Delay after keystrokes)

A delay in the number of microseconds selected is applied after each keystroke is released.

You may choose to use no delays, one of the delays or both. The default setting is 300 microseconds with both options turned on, which should be optimal for most systems.

Delay After Keystrokes in the Text Type Command

It is possible when typing back long strings of text that the Keyboard Buffer may overflow. When this happens some of the text may be missing or scrambled. Or the computer may emit an odd high pitched sound. In this situation, the text is trying to play back too fast. This Delay option is a global setting for all macros that use the Text Type command. The default setting is zero.

If the scrambled text or high pitched sound occurs when playing back this type macro, then adding a delay will help. The delay is in micro seconds, or thousandths of a millisecond. Start at 1000 microseconds, test the results and so on until finding the ultimate setting.

Use the Hardware High Speed Timer when Possible

Some computers make use of a high speed timer. Macro Express makes use of this timer in calculating delays. If the system does not use this timer, then leave this option unchecked.

On other systems the timer does not work properly. Do not select this option in this circumstance either.

Wait _____ Milliseconds between Clipboard Commands

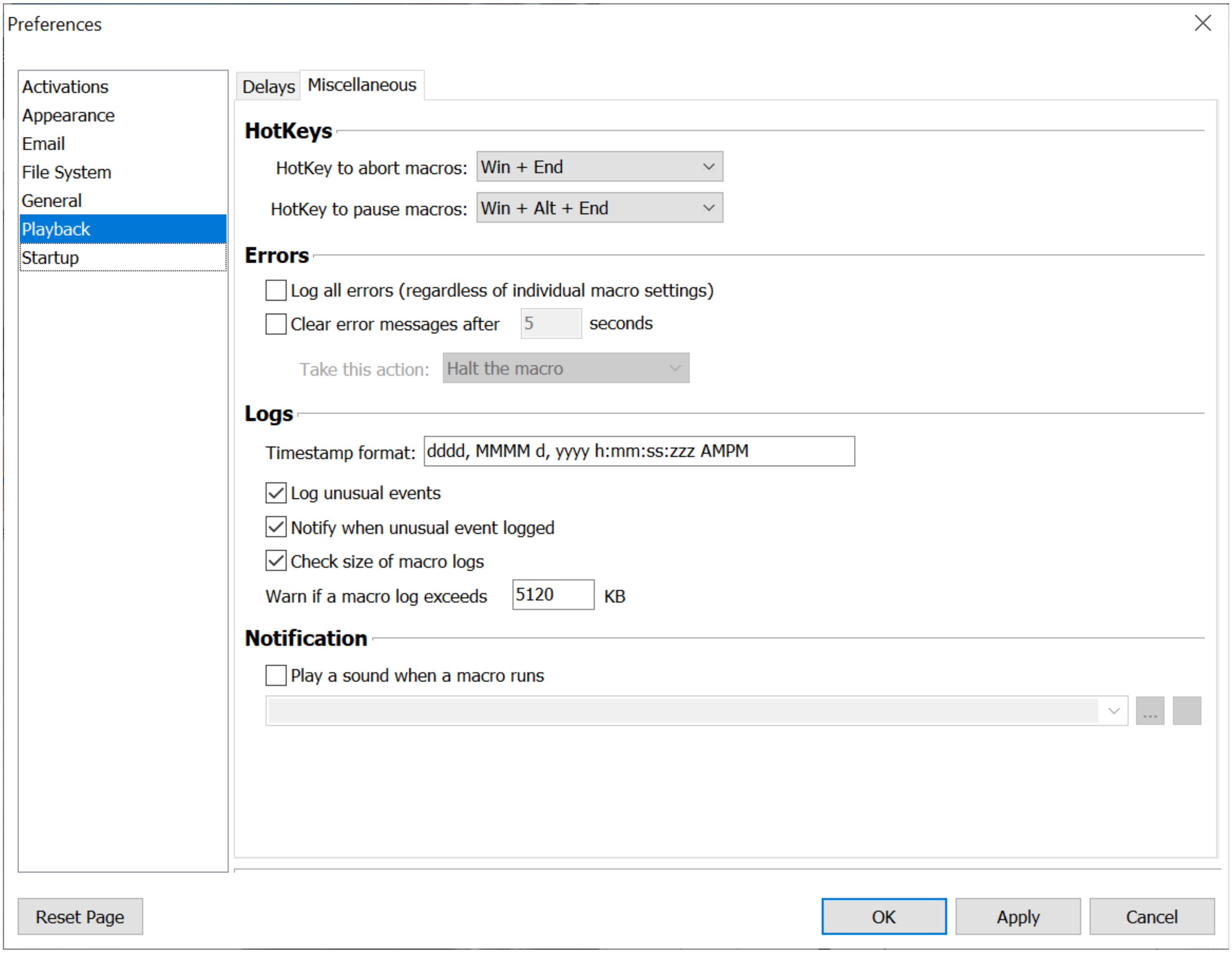
This option inserts a default 250 millisecond delay for each clipboard command in a macro. Windows requires a little bit of time to process a clipboard command. A 250 millisecond delay is adequate for most systems.

If the clipboard commands don't seem to be responding correctly, experiment with the timing and adjust the setting. This option also reduces the need to insert Delay commands after each clipboard command.

Wait _____ Milliseconds after Executing the Window Activate Command

This option instructs the macro to wait a minimum of 250 seconds after activating a window before continuing with the next macro command. The default setting is 250 milliseconds. This setting may also be adjusted as needed.

Miscellaneous


Preferences

HotKeys

HotKey to abort macros: Win + End

HotKey to pause macros: Win + Alt + End

Errors

☐ Log all errors (regardless of individual macro settings)

☐ Clear error messages after 5 seconds

Take this action: Halt the macro

Logs

Timestamp format: dddd, MMMM d, yyyy h:mm:ss:zzz AMPM

☑ Log unusual events

☑ Notify when unusual event logged

☑ Check size of macro logs

Warn if a macro log exceeds 5120 KB

Notification

☐ Play a sound when a macro runs

Macro Express Explorer > Options > Preferences > Playback > Miscellaneous

HotKeys

HotKey to abort macros

Choose from among three hot key combinations to stop a macro that is currently running. For example, if the Win + ~ combination is selected, press this combination to stop a running macro.

If displaying the Running Status icon when a macro runs, alternatively right mouse click on the icon to terminate the macro.

- ✓ **Note:** The Timed Delay in Seconds command and the Wait for Time to Elapse command suspend Macro Express for the amount of time specified. During this delay time the macro cannot be terminated.

Please note that macros that consist of numerous keystrokes can send the keystrokes rather quickly and Windows buffers up the keystrokes. Once such commands have been buffered in Windows, aborting a macro does not cancel them. In these cases, it may appear that the abort command did not work when in reality the macro has already finished issuing its commands and Windows is now processing them.

HotKey to pause macros

Select from among four hot key combinations to pause a macro that is currently running. When ready to resume the macro, press the original hot key or click on the Resume button of the Pause window that displays.

Errors

Log all errors (regardless of individual macro settings)

This is a quick way to log errors for all macros. There is no need to individually set the error logging for each macro through the macro properties. It automatically logs all information as long as the error logging commands are included in the macros. Each individual macro generates its own separate log file.

Clear error messages after _________ seconds

If an error message generated by the Macro Express Player occurs during the playback of a macro, this option closes the error message after the number of seconds designated. When not selected, manually click on the OK button of the error message to close the dialog.

The default setting to is leave this option unchecked, so that error messages are displayed until manually closed.

- ✓ **Note:** Setting the time to (0) zero seconds causes the error message to be displayed indefinitely.

The Clear Error Messages option may be preferable when several macros are scheduled to run unattended. If for some reason the first macro fails and displays an error message, the message closes automatically after the number of seconds specified. When the message window closes, the macro terminates and allows the next scheduled macro to run. Otherwise the first macro would not close, preventing subsequent macros from running as scheduled.

Typical error messages coming from the Player would be Time Out messages while waiting for windows, a file or folder wasn't found or a networking error.

Take this action

Set a default course of action to take when the error message occurs.

- *Halt the Macro* - When the error occurs the macro stops and terminates.
- *Ignore the Error* - The error is ignored and the macro continues. Most likely the command that generated the error is skipped.
- *Handle the Error* - Most of the macro commands have an error handling option. If using Error Handling commands in the macro script immediately following the command that generated the error, the macro will take the actions specified.

Macro Logs

Log timestamp format

This option formats the time stamp saved in the log file to any pattern preferred. The guide in the Date/Time help topic can be used for determining the abbreviations and setting the time stamp.

Log unusual events

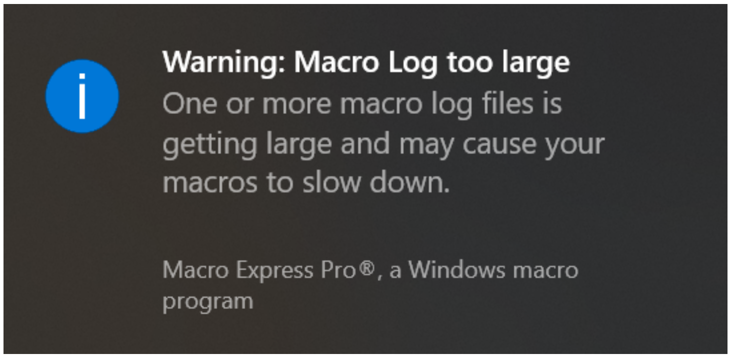
Any unusual events that occur when playing back a macro will be logged. The information will be stored in the same file where all macro log information is stored.

Notify when unusual event logged

This option, enabled by default, displays a notification when an unusual event is logged.

Check size of macro logs

The macro log files can become quite large over time and potentially cause performance issues. This function will warn you if one of the log files is getting too large. A message appears on startup of Macro Express stating that one or more log files may be getting too large.



Warning: Macro Log too large
One or more macro log files is getting large and may cause your macros to slow down.

Macro Express Pro®, a Windows macro program

Warn if a macro log exceeds _____ KB

Enter the size in KB of the macro log file. When the file exceeds this size, the warning will appear the next time Macro Express is started. The default size is 5120 KB.

The options are to ignore the error, delete the log to let it start over, rename the existing log to retain the log information or edit the log file and delete the older entries.

Play a sound when a macro runs

Use this function to help debug macros. When testing a macro, this will alert that the macro has successfully started. When finished testing, you may want to turn off this option so that the sound doesn't play every time a macro starts.

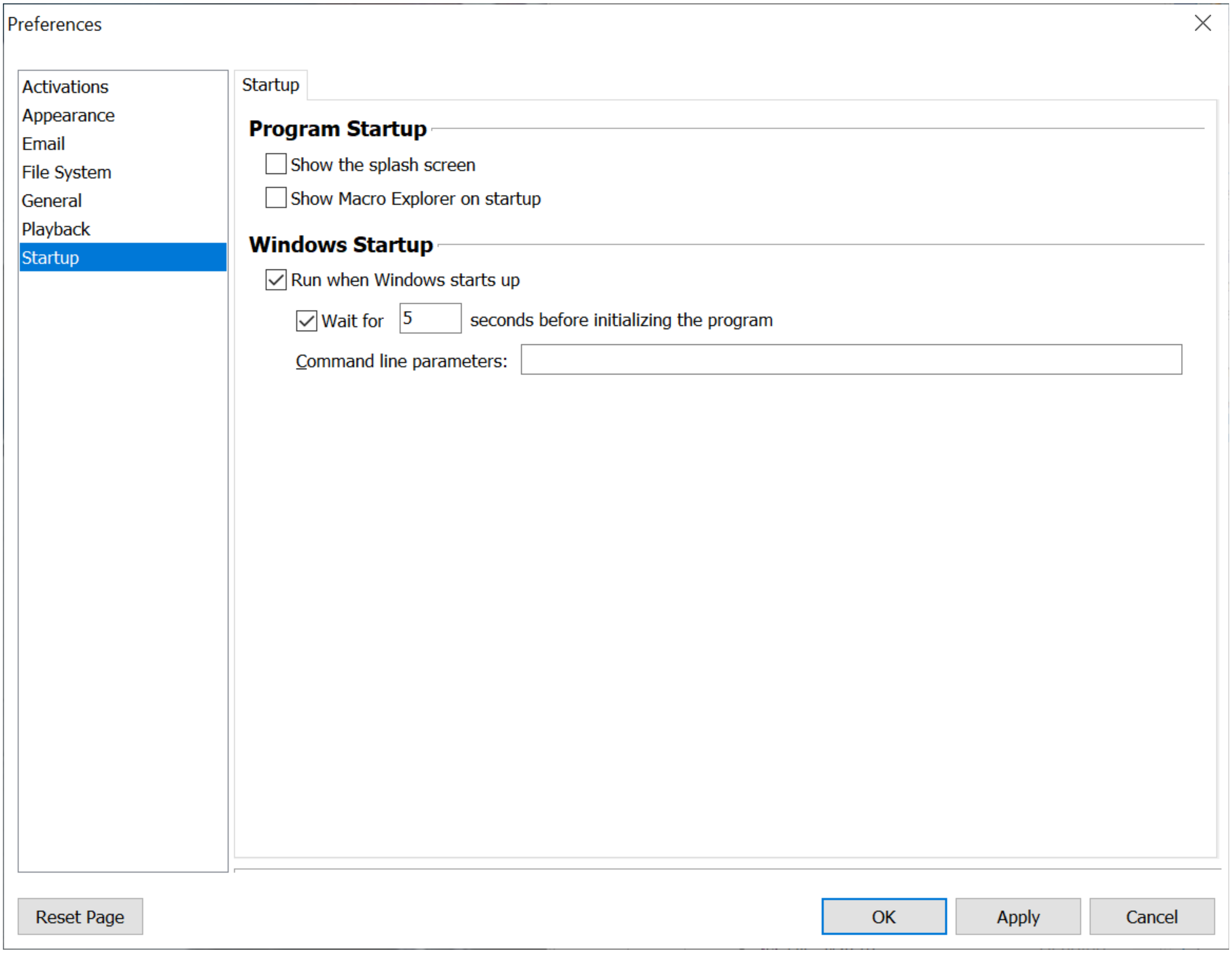
To select a sound file click the Browse button to locate a .wav file on the computer or type the full path and file name of the .wav file.

Or select one of the system sounds available on the computer by choosing from the drop down list. A system sound is a sound assigned to specific Program Events within Windows such as Asterisk, Device Connect or Windows Logoff. The sounds assigned to these events are selected in the Windows Control Panel either individually or as part of a theme. Not all system sounds listed in the Control Panel are available in this command.

The Default Beep sound, defined in the Control Panel, plays if a system sound is not defined or no sound is assigned to the specified system sound in the Control Panel.

Click on the Play button to listen to the sound. This is a handy way to quickly test if the correct sound was selected. (The sound is played immediately upon clicking the button.)

Startup



Program Startup

□ Show the splash screen

□ Show Macro Explorer on startup

Windows Startup

☑ Run when Windows starts up

☑ Wait for 5 seconds before initializing the program

Command line parameters:

Macro Express Explorer > Options > Preferences > Startup > Startup

Program Startup

Show the splash screen

Displays the Macro Express Splash screen for a few seconds while the program is loading.

Show Macro Explorer on startup

Instructs the Macro Express Pro - Explorer to load and display when Macro Express starts up.

Windows Startup

Run when Windows Starts Up

This option instructs Macro Express to load when Windows starts.

Wait for _____ Seconds before Initializing the Program

On some computers Windows is not completely ready for Macro Express when it tries to load the program. This may be caused by other programs that are loaded when Windows starts.

This option instructs Macro Express to wait the number of seconds specified to ensure that Windows is fully loaded. Most computers do not need this delay. If necessary, experiment with the amount of time to wait.

Command Line Parameters

Macro Express provides a number of command line parameters that are available for use at program startup. Enter the parameters in this edit field so that they will be in effect when the program starts on Windows startup.

For example, to have the program run in Browse Mode and in Network mode, insert the following into the edit field: /B /N

**The Windows Startup options are not available in the portable version.

Pack

Saving macros and deleting macros from a macro file creates holes in the macro file structure or causes it to become fragmented. The Pack command compresses the file and fills in the holes created by the changes to the file. This is similar to performing a defrag on a hard drive, only it will take less time.

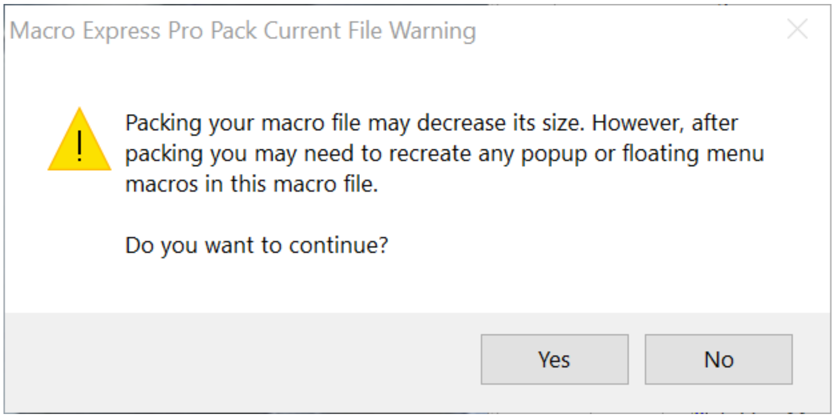
Pack Current File

Select this option to only pack the macro file that is currently in use.

Pack All Files

Select this option to pack all macro files that are currently loaded into Macro Express.

You may see the warning below as you begin the Pack process. Since Macro Express uses index numbers to track each macro in the file, packing may reorder the index numbers. This may cause the macros in the popup or floating menus to be reordered.



Macro Express Pro Pack Current File Warning

Packing your macro file may decrease its size. However, after packing you may need to recreate any popup or floating menu macros in this macro file.

Do you want to continue?

Macro Express - Explorer > Tools > Pack

Restore Keyboard & Mouse Hooks

It is possible for other programs which are run after Macro Express to improperly install themselves into the system wide Windows keyboard hooks. This causes Macro Express to no longer be able to monitor keystrokes and/or mouse events. When this occurs, Macro Express will not recognize a hot key, shortkey or mouse activated macros.

This tool reinstalls the Windows keyboard and mouse hooks used by Macro Express. It provides Macro Express with the first look at keystrokes and mouse movements/clicks in order to respond to hot keys, shortkeys or mouse activated macros.

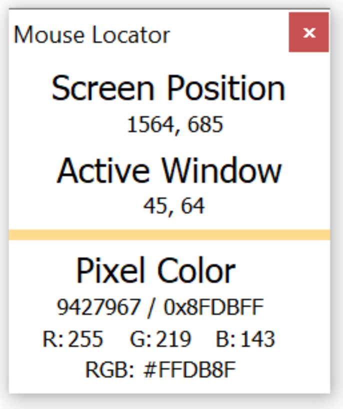
Macro Express - Explorer > Tools > Restore Keyboard & Mouse Hooks

Launch Mouse Locator

The Mouse Locator tool displays the X and Y coordinates of the mouse pointer relative to the entire screen or to the window that is active. This is useful when wanting to include mouse clicks in macro scripts.

The locator displays the exact position of the mouse in order to plot those coordinates into a macro. Just plug in the coordinates when using the Mouse Commands found in the Scripting Editor or Direct Editor. (This function is not necessary if using the Capture option.)

Also activate this window by clicking on the Launch Mouse Locator button provided in the Mouse Move command or the Get Pixel Color command.



Screen Position
1564, 685
Active Window
45, 64
Pixel Color
9427967 / 0x8FDBFF
R: 255 G: 219 B: 143
RGB: #FFDB8F

Macro Express Explorer > Tools > Launch Mouse Locator

If the Mouse Locator window pops up in an area of the screen that covers important data, click on the window, hold the left mouse button down on the title bar at the top of the window and drag to another area of your screen.

Screen Position

The set of numbers directly under Screen Position lists the position of the mouse relative to the computer screen. When the mouse is moved, the numbers change in relation to the mouse cursor position. The number to the left is the X coordinate, or how many pixels from the left edge of the screen. The number on the right is the Y coordinate, or how many pixels the mouse is positioned from the top of the screen.

Active Window

The set of numbers directly under Active Window lists the position of the mouse relative to the active window or the window that has focus. When the Mouse Locator window is first launched, it receives focus. Clicking on a window outside of the Mouse Locator gives focus to that window. The Mouse Locator window remains on top in order to easily read the coordinates for the window with focus.

When the mouse is moved, the numbers change in relation to the mouse position. The number to the left is the X coordinate, or how many pixels the mouse is positioned from the left edge of the active window. The number on the right is the Y coordinate, or how many pixels the mouse is positioned from the top of the active window.

Pixel Color

The color of the pixel to which the mouse is pointing is displayed in a thin bar above the Pixel Color header.

Windows assigns a numeric value to each color, normally from 6 to 8 digits in length. The left side is the decimal value. This is the value returned by the Get Pixel Color command. Use the decimal value in macros that compare color values. The right side displays the hexadecimal value in BGR format.

The middle row displays the pixel color in separate values for Red, Green and Blue. These values are used when entering colors in programs, such as word, that accept separate values for Red, Green and Blue.

The bottom row displays the hexadecimal pixel color value in RGB format.

Which Pixel Color formats are displayed can be adjusted using the Pixel Color options found in Options, Preferences, Appearance, Mouse Locator.

How colors are displayed

To display colors, computers use the RGB Additive Color System. The letters R, G and B represent Red, Green and Blue respectively. Separate values are used for the amount of Red, Green and Blue. On a Windows computer, each color ranges from 0, all the way off, up to 255 (or FF hexadecimal) all the way on. Combining different values for each basic color allows millions of different colors to be displayed. For example, White has a value of 000000, Green has a value of 00FF00 and black has a value of FFFFFF.

The hexadecimal number that represents a color can also be represented as an integer value. For example, one orange color has a hexadecimal value of 2D42D4. When displayed as an integer this value is 2966228. The Get Pixel Color command returns the integer value of the color for a given pixel. This allows the integer variable commands to test for specific colors in your macros.

The Mouse Locator displays colors in a variety of formats to make it convenient to enter colors into other programs. For example, when entering a color for a web page the color value is in hexadecimal RGB format. To enter a color in Word requires separate integer values for Red, Green and Blue. And some programming languages require a hexadecimal BGR (Blue, Green, Red) format.

Copy to Clipboard

You can use the Mouse Locator to get the value of a color on your computer monitor. Make sure the Mouse Locator is the active window, then move your mouse over the color you want and press Ctrl-c. The color will be copied to the clipboard so you can paste it into another program.

By default, several values are copied to the clipboard including the Screen Position of the mouse and the Pixel Color in a variety of formats. Here is an example:

```
990,385,2966228,2D42D4,212,66,45,#D4422D
```

What is copied to the clipboard can be adjusted using the Copy To Clipboard options found in Options, Preferences, Appearance, Mouse Locator.

Restore Program Defaults

Clicking on Tools > Restore Program Defaults from the Macro Express Pro - Explorer menu restores the program settings back to their default values. This primarily affects any settings made in the Options > Preference area. Other things that may be reset include the default macro file that opens, macro sort order and categories.

Macro Express - Explorer > Tools > Restore Program Defaults

Import Program Configuration

This option imports all of the program configuration settings into Macro Express, if the configuration settings were previously exported to a file. The file replaces the current configuration settings in Macro Express. This would include any Preference settings selected, macro file in use, etc.

Click on Tools > Import Program Configuration to open a browse window to navigate to the file previously exported. Highlight the file and import.

This is a handy way to save program settings if planning to migrate to another computer. Easily import the settings back into the program after installing Macro Express on the other computer. If more than one user shares a computer, each user could save their own configuration settings and load them in while they are at work on the computer.

Another option allows the individual settings to be loaded at startup for each user by setting a Command Line Parameter. Each individual user would need to create their own startup shortcut in order for this to work correctly.

- ✓ **Note:** Importing the program configuration from a newer version of Macro Express into an older version of the program may cause problems. The newer configuration file may contain additional data that is not supported in the older version of Macro Express.

Macro Express - Explorer > Tools > Import Program Configuration

---

**This import option is not available in the portable version.

Export Program Configuration

This option saves all of the program configuration settings to a file. This includes any Preference settings selected, macro file in use, etc. Click on Tools > Export Program Configuration to do this. A Save File box appears prompting to name the configuration file and select the file location. Use the Tools > Import Program Configuration to retrieve the file and reload the settings.

This is a handy way to save program settings if planning to migrate to another computer. Easily import the settings back into the program after installing Macro Express on the other computer. If more than one user shares a computer, each user could save their own configuration settings and load them in while they are at work on the computer.

Another option allows the individual settings to be loaded at startup for each user by setting a Command Line Parameter. Each individual user would need to create their own startup shortcut in order for this to work correctly.

Macro Express - Explorer > Tools > Export Program Configuration

**This export option is not available in the portable version.

Sent Email Log

If keeping a log of each email sent through Macro Express, click on Tools > Sent Email Log to view the emails sent. A window opens showing the subject, recipient(s) and date of the email. Highlight an email and click on the View button to display its contents.

Macro Express - Explorer > Tools > Sent Email Log

Macro Logs

The Macro Logs menu item opens the folder where the macro logs are stored. Click on any of the files located in the folder to review the log details.

From the Macro Express Explorer window click on Options > Preferences > File System, File Paths tab to edit or change the location of the log files.

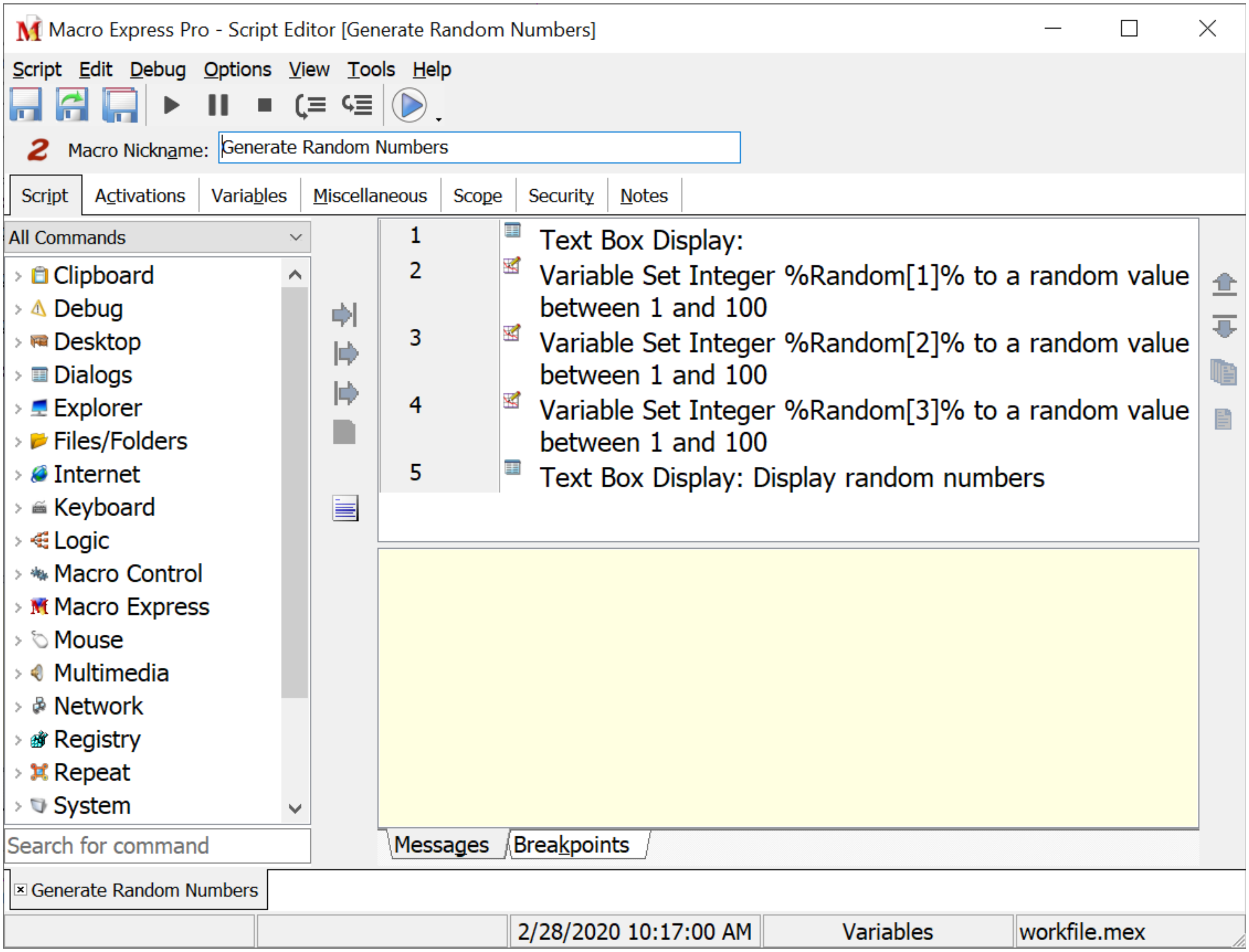
Macro Express - Explorer > Tools > Macro Logs

Building Macros with the Editors

Script Editor

Use the Script Editor to build a macro or to edit a macro already created. The Script Editor is divided into three main sections, the Commands, the Macro Script and the Debugging tabs. Select the commands to insert into the Macro Script. The Macro Script is written so that the top line in the script is the first command that plays back in the macro.

✓ **Note:** More than one macro may be open in the editor at the same time. Tabs at the bottom of the window allow for navigation between open macro scripts. Either click on the tab to open or press Ctrl Alt Left Arrow or Ctrl Alt Right Arrow to move between the tabs.



Macro Express Pro - Script Editor [Generate Random Numbers]

Macro Nickname: Generate Random Numbers

Text Box Display:
Variable Set Integer %Random[1]% to a random value between 1 and 100
Variable Set Integer %Random[2]% to a random value between 1 and 100
Variable Set Integer %Random[3]% to a random value between 1 and 100
Text Box Display: Display random numbers

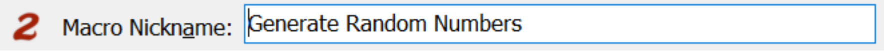
2/28/2020 10:17:00 AM | Variables | workfile.mex

Macro Express Explorer > Create a new macro or edit an existing macro

The Script tab contains the list of commands and the macro script. Additional information about the macro can be viewed or edited in the Activations, Variables, Miscellaneous, Scope, Security and Notes tabs.

Macro Icon and Nickname

Located near the top of the editor window is the Macro Nickname field and a macro icon. In the example below, the icon is a red number two.



Macro Nickname: Generate Random Numbers

Macro Icon

The icon (in the example above, a blue number one) is displayed to the left of the macro nickname. Click on the icon to open the Select an Icon window. Click on the browse button to open a folder with a number of icon options. Or navigate to another location on the hard drive to find other icons on the computer. If an icon was not selected at the time the macro was created, the default Macro Express icon is displayed.

Macro Nickname

The nickname is used to help remember the function of the macro. If a nickname was entered when assigning the macro activation, the nickname is displayed as shown in the image above. The nickname may be changed in this field if desired.

Command View

Select from the drop down menu which command view to display.

All Commands

This option displays all of the commands that may be used in Macro Express. To insert a command into the script, scroll until finding the command category wanted. Click on the category to expand the list and display all of the commands available for that category.

Double click on the command to display a window to define what the command should do. Then click OK on the new window to insert this command into the Macro Script.

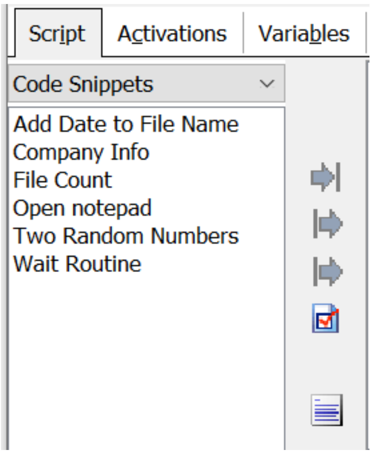
Double clicking on some of the commands that do not require any definition inserts the command directly into the Macro Script, without an intermediary window.

Favorite Commands

This option displays just the commands that are assigned as favorites, making it easier to find the most frequently used commands. From the Script Editor menu click on Options > Customize Favorites to assign the favorites.

Code Snippets

Use the Code Snippets function to create snippets of a macro script, such as routines for use in multiple macros. Save the code snippet to later insert into a macro at any time. Select Code Snippets from the drop down list as shown below to view, create or insert snippets.



Creating a Snippet

Select (highlight) one or more lines of a macro script in the Script Editor. Right mouse click on the highlighted lines and left click on Add to Snippets from the menu that opens. (Or click on Edit > Add to Snippets from the menu.)

Inserting a Snippet into the Script

Highlight the Code Snippet to use and click on one of the green insert buttons to insert the snippet into the script. The Macro Script Navigation topic below defines where in the script the snippet will be placed.

Search for command

Use this function to quickly locate a specific command within the Commands list. Type the first few letters of the command to find it. To look for the Password Protection command, enter "pa" without the quotes. This highlights the Password Protection command within the list of commands. Typing "pau" without the quotes moves the highlight to the Pause command.

Macro Script

This area lists each of the commands that the macro will perform. The macro plays back sequentially through the list beginning with the top item. Edit any of the commands in the list by double clicking on the line of the script to be edited.

There is also a right mouse click context menu available for the items in the Macro Script. Highlight the line or lines of the script needed and then right mouse click. A context menu appears with a variety of options, also available through the Scripting Editor menus.

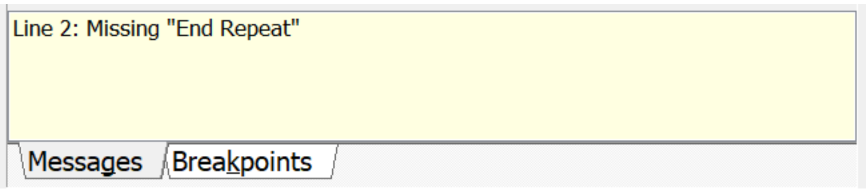
One item unique to the context menu is the Edit Macro option. Right mouse click on a Macro Run command in the script and select the Edit Macro option. The contents of the macro from the Macro Run command are displayed in a separate tab in the Script Editor.

Debugging Tabs

The Debugging Tabs located near the bottom of the Editor window provide information when attempting to debug a macro. Also click on Debug > View Variable Values or press the F5 key to see a display of all variables used in the macro and the values saved to those variables. These are displayed when running the macro in Test Mode.

Messages

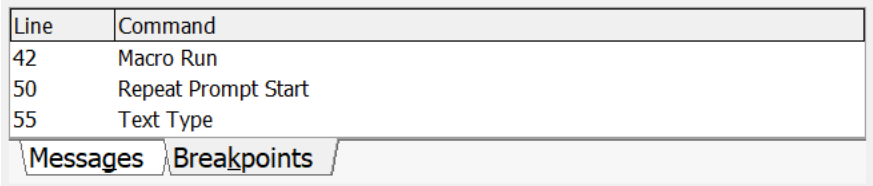
Any syntax errors or other problems with the macro are displayed in the Messages tab when saving a macro. This tab will alert you to potential problems with the macro script, such as having a Repeat Start command without a matching Repeat End, or an If command without a matching End If.



Line 2: Missing "End Repeat"

Breakpoints

This tab displays any breakpoints that were defined in the macro for debugging purposes. The line in the script and the command with the breakpoint are displayed.



Line	Command
42	Macro Run
50	Repeat Prompt Start
55	Text Type

Macro Script Navigation

This section describes the functionality of the buttons available to help build and modify the macro script.

Button	Function	Description
➡|	Insert Command Before Selected	Inserts the macro command selected from the Commands list above or before the command that is highlighted in the Macro Script edit box. If a script item is not highlighted in the edit box, the macro command is placed as the first or uppermost item in the script.
|➡	Insert Command After Selected	Inserts the macro command selected from the Commands list below or after the script item that is highlighted in the Macro Script edit box. If a script item is not highlighted in the edit box, the macro command is placed below or after the first or uppermost item in the script.
|➡	Add Command to End of Script	Inserts the macro command selected from the Commands list as the last item in the script, below all other items.

☑	Disable Command	Highlight a line in the Macro Script and then click on this button. It disables the highlighted line in the script, so that it does run during macro playback. A red line is drawn through the disabled script, for a quick visual acknowledgment.
☑	Enable Command	Highlighting a line in the Macro Script that is disabled displays this symbol. Click on the button to enable the line in the script to allow the command to run during macro playback.
≣	Direct Editor	Switch from the Script Editor view to the Direct Editor.
⬆	Move Command Up	Moves the highlighted command up one spot in the Macro Script, ahead of other script items already in place. Each click moves the command up one space in the script. Also highlight several consecutive lines in the script and move them all at the same time.
⬇	Move Command Down	Moves the highlighted command down one spot in the edit box, beneath the other script items already in place. Each click moves the command down one space in the script. Also highlight several consecutive lines in the script and move them all at the same time.
⧉	Duplicate Macro Command	Highlight one or more lines of the macro script and click on the Duplicate button. This adds another copy of the command line or lines to the script. This is essentially a one step method for copy and paste. Or press CTRL D to duplicate the highlighted command or commands.
🗎	Insert Comments	Highlight a line in the macro script and click on the Insert Comments button or press Ctrl Alt C to open a Comments edit box. Type any comments about the line in the script into this box and click OK. The comments are inserted next to the command listed in the script.

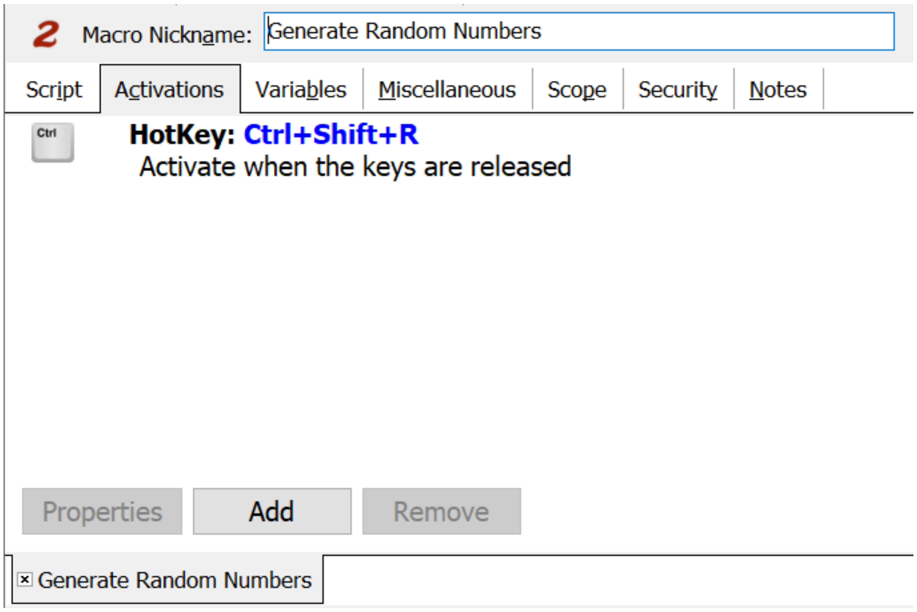
Keyboard Shortcuts

These are the Keyboard Shortcuts available in the Script Editor.

Keystroke	Action
Alt+Right Arrow	Move focus to the macro script area
Alt+Left Arrow	Move focus to the All Commands/Favorite Commands/Co
Alt+Down Arrow	Move the text cursor to the Search for command box
Ctrl+Alt+Left Arrow	Move to the previous macro tab
Ctrl+Alt+Right Arrow	Move to the next macro tab

Activations

Typically the activation of a macro is assigned when the macro is first created. However, this screen may be used to change the macro activation or assign more than one activation method to a macro. Activations and details about the activations are listed in this tab.



Macro Nickname: Generate Random Numbers

HotKey: Ctrl+Shift+R
Activate when the keys are released

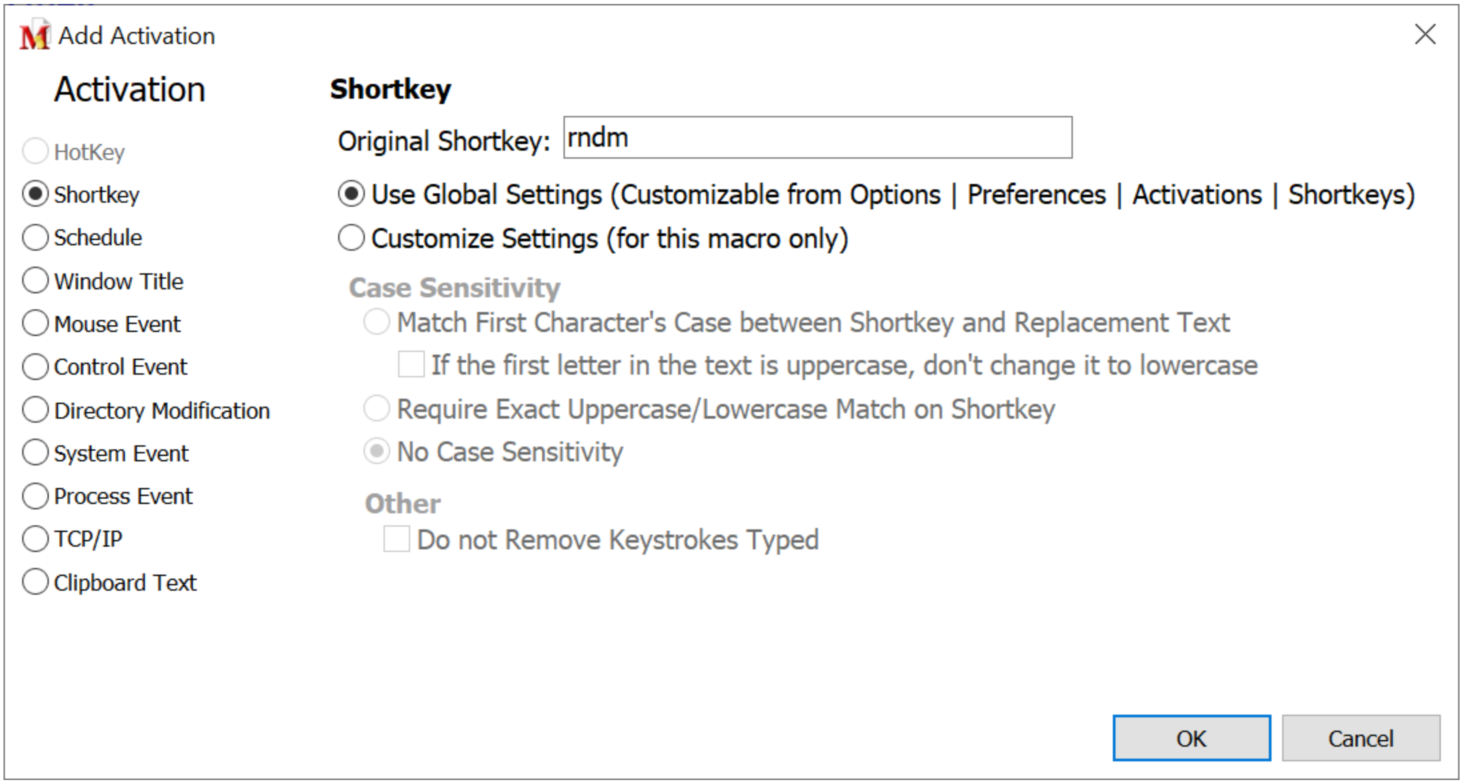
Script Editor > Activations Tab

Properties

To view the Properties, highlight an activation listed and click on the Properties button. This opens an edit field where changes may be made to the activation selected. For example if the activation is a hot key of Ctrl+Shift+R, assign the activation to another combination.

Add

Click on the Add button to open the Add Activation window, as shown below. Use this window to add a second or third activation type to your macro.


Add Activation

Activation

○ HotKey
● Shortkey
○ Schedule
○ Window Title
○ Mouse Event
○ Control Event
○ Directory Modification
○ System Event
○ Process Event
○ TCP/IP
○ Clipboard Text

Shortkey

Original Shortkey: rndm

● Use Global Settings (Customizable from Options | Preferences | Activations | Shortkeys)
○ Customize Settings (for this macro only)

Case Sensitivity
○ Match First Character's Case between Shortkey and Replacement Text
□ If the first letter in the text is uppercase, don't change it to lowercase
○ Require Exact Uppercase/Lowercase Match on Shortkey
● No Case Sensitivity

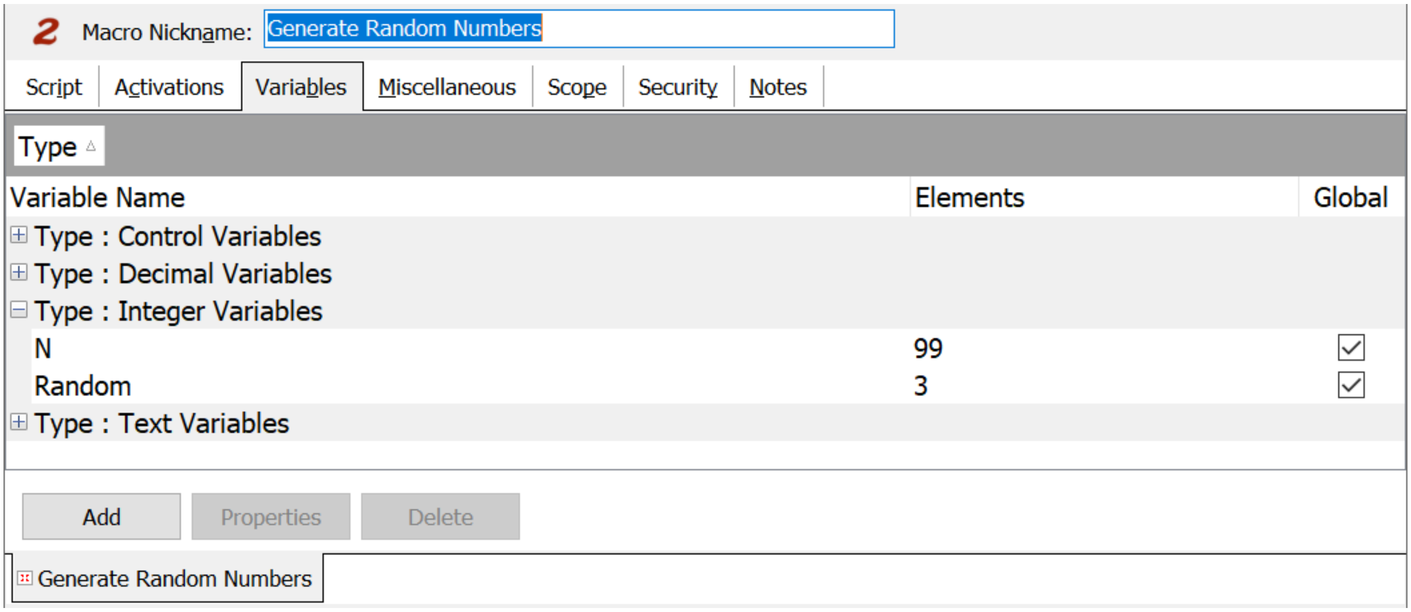
Other
□ Do not Remove Keystrokes Typed

Remove

Highlight a macro activation and click on the Remove button to remove this activation from the macro.

Variables

This tab lists all variables available for use in the macro. Use this option to create, edit or delete variables.



Macro Nickname: Generate Random Numbers

Variable Name	Elements	Global
⊞ Type : Control Variables		
⊞ Type : Decimal Variables		
⊟ Type : Integer Variables		
N	99	☑
Random	3	☑
⊞ Type : Text Variables		

Script Editor > Variables Tab

Type

The variable type is listed in this column.

Variable Name

A list of the variable names is displayed in this column.

Elements

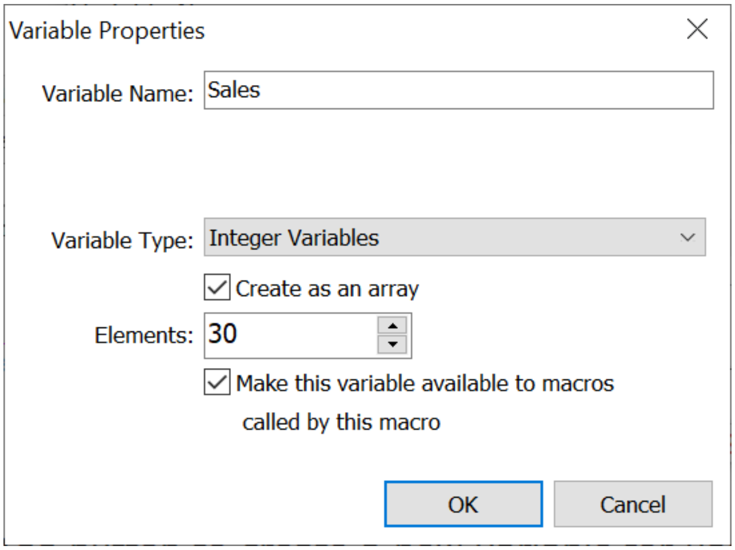
This column displays the number of array elements associated with the variables.

Global

If the Global check box is selected for a variable, the variable values are available to other macros. The values may be passed using the Macro Run command or the Variable Save and Variable Restore commands.

Add, Properties, Delete Buttons

These three buttons provide options to create a new variable, edit an existing variable or delete it.



Variable Properties

Variable Name: Sales

Variable Type: Integer Variables

☑ Create as an array

Elements: 30

☑ Make this variable available to macros called by this macro

Script Editor > Variables Tab > Add Button

Add

Click on the Add button to create a new variable for use in the macro. The Variable Properties window, as shown above, opens for generating new variables. This option does not allow for assigning values to a variable. Use this function to create the variable name and type. Use the Variable commands to assign values to variables.

Variable Name

Assign a name to the variable, preferably something that is easy to remember or that identifies the variable. In the example above the variable is named "sales". Variable names may not exceed 31 characters.

✓ **Note:** If assigning original names to variables, instead of using T[1], D[1], etc., then the following characters are not allowed in the variable name: [ ] < > , . - + = \ | / * %

Variable Type

Select from the eight different types of variables. The example above displays the Text Variable option.

Create as an Array

Select this option to set the variable as an array and then choose the number of elements to make available in the array.

Elements

Select the number of elements to place in the array. In the example above, 30 elements have been selected, allowing for the use of up to 30 variables in the array. Array elements start with [1] and not [0]. The variable format looks like the following:

```
sales[1]
sales[2]
sales[3]
...
sales[30]
```

Make this Variable Available to Macros Called by this Macro

The variable value may be passed between macros when using the Macro Run command or when using the Variable Save and Variable Restore commands.

When this option is selected the variables are also referred to as Global variables, because they are available to other macros. When not selected, the variable values are local and may only be seen in the current macro.

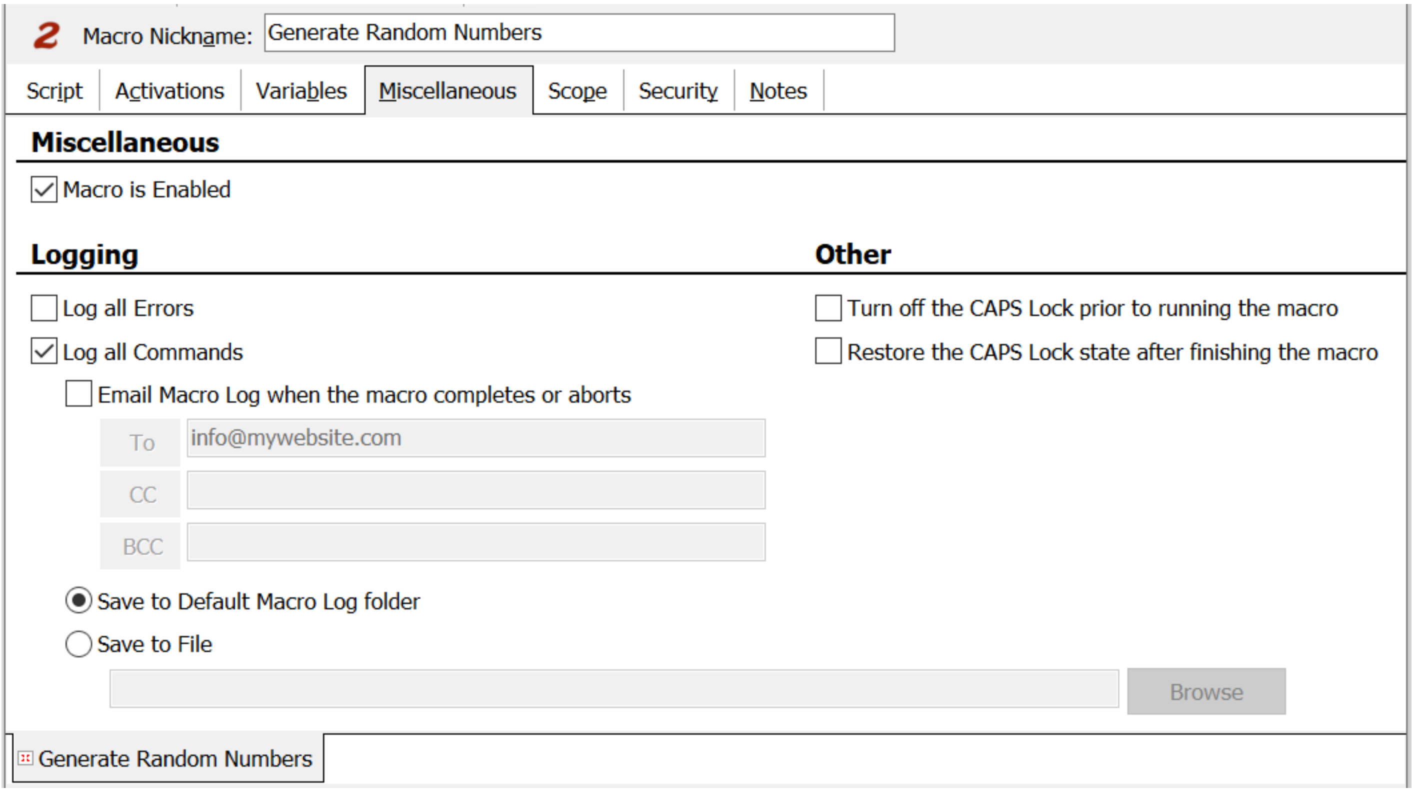
Properties

Highlight a variable that has already been created. Click on the Properties button in order to edit the variable using the Variable Properties window.

Delete

Highlight a variable that has already been created. Click on the Delete button in order to remove the variable selected.

Miscellaneous



Macro Nickname: Generate Random Numbers

Miscellaneous

☑ Macro is Enabled

Logging

□ Log all Errors

☑ Log all Commands

□ Email Macro Log when the macro completes or aborts

To: info@mywebsite.com

CC:

BCC:

● Save to Default Macro Log folder

○ Save to File

Other

□ Turn off the CAPS Lock prior to running the macro

□ Restore the CAPS Lock state after finishing the macro

Script Editor > Miscellaneous Tab

Macro is Enabled

This check box shows whether the macro has been disabled or is still an active macro. A check in the box signifies that the macro is enabled (active).

Logging

The logging option captures any error message or command logging in the macro. Decide what to log and where to store the information. The information may also be viewed by right clicking on the macro in the Macro Explorer window. Left click on Macro Properties from the menu that opens. Then click on the Logging tag to see the results.

Log All Errors

This option logs any error messages generated during playback of the macro.

Log All Commands

This option logs each command that runs successfully and date/time stamps the command.

What to do with the Log File

After determining what to log, select what to do with the file.

Email Log File when the Macro Completes or Aborts

Use this option if the file is to be sent via email. Enter a recipient email address and optional carbon copy and blind carbon copy addresses.

Save to Default Error Log

The log details are saved to the default error log. This log location is defined in Options > Preferences > File Systems > File Paths. A separate log file is created for each macro that includes logging details.

Save to File

Save the log file to the desired file name and location. Enter the file name and path or click on the Browse button to locate the file.

Turn Off the CAPS Lock Prior to Running a Macro

With this option selected, the macro turns off the CAPS Lock key before starting the macro playback.

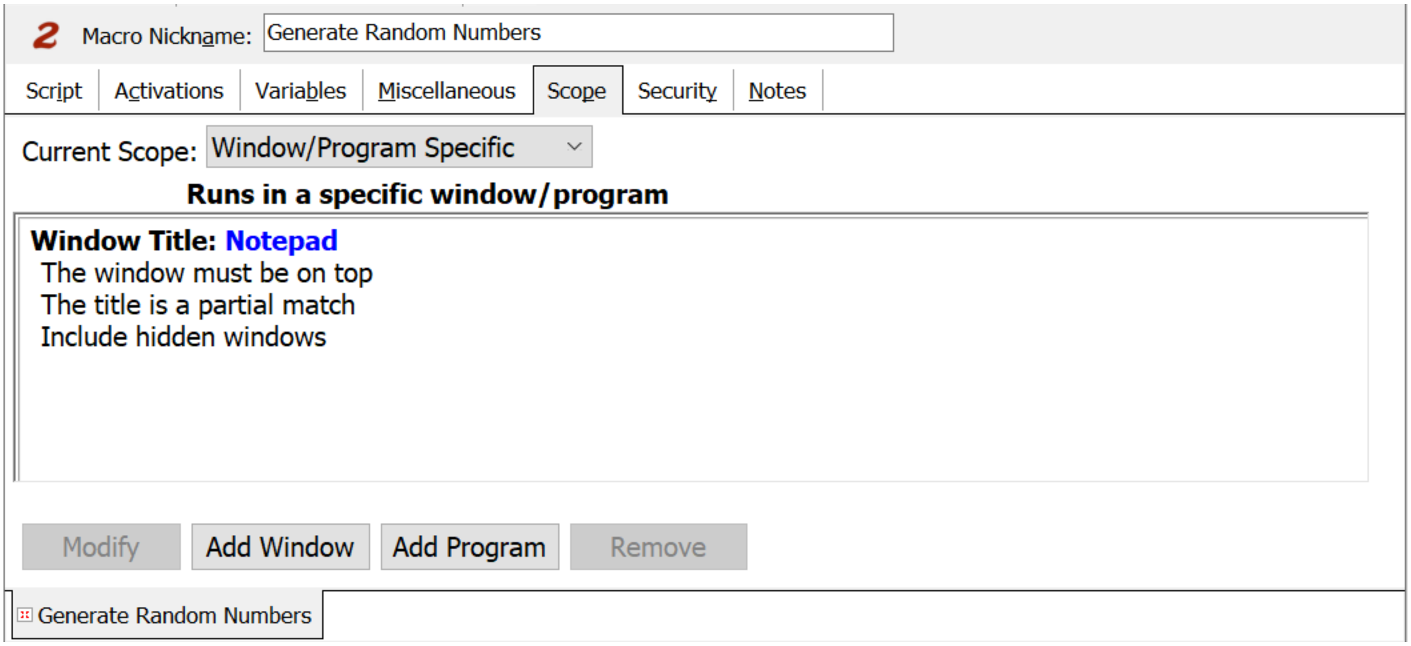
Restore the CAPS Lock State after Finishing a Macro

With this option checked, the CAPS Lock key is returned to the state it was in before the macro ran. For example, assume that the Caps Lock key is initially turned off and a macro runs that turns the Caps Lock key on during the playback. After the macro has finished playing, the Caps Lock key is turned back off again. In the previous scenario, leaving this option unchecked would keep the Caps Lock key on after the macro finished playing.

- ✓ **Note:** These two CAPS Lock options previously were global options set through the Preferences in version 3 and prior. Due to the fact that version 4 allows more than one macro to run at a time, these settings needed to be moved to the individual macro level to avoid conflicts.

Scope

Scope essentially refers to which program or programs allow the macro to activate. The default setting is Global for each new macro created. Choose to have the macro run in all applications, limit it to specific programs or windows, or have it play back in all windows and programs except for the ones specified.



Macro Nickname: Generate Random Numbers

Current Scope: Window/Program Specific

Runs in a specific window/program

Window Title: Notepad
The window must be on top
The title is a partial match
Include hidden windows

Script Editor > Scope Tab

Global

The global option allows a macro to play back in any Windows application.

Window/Program Specific

Use this option to select a specific window or program in which the macro starts. If the window or program selected is not active, the macro does not start. Click on the Add Window or Add Program button to select the window or program wanted.

- ✓ **Note:** A Window or Program Specific Scope does not prevent a macro from playing back in an application not assigned in the scope. It prevents a macro from starting unless the specified window or program is active.

For example, assume a hot key macro of F12, with a scope is set to program specific and Notepad is the selected program. Press F12 in Wordpad and the macro does not run. Press F12 in Notepad and the macro runs. The F12 macro may insert text into Notepad and then later open Wordpad during the same macro. The macro scope does not prevent this.

Global Except

The Global Except option allows the macro to play back in all Windows applications, except in the programs or windows selected. Click on the Add Window or Add Program button to select the window or program required.

- ✓ **Note:** A hot key may only be assigned to one macro if the scope is set to General. The same hot key may be used multiple times if the scope is assigned to Window or Program Specific.

Buttons

These four buttons are only available when using the Window/Program Specific or Global Except scopes.

Modify

Highlight an entry made in the scope window and click on the Modify button to change the window or program selected previously.

Add Window

Click on the Add Window button to view all windows that are currently running on the computer. One tab lists all visible windows currently running and a second tab lists all windows running that are hidden. Hidden windows are not displayed when an application is running, but are active.

If the window is currently running, scroll through the list and click on the title. The window selected is placed in the Window Title edit field. If the window is not currently running, type the window name in the Window Title edit field.

Exact Match/Partial Match

The exact match option requires that the Window title match exactly before the macro plays back. A partial match only requires a portion of the window title in order to work. For example, "notep" would be sufficient to find the "notepad" window.

The title contains wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

A Window with this Title Must Be on Top

This option plays the macro back only if the window is the current top-most window (the one that is showing on top of all other windows). If this option is not selected, the macro will start if the window is running in the background and play back in whatever window has focus.

Include Hidden Windows in the Scope

By default the macro will only look for windows that are visible, not those that run hidden and are not seen by the user. Select this option to have the scope apply to the windows listed in the Hidden tab, as well as those that are visible.

After making a selection, click OK and the window is listed in the Scope tab. Repeat the process to add additional windows if necessary.

Add Program

Click on the Add Program button to view all programs that are currently running on the computer. One tab displays the tree view of the programs in the Start Menu. The other tab displays a list of processes currently running. Scroll through the list or tree view and highlight the program wanted. The program name displays in the Executable Path edit box. If the program is not found in either tab, manually type the file name and path or click on the Browse button to find the file on the computer.

This Program Must Have a Window on Top

This option only runs the macro if the program is the current top-most program (the one that is showing on top of all other windows). If this option is not selected, the macro will start if the program is running in the background and play back in whatever program has focus.

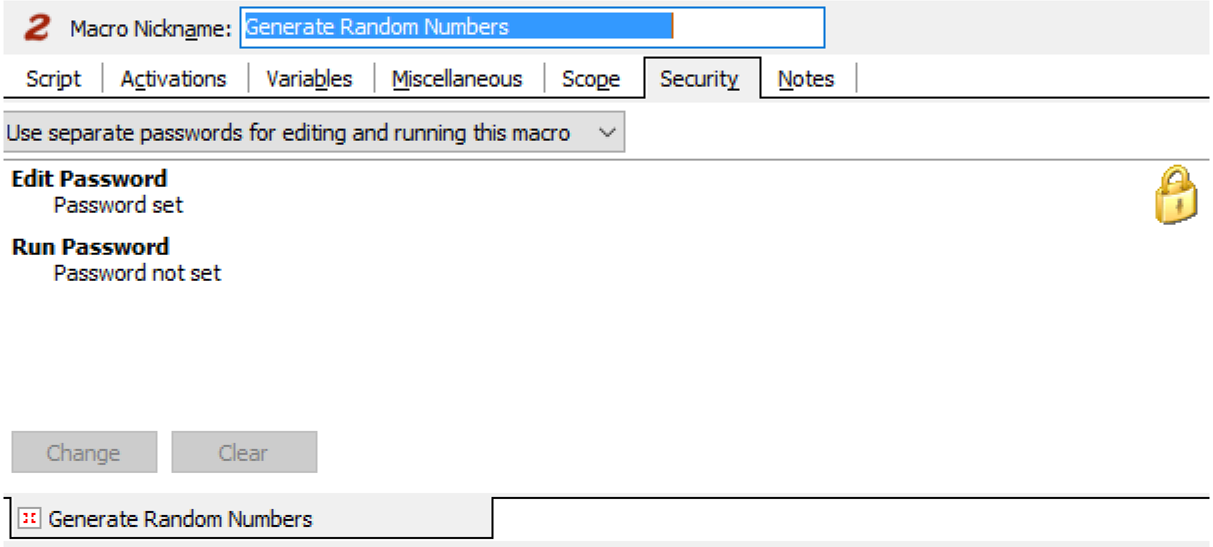
After making a selection, click OK and the program is listed in the Scope tab. Repeat the process to add additional programs if necessary.

Delete

Highlight an entry made in the scope window and click on the Delete button. The scope entry previously made is removed.

Security

Use this option to password protect a macro before it can be edited, played back or both. From the Macro Explorer, click on the Edit Macro icon to open the Script Editor. Or double click on a defined macro in the list of macros to open the Script Editor. Then click on the Security tab to edit the security settings. Three security options are available.



Macro Nickname: Generate Random Numbers

Use separate passwords for editing and running this macro

Edit Password
Password set

Run Password
Password not set

Script Editor > Security Tab

Do Not Password Protect this Macro

This is the default setting. No password is required to view, edit or run the macro.

Use a Single Password to Protect this Macro

Select this option and click on the Change button to create a password. With the password information defined, the password must be entered before viewing or editing the contents of the macro in the Script Editor. This same password is also required before running the macro.

Use Separate Passwords for Editing and Running This Macro

Select this option to create one password to edit the macro and another password to run the macro. Or use a password for either the Edit or Run functions and not use a password for the other.

Edit Password

Highlight Edit Password and click on the Change button to set up a password.

With the password information defined, the password must be entered before viewing the contents of the macro in the Script Editor. This is one way to prevent others from viewing the macros or being able to alter them.

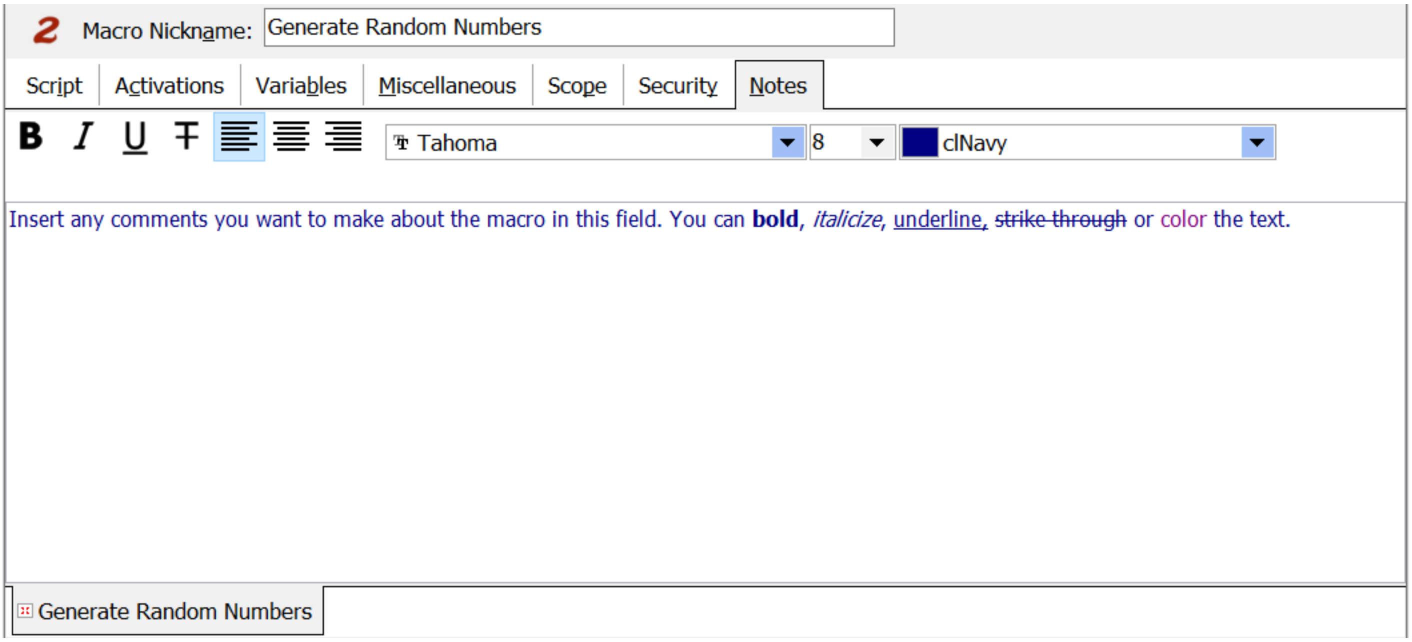
Run Password

Highlight Run Password and click on the Change button to set up a password. With this password information defined, the password must be entered before the macro runs.

✓ **Note:** Password protect the entire macro file from viewing or editing. Also prevent Macro Express from playing back any macros until a proper password has been entered.

Notes

This tab opens the Notes field. This is a free form text editor for keeping notes about the macro. Make notes about what the macro does, how it is to be used or how frequently. Essentially this field can be used to input any information about the macro.



Macro Nickname: Generate Random Numbers

Insert any comments you want to make about the macro in this field. You can **bold**, *italicize*, underline, ~~strike through~~ or color the text.

Generate Random Numbers

Script Editor > Notes Tab

The Notes field supports Rich Text editing. Optionally use colored fonts, bold, italics, underlines and strikeouts.

Editor Menu - Script

Import from Text File - (Direct Editor Only)

Use this option to import the entire text contents of a text file into the script field in the Direct Editor.

Save Macro

Save the macro data to the macro file.

Save and Close

Save the macro data to the macro file and close the Editor.

Save All

Save all macros that are currently open in the Editor.

Print Script

Print the macro script. The script prints as displayed in the Editor.

Close Current Tab

If more than one macro is loaded in the editor, this option closes the tab (macro script) that is currently being viewed.

Close

This option closes the Editor.

Editor Menu - Edit

Modify Command - (Script Editor Only)

To modify a command already inserted into the script, highlight that command and do one of the following:

- Click on Edit > Modify Command from the menu
- Press Ctrl + M
- Right click on the command and select Modify Command from the context menu
- Double click on the command

Duplicate Command - (Script Editor Only)

Use this option to make a copy of a command or string of commands in the script. This is essentially a one step method for copy and paste. Highlight one or more lines of the macro script and then do one of the following:

- Click on the Duplicate button
- Click on Edit > Duplicate Command from the menu
- Press CTRL + D

Comment - (Script Editor Only)

Insert a comment next to a line of script in the Editor. Highlight a line in the macro script and do one of the following to open the Comments edit box:

- Click on the Insert Comments button
- Click on Edit > Comment
- Press Ctrl Alt C

Type any comments about the line in the script into the Comments box and click OK. The comments are inserted to the right of the command listed in the script.

Delete Command - (Script Editor Only)

To remove a command or commands from the script, highlight one or more lines and do one of the following:

- Click on Edit > Delete Command from the menu
- Right mouse click on the highlighted lines and left click on Delete Command from the context menu.
- Press the Delete key on the keyboard

Enable/Disable - (Script Editor Only)

Rather than deleting a command from the script, sometimes it may be advisable to disable the command. A disabled line of the script does not run during macro playback. It is displayed with a red line through the text.

Disable a command

To disable a command or commands highlight one or more lines of the macro script and do one of the following:

- Click on the Disable ☑ button
- Click on Edit > Disable from the menu
- Press CTRL + N

Enable a command

To enable a disabled command or commands, highlight the disabled line or lines in the macro script and do one of the following.

- Click on the Enable ☑ button
- Click on Edit > Enable from the menu
- Press CTRL + N

Cut

Standard Windows cut command. Removes the highlighted command from the script and saves it to the Windows clipboard.

Copy

Standard copy to Windows clipboard command. This copies a highlighted command in the script and saves to the clipboard.

Paste

Standard Paste from Windows clipboard. This pastes a command that was copied to the clipboard into the script.

Undo

This option removes the last change made to the script and restores the script to the way it was before the change was made.

Copy Command Text - (Script Editor Only)

Highlight one or more lines of the macro script. Then click on Edit > Copy Command Text. This option copies the lines of script to the Windows clipboard. Paste the script into another application and the contents are displayed shown in the Script Editor.

Typically the contents display in a macro format similar to the Direct Editor script when copying the script to the clipboard and pasting. Use the Copy Command Text function to display the contents exactly as they are shown in the Script Editor.

Add to Snippets

Highlight one or more lines of the macro script. Click on Edit > Add to Snippets open a prompt to assign a name to the lines of code to save. Enter a name and click OK to save. The script lines are saved as a snippet which can easily be inserted into other macros.

Set Text Color - (Script Editor Only)

Highlight one or more lines of the macro script and click on Edit > Set Text Color. Select a text color and save. The text of the line or lines in the script are displayed in the selected color.

Set Background Color - (Script Editor Only)

Highlight one or more lines of the macro script and click on Edit > Set Background

Color. Select a text color and save. The background of the line or lines in the script are displayed in the selected color.

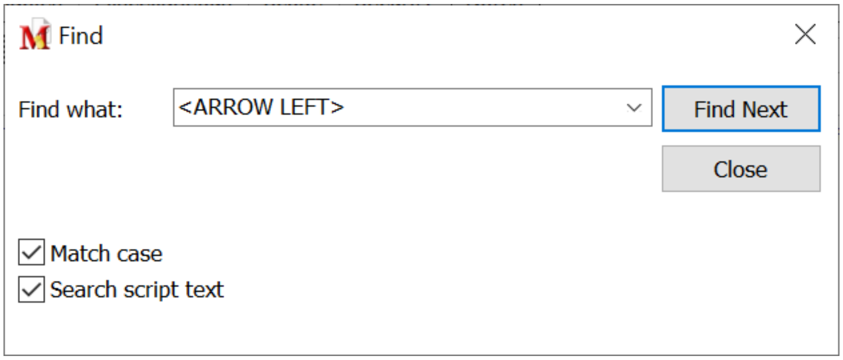
Clear Colors - (Script Editor Only)

Select this option to remove any text colors or background colors of the highlighted lines.

Find

Use the Find command to locate a key word in the text of the macro being modified. Click on Edit > Find or Ctrl F to activate the Find window as shown below. Type in the word or phrase to find and click the Find Next button or press Enter. The search scans the macro for a potential match. If one is found, the first item located is highlighted. If no match is found, a message displays indicating that no match was found.

The Find process starts from the location of the text cursor through the end of the text.



Find what: <ARROW LEFT>

☑ Match case

☑ Search script text

The drop down list contains the previous ten search items.

Find Again

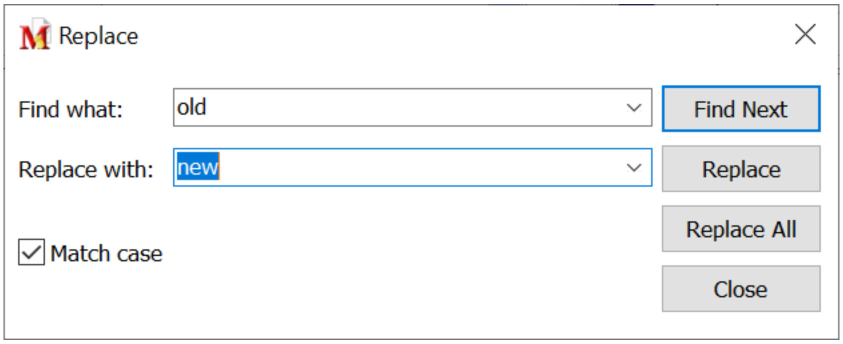
Click on Edit > Find Again or press the F3 key to look for the next item that matches the search criteria as discussed above. Continue using the Find Again feature until a message displays advising that the entire replacement text has been searched and no more matches have been found.

Replace

Use this feature to quickly replace a word found in the macro with another. Click on Edit > Replace or Ctrl R to activate the Replace window as shown below.

- Type in the word or phrase to replace in the box next to Find What.
- Then input the replacement word or phrase in the box next to Replace With.
- Click the Find Next button to begin a search. If a match is found, the word or phrase in the script is highlighted.
- Click on Replace to just replace that one instance of a match. Or click Replace All to replace all matches found in the macro script.

The Replace process starts from the location of the text cursor through the end of the text. The drop down list contains the previous ten find or replace items.



Replace

Find what: old

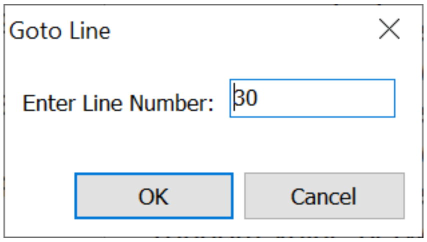
Replace with: new

☑ Match case

❖ Be advised that in addition to replacing editable text in the macro, the command structure text may also be replaced, which could possibly render the macro inoperable.

Goto Line - (Script Editor Only)

This option prompts to input the number of the line in the script to be viewed or edited. This is handy for larger macros and is quicker than scrolling through the script.



Goto Line

Enter Line Number: 30

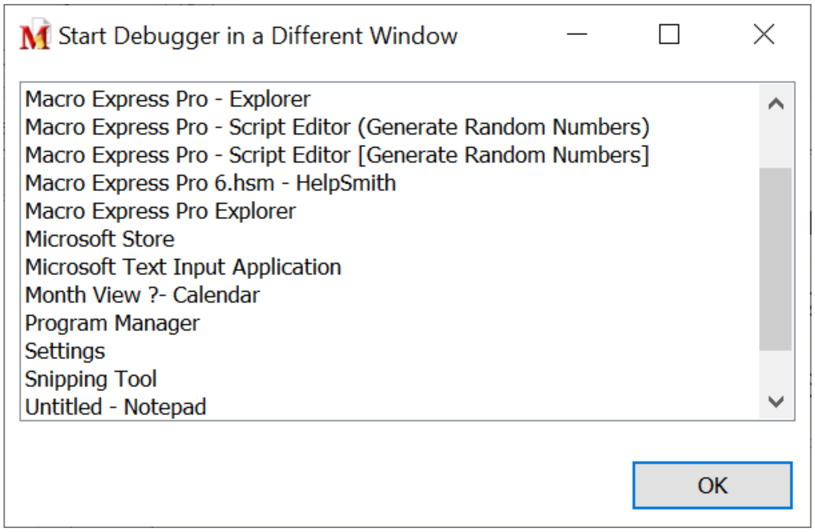
Select All

Select All highlights the entire macro contents of either the macro script in the Script Editor or macro text in the Direct Editor.

Editor Menu - Debug

Test Run Macro

Press the Test Run button on the tool bar, press F9 or click through the menu option to test run the macro open in the Editor. This is a quick way to determine if any fine tuning is necessary. Before the macro starts, a dialog similar to the one shown below is displayed. Choose whether to test run the macro in a specific window or leave focus as is.


Start Debugger in a Different Window
Macro Express Pro - Explorer
Macro Express Pro - Script Editor (Generate Random Numbers)
Macro Express Pro - Script Editor [Generate Random Numbers]
Macro Express Pro 6.hsm - HelpSmith
Macro Express Pro Explorer
Microsoft Store
Microsoft Text Input Application
Month View ?- Calendar
Program Manager
Settings
Snipping Tool
Untitled - Notepad
OK

Which window to use will depend on the first line or two of the macro.

For example if the first line of the macro types text, then select a window from the list where the text can be inserted. The macro needs to know where to type the text during the debug testing. With the (Don't Switch Focus) option the macro will attempt to insert the text into Macro Express, since this is the window that currently has focus.

If the macro does not use mouse clicks or type text before opening a program or window, then choose the (Don't Switch Focus) option. Or if the first line of the macro is to open a program or window and then insert text, use the (Don't Switch Focus) option as the macro will determine the focus.

Turning off the Switch Focus menu option, discussed below, will run the macro without opening the Start Debugger in a Different Window dialog.

Pause (Script Editor Only)

While the macro is running in test run mode click on Debug > Pause or the Pause button on the tool bar. This pauses the test run of the macro. Click on the Test Run button to resume.

Stop (Script Editor Only)

While the macro is running in test run mode, click on Debug > Stop or the Stop button on the tool bar. This stops the test run of the macro.

Toggle Breakpoint (Script Editor Only)

Highlight a line in the script to set as a breakpoint and then select the Toggle Breakpoint menu option or press CTRL B. When test running a macro, the macro stops at the set breakpoint. During the break, it's possible to view the Variables window (Debug > Show Variable Values from the menu) and verify that the variables are being populated correctly.

Press the Test Run Macro button to resume until finished or until the next breakpoint. To remove a Breakpoint, highlight the command in the script and press CTRL B or click on the menu item Debug > Toggle Breakpoint.

- ✓ **Note:** The breakpoints are saved when the macro is saved and are available for use the next time the macro is run through the debugger. The breakpoints are ignored when running the macro outside of the debug mode.

Step Options (Script Editor Only)

Select the Step Over or Step Into option to test run the macro and have the macro pause after each line in the script. Pressing the F8 or F7 button causes the macro to run the next command and so on, until the macro has finished. This simplifies the process for debugging a macro.

Step Into - F7

If the script contains a Macro Run command, pressing the F7 key or clicking on the Step Into button loads the script of the called macro into the editor and steps through each line of the called macro script.

Step Over - F8

If the script contains a Macro Run command, pressing the F8 key or clicking on the Step Over button plays the called macro at the appropriate point in the script, but does not step through the called macro.

Switch Focus (Script Editor Only)

When using the F7 or F8 key to step through a macro, it is necessary to click the mouse to return focus to the Editor between each step. In order to avoid this, select the Switch Focus option. The focus returns to the prior window when stepping through a macro.

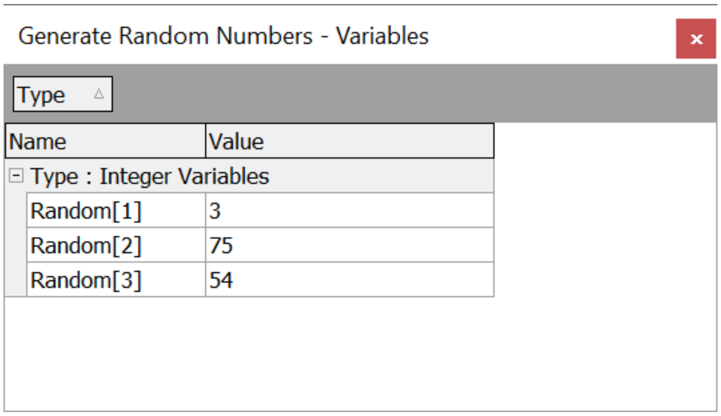
Save Before Test Run

This option saves the macro script first and then performs a test run of the macro.

Show Variable Values

The values of the variables are displayed in a separate window when debugging a macro. Use this window to determine that the correct values are being saved to the variables. The window displays the variable type, variable name and the corresponding value.

Click on Debug > Show Variable Values or press the F5 key to open a window similar to the one displayed below. Give focus back to the Script Editor, press F7, F8, F9 or click on the Test Run button to run the macro in a test mode and watch the window populate with the variable values. Running in test mode again clears the values and repopulates the window with new data as the macro runs.


Generate Random Numbers - Variables

Name	Value
⊟ Type : Integer Variables	
Random[1]	3
Random[2]	75
Random[3]	54

Helps for Debugging Macros

1. Did the macro run or not?

Depending on the macro, it may be difficult to determine whether the macro started and just played back incorrectly or never started at all. One method to help determine this would be to place a Sound File command at the beginning of the macro. The sound alerts when the macro starts.

Or log the commands and check the log file later to see if the macro at least started. This would be beneficial for debugging Scheduled macros, where you may not be around when the macro is set to run.

2. Variables

If you are seeing odd values being used in the macros, you may need to check what values are being saved to variables. One method to do this would be to use the Text Box Display command to show what values are saved to each variable.

If a variable value is not correct, you can look to see if the wrong field is being copied and saved to a variable, not enough time was allowed to copy the value or any number of other causes.

3. Timing

Timing may cause a macro to fail or to do unexpected things. Macro Express has quite a few options to help. Below are several of the more common timing commands.

a) Wait for Window Title instructs the macro to wait for the window to be ready to accept mouse or keyboard input before continuing.

b) Delay adds a delay of a specified duration to allow Windows or the application to be ready before accepting additional macro commands.

c) Wait for Text Playback instructs the macro to not continue until all of the text from a Text Type command has been typed. This is useful for longer strings of text.

d) Keystroke Speed slows down the playback of text in situations where the receiving application has difficulty high speed input.

e) When capturing (recording) a macro, the Macro Speed command is inserted in the first line of the macro script. Adjust the speed value to slow down or speed up the macro playback.

Editor Menu Options

The three insert commands are the equivalents of the buttons found in the Script Editor. These options determine where the command is positioned in the macro script. Choose from one of the three options to be the program default.

Insert Command Before Selected - (Script Editor Only)

With this option selected, any commands added to the script are inserted above the highlighted line in the script.

Insert Command After Selected - (Script Editor Only)

With this option selected, any commands added to the script are inserted to the line below the highlighted line in the script.

Add Command to End of Script - (Script Editor Only)

With this option selected, any commands added to the script are inserted as the last line of the macro script.

Example
Assume the macro script contains 10 lines and Insert Command Before Selected is assigned as the default. A new Text Type command needs to be inserted between lines 5 and 6. To do this, first highlight line number 6 in the script. Then expand the Keyboard category and double click on the Text Type command. Insert the text and click OK to save. The Text Type command is inserted before line 6 in the macro script.

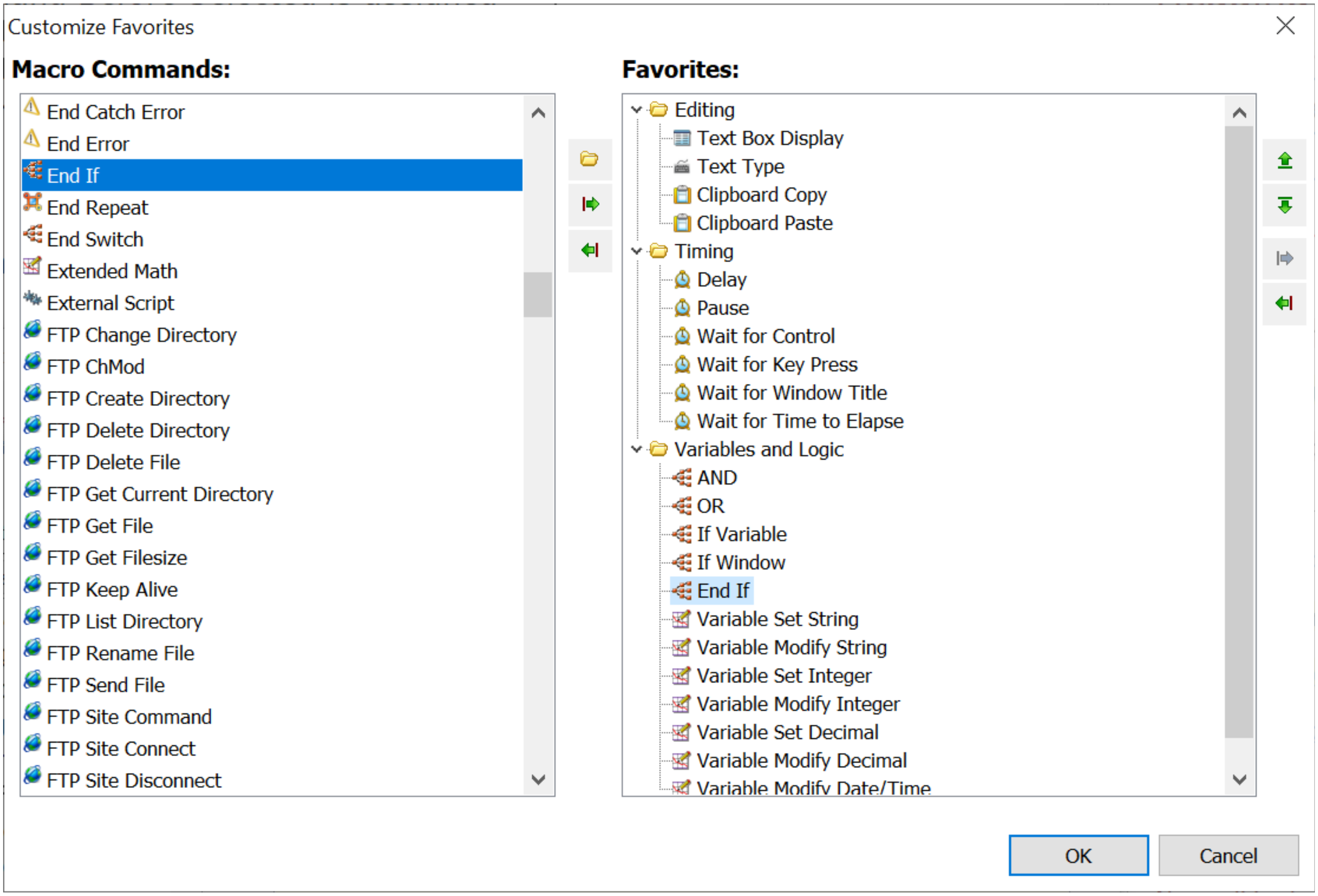
Now, suppose the command needs to be inserted at the end of the script. The alternate method would be to expand the Keyboard category and highlight the Text Type command. Click on the Add Command to End of Script button to open the Text Type command. Insert the text and save. The button overrides the default and places the Text Type command at the end of the script.

Customize Favorites

The Script Editor allows for the display of all commands, code snippets or a list of the most frequently used macro commands for quicker macro creation.

Use this Customize Favorites option to create a favorites list. Select the command and click on the insert button to move it to the Favorites column. Click on the Folder icon to insert a folder in the Favorites field. The directional buttons on the right of the window allow for positioning the command in the Favorites list. When finished, click on the OK button to save.

Click on the drop down menu in the Script Editor to display the Favorite Commands.



Customize Favorites

Macro Commands:

- End Catch Error
- End Error
- End If
- End Repeat
- End Switch
- Extended Math
- External Script
- FTP Change Directory
- FTP ChMod
- FTP Create Directory
- FTP Delete Directory
- FTP Delete File
- FTP Get Current Directory
- FTP Get File
- FTP Get Filesize
- FTP Keep Alive
- FTP List Directory
- FTP Rename File
- FTP Send File
- FTP Site Command
- FTP Site Connect
- FTP Site Disconnect

Favorites:

- Editing
  - Text Box Display
  - Text Type
  - Clipboard Copy
  - Clipboard Paste
- Timing
  - Delay
  - Pause
  - Wait for Control
  - Wait for Key Press
  - Wait for Window Title
  - Wait for Time to Elapse
- Variables and Logic
  - AND
  - OR
  - If Variable
  - If Window
  - End If
  - Variable Set String
  - Variable Modify String
  - Variable Set Integer
  - Variable Modify Integer
  - Variable Set Decimal
  - Variable Modify Decimal
  - Variable Modify Date/Time

Script Editor > Options > Customize Favorites

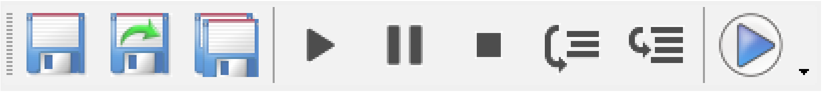
Editor Menu - View

Direct Editor

From the Script Editor click on View > Direct Editor to open the Direct Editor. Or click on the Toggle Direct Editor button ▤ to change Editors.

Toolbar

Choose whether to display or hide the Toolbar containing the icons shown below.



Line Numbers

Select this option to display line numbers next to each command in the script. The line numbers make it easier to find specific lines in longer scripts.

Word Wrap

The word wrap selection displays several lines of macro information in the Script Editor (if more than one line is required).

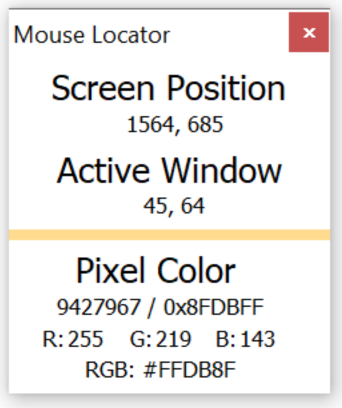
Editor Menu - Tools

Launch Mouse Locator

The Mouse Locator tool displays the X and Y coordinates of the mouse pointer relative to the entire screen or to the window that is active. This is useful when wanting to include mouse clicks in macro scripts.

The locator displays the exact position of the mouse in order to plot those coordinates into a macro. Just plug in the coordinates when using the Mouse Commands found in the Scripting Editor or Direct Editor. (This function is not necessary if using the Capture option.)

Also activate this window by clicking on the Launch Mouse Locator button provided in the Mouse Move command or the Get Pixel Color command.


Mouse Locator

Screen Position
1564, 685

Active Window
45, 64

Pixel Color
9427967 / 0x8FDBFF
R: 255 G: 219 B: 143
RGB: #FFDB8F

Macro Express Script or Direct Editor > Tools > Launch Mouse Locator

If the Mouse Locator window pops up in an area of the screen that covers important data, click on the window, hold the left mouse button down on the title bar at the top of the window and drag to another area of your screen.

Screen Position

The set of numbers directly under Screen Position lists the position of the mouse relative to the computer screen. When the mouse is moved, the numbers change in relation to the mouse cursor position. The number to the left is the X coordinate, or how many pixels from the left edge of the screen. The number on the right is the Y coordinate, or how many pixels the mouse is positioned from the top of the screen.

Active Window

The set of numbers directly under Active Window lists the position of the mouse relative to the active window or the window that has focus. When the Mouse Locator window is first launched, it receives focus. Clicking on a window outside of the Mouse Locator gives focus to that window. The Mouse Locator window remains on top in order to easily read the coordinates for the window with focus.

When the mouse is moved, the numbers change in relation to the mouse position. The number to the left is the X coordinate, or how many pixels the mouse is positioned from the left edge of the active window. The number on the right is the Y coordinate, or how many pixels the mouse is positioned from the top of the active window.

Pixel Color

This option displays the color of the pixel to which the mouse is pointing. Windows assigns a numeric value to each color, normally from 6 to 8 digits in length. The left side is the decimal value. This is the value returned by the Get Pixel Color command. Use the decimal value in macros that compare color values. The right side displays the hexadecimal value in BGR format.

The middle row displays the pixel color in separate values for Red, Green and Blue. These values are used when entering colors in programs, such as word, that accept separate values for Red, Green and Blue.

The bottom row displays the hexadecimal pixel color value in RGB format.

Which Pixel Color formats are displayed can be adjusted using the Pixel Color options found in Options, Preferences, Appearance, Mouse Locator.

How colors are displayed

To display colors, computers use the RGB Additive Color System. The letters R, G and B represent Red, Green and Blue respectively. Separate values are used for the amount of Red, Green and Blue. On a Windows computer, each color ranges from 0, all the way off, up to 255 (or FF hexadecimal) all the way on. Combining different values for each basic color allows millions of different colors to be displayed. For example, White has a value of 000000, Green has a value of 00FF00 and black has a value of FFFFFF.

The hexadecimal number that represents a color can also be represented as an integer value. For example, one orange color has a hexadecimal value of 2D42D4. When displayed as an integer this value is 2966228. The Get Pixel Color command returns the integer value of the color for a given pixel. This allows the integer variable commands to test for specific colors in your macros.

The Mouse Locator displays colors in a variety of formats to make it convenient to enter colors into other programs. For example, when entering a color for a web page the color value is in hexadecimal RGB format. To enter a color in Word requires separate integer values for Red, Green and Blue. And some programming languages require a hexadecimal BGR (Blue, Green, Red) format.

Copy to Clipboard

You can use the Mouse Locator to get the value of a color on your computer monitor. Make sure the Mouse Locator is the active window, then move your mouse over the color you want and press Ctrl-c. The color will be copied to the clipboard so you can paste it into another program.

By default, several values are copied to the clipboard including the Screen Position of the mouse and the Pixel Color in a variety of formats. Here is an example:

```
990,385,2966228,2D42D4,212,66,45,#D4422D
```

What is copied to the clipboard can be adjusted using the Copy To Clipboard options found in Options, Preferences, Appearance, Mouse Locator.

Direct Editor

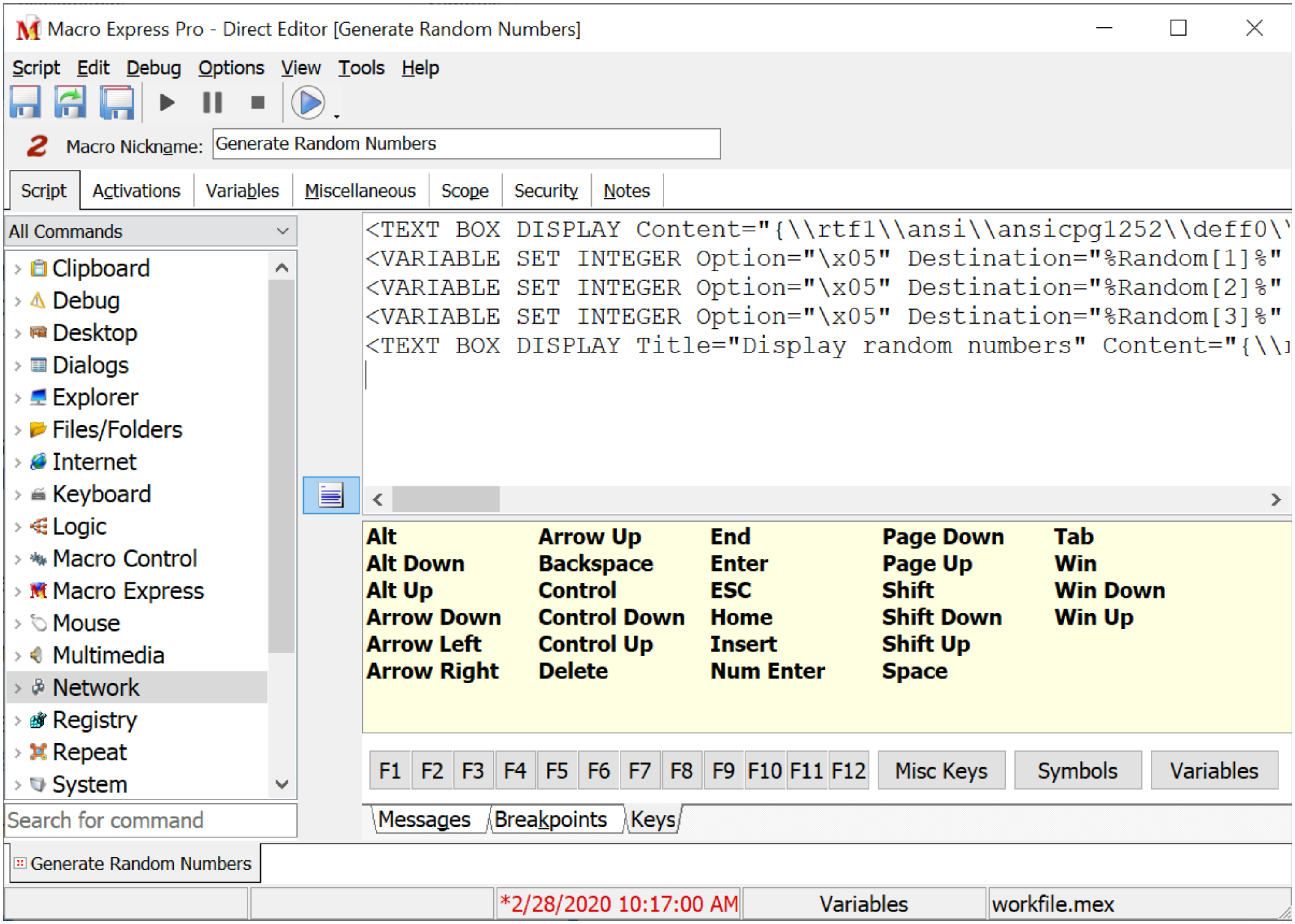
In the Script Editor, macro commands are displayed in an easily recognized and understood format. To achieve this simplicity, some of the details about each command are only displayed in the edit dialog for the macro command. In the Direct Editor, all details about each macro command are visible. When describing these two ways of viewing macros, we refer to them as the Script and ASCII representations.

To view the ASCII representation of a macro click View > Direct Editor from the Script Editor menu. Note that when in the Direct Editor view, all open macros are displayed in their ASCII form. You can use either the Script Editor or the Direct Editor to build or edit a macro.

The Direct Editor is divided into three main components, the Commands, the Macro Text and Edit Buttons/Debugging tabs. Select from the commands what to insert into the Macro Text. The Macro Text reads from left to right so that the top line in the text, starting at the left, is the first command that plays back in the macro.

The Script tab contains the list of commands and the macro script. Additional information about the macro can be viewed or edited in the Activations, Variables, Miscellaneous, Scope, Security and Notes tabs.

- ✓ **Note:** More than one macro may be open in the editor at the same time. Tabs at the bottom of the window allow for navigation between open macro scripts.



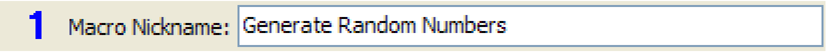
Macro Nickname: Generate Random Numbers

```
<TEXT BOX DISPLAY Content="{\\rtf1\\ansi\\ansicpg1252\\deff0\
<VARIABLE SET INTEGER Option="\x05" Destination="%Random[1]%"
<VARIABLE SET INTEGER Option="\x05" Destination="%Random[2]%"
<VARIABLE SET INTEGER Option="\x05" Destination="%Random[3]%"
<TEXT BOX DISPLAY Title="Display random numbers" Content="{\\r
```

Script Editor > View > Direct Editor

Macro Icon and Nickname

The Macro Nickname field and a macro icon are displayed near the top of the Direct Editor window. In the example below the icon is a blue number one.



Macro Nickname: Generate Random Numbers

Macro Icon

The icon (in the example above, a compass) is displayed to the left of the macro nickname. Click on the icon to open the Select an Icon window. Click on the browse button to open a folder with a number of icon options. Or navigate to another location on the hard drive to find other icons on the computer. If an icon was not selected at the time the macro was created, the default Macro Express icon is displayed.

Macro Nickname

The nickname is used to help remember the function of the macro. If a nickname was assigned when creating the macro, the nickname is displayed as shown in the image above. The nickname may be changed in this field if desired.

Command View

Select from the drop down menu which command view to display.

All Commands

This option displays all of the commands that may be used in Macro Express. To insert a command into the script, scroll until finding the command category wanted. Click on the category to expand the list and display all of the commands available for that category.

Double click on the command to display a window to define what the command should do. Then click OK on the new window to insert this command into the Macro Script.

Double clicking on some of the commands that do not require any definition inserts the command directly into the Macro Script without an intermediary window.

Favorite Commands

This option displays only the commands that are assigned as favorites, making it easier to find the most frequently used commands. From the Script Editor menu click on Options > Customize Favorites to assign the favorites.

Code Snippets

Code snippets are pieces of macro script that may be reusable in other macros. Rather than rewriting the section of macro each time, save the lines of the macro as a code snippet and retrieve at any time as needed.

✓ **Note:** Code Snippets may only be created and saved in the Script Editor view.

Inserting a Snippet into the Script

Highlight the Code Snippet to use and drag it onto the Direct Editor script.

Search For Command

Use this function to help locate a specific command within the Commands list. To look for the Password Protection command, enter "pass" without the quotes. This highlights the Password Protection command within the list of commands. Notice that entering the "p" and the "a" first displays the Pause command. Adding the "ss" moves the highlight to the Password Protection command.

Macro Text

This area lists each of the commands that the macro performs. The macro plays back sequentially from top to bottom. Each command in the macro is typically displayed on a separate line. But it is possible to insert more than one command on a line. To insert a command into the text, place the mouse cursor where the command should be positioned. Then select the command and it is inserted to the right of the mouse cursor in the Macro Text field.

- ✓ **Note:** It is not necessary to use the Text Type command in the Direct Editor. Just type the text to be played back directly into the edit field. Macro Express converts the text to a Text Type command when the macro is saved and the Editor is reopened.

Debugging Tabs

These tabs provide information when trying to debug a macro. Also click on Debug > View Variable Values or press the F5 key to see a display of all variables used in the macro and the values saved to those variables. These are displayed when running the macro in Test Mode.

Messages

When saving a macro, any syntax errors or other problems with the macro are displayed in the Messages tab.

Breakpoints

This tab displays any breakpoints defined in the macro for debugging purposes. The line number of the Break and the command are displayed.

Keys Tab

Use the Keys tab to insert symbols or other keystrokes directly into the script.

Special Keys and the F1 - F12 Keys

Click on any of the entries (keys) in the yellow box area. For example, clicking on the Alt key places the Alt command into the Macro Text field. The Alt key plays back in the macro at that point in the script. Or Tab to the entry and press

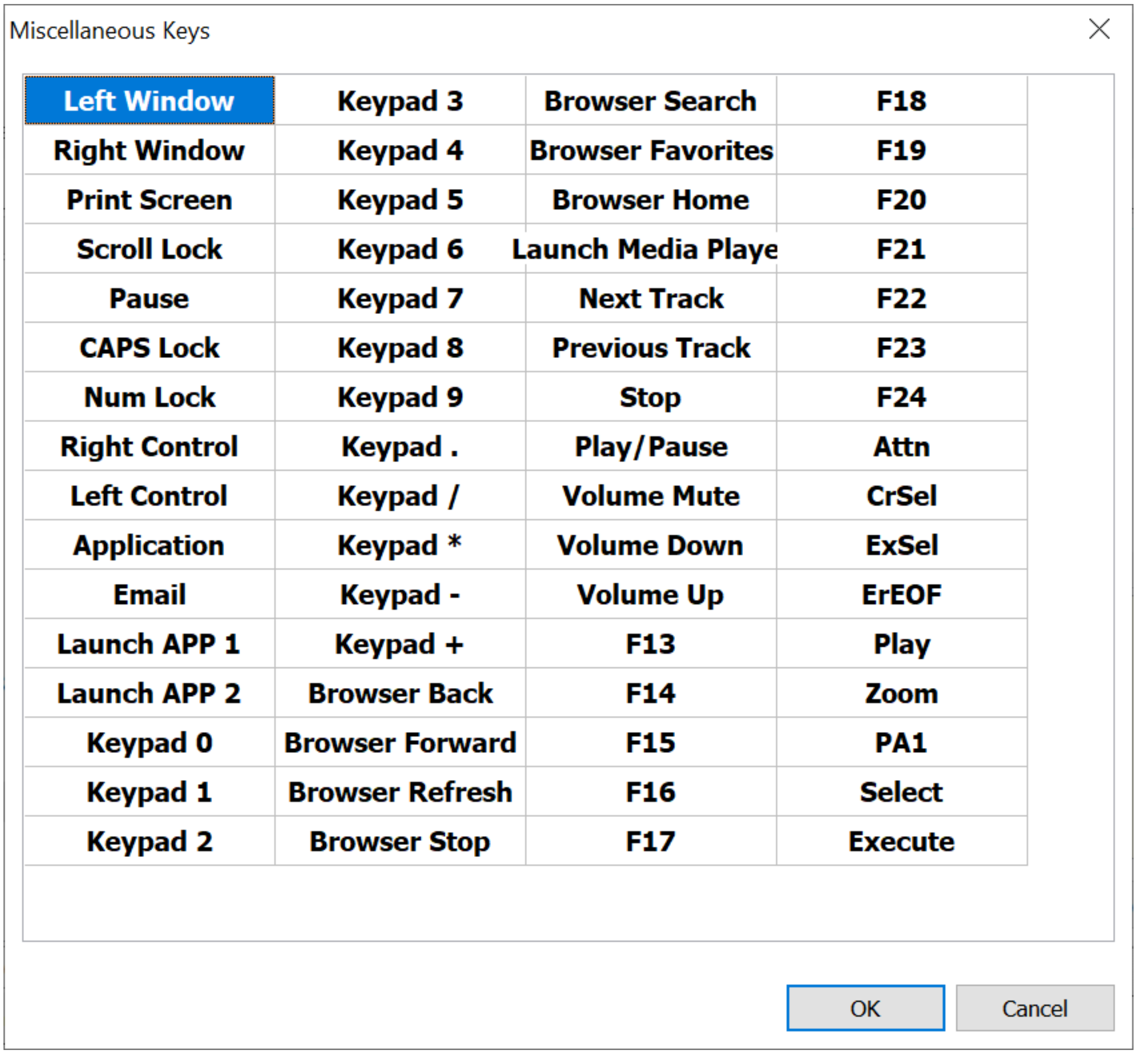
Enter. Click on the F1 - F12 keys to input these keystrokes in the Macro Text field and have them play back in the macro.

Symbols

To insert a symbol into the Macro Text click on the Symbols button to display a window with a number of symbols. Double click on the symbol wanted and it is inserted into the Macro Text. Or highlight the symbol and click on the OK button. Also use this function to insert Alt Key Sequences. See the Text Type topic for more information.

Misc Keys

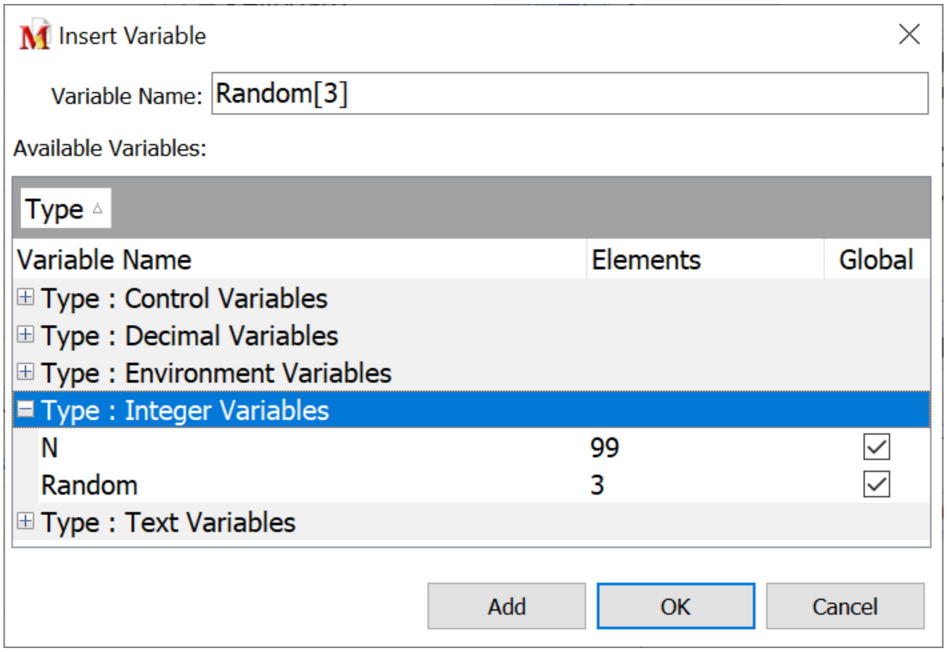
Miscellaneous keystrokes such as the Print Screen, Right Window Key, etc. may be inserted into the script. Click on the "Misc Keys" button to open the dialog as shown below. Select the key to insert and click on the OK button. The Miscellaneous Key is placed into the Macro Text at the location where the text cursor is positioned.



Left Window	Keypad 3	Browser Search	F18
Right Window	Keypad 4	Browser Favorites	F19
Print Screen	Keypad 5	Browser Home	F20
Scroll Lock	Keypad 6	Launch Media Playe	F21
Pause	Keypad 7	Next Track	F22
CAPS Lock	Keypad 8	Previous Track	F23
Num Lock	Keypad 9	Stop	F24
Right Control	Keypad .	Play/Pause	Attn
Left Control	Keypad /	Volume Mute	CrSel
Application	Keypad *	Volume Down	ExSel
Email	Keypad -	Volume Up	ErEOF
Launch APP 1	Keypad +	F13	Play
Launch APP 2	Browser Back	F14	Zoom
Keypad 0	Browser Forward	F15	PA1
Keypad 1	Browser Refresh	F16	Select
Keypad 2	Browser Stop	F17	Execute

Variables

Click on the Variables button to open the Insert Variables window and choose a variable to play back in a macro. Select the variable from the variable types available and click on the OK button. This places the variable into the Macro Text at the location where the cursor is positioned. Or click on the Add button to name a new variable to use in the macro.



Insert Variable

Variable Name: Random[3]

Available Variables:

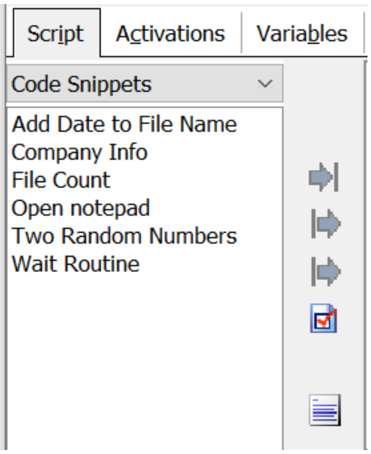
Type

Variable Name	Elements	Global
⊞ Type : Control Variables		
⊞ Type : Decimal Variables		
⊞ Type : Environment Variables		
⊟ Type : Integer Variables		
N	99	☑
Random	3	☑
⊞ Type : Text Variables		

Direct Editor > Variables Button

Code Snippets

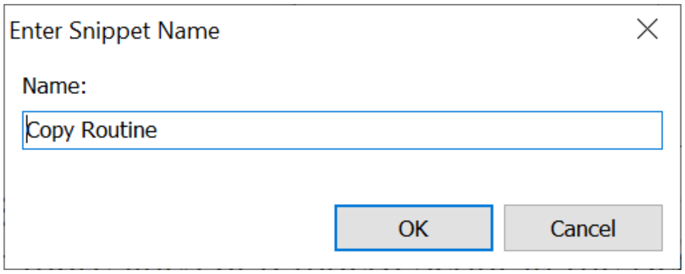
Use the Code Snippet function to create reusable macro code. Create common routines or functions that may be easily inserted into other macro scripts. Save the code snippet to later insert into a macro at any time. Select Code Snippets from the drop down list as shown below to view, create or insert snippets.



Add Date to File Name
Company Info
File Count
Open notepad
Two Random Numbers
Wait Routine

Creating a Snippet

Select (highlight) one or more lines of a macro script in the Script Editor. Right mouse click on the highlighted lines and left click on Add to Snippets from the context menu. (Or click on Edit > Add to Snippets from the Script Editor menu.)



Name: Copy Routine

A small dialog box opens prompting to name the snippet. The name helps to later identify what the snippet does. Click OK to save the name and add it to the list of existing snippets.

Inserting a Snippet into the Script

Highlight the code snippet needed and click on one of the insert buttons to insert the snippet into the script in the location required.

Deleting a Snippet

If a code snippet is no longer needed, highlight the snippet and press the Delete button on the keyboard. A prompt appears to verify that the snippet should be deleted. Press Yes to delete or No to cancel.

Editing a Snippet

There is not a direct method for editing an existing code snippet. The current method consists of the following four steps.

1. Insert the snippet into a macro script.
2. Edit the changes directly in the macro script.
3. Delete the snippet from the list.
4. Save the script changes as a new snippet with the same name as the one deleted.

Capture (Record) Macros

Capture Settings

Macro Express can capture keyboard and mouse activity for later playback as a macro. Options are available to determine what to capture and how much. Optionally capture keystrokes, mouse movements and clicks and actual time between actions.

There are two methods for starting a macro recording. One is to select the activation (hot key, shortkey, etc.) before recording. The other is to define the macro activation after recording.

Press System Hot Key to Start Capture

One of the System Macros is defined as CTRL+ALT+SHIFT+X by default. This is the Start/Stop Capture system hot key. Pressing CTRL+ALT+SHIFT+X starts the recording of a macro and pressing it a second time stops the recording. The default key may be changed to another combination available in Macro Express, such as ALT+F1.

While recording, a Camera icon displays in the Windows notification area to indicate that a capture is taking place. Besides pressing the system hot key to stop the recording, right click on the Camera icon to stop the capture. Open Options > Preferences > General > Notification Area and turn off the Show Recording Status option to not display the Camera icon.

When the recording stops, a Capture Complete window appears. Assign the captured macro to a Hot Key, Shortkey or other activation.

Before recording a macro using the System Hot Key method, make sure that the default settings have been defined. Select Options > Preferences > General > Capture from the Macro Explorer to define the capture settings. These are the settings used when utilizing the System Start/Stop Capture macro.

Click on Capture Icon to Start Capture

This option requires defining the macro activation and other information before recording the keystrokes and mouse moves. Upon clicking on the capture icon in the Macro Explorer window, the Create New Macro Window opens. Select an activation and optionally assign a nickname to the macro. Click OK to open the capture settings window to determine what to capture, etc. The options are listed below.

Scope

The Global scope allows for the playback of the macro into any Windows program. The Program scope indicates that the recorded macro will only play back in the program it is recorded in. Only use the "program" option if recording all of the macro in the same program. The macro playback will have unpredictable results, if recording in several programs and the Program scope is selected..

Capture the Keystrokes

Select this option to record all keystrokes made during the capture process. If this option is not selected, then only mouse moves and clicks are recorded.

When the Alt Key is Pressed, Capture It Twice

Use of the <ALT> key does not always work properly for activating menu commands. (Example: "<ALT>f" for activating the file menu command). To ensure that a menu command will always work, a double <ALT> is used. (Example: "<ALT><ALT>f") Selecting this option captures a double <ALT> whenever the ALT key is pressed. Use this option if planning to capture keystroke commands that activate window menus.

Capture the Mouse

Choose this option to record all mouse clicks and movements made during the capture process. If not selected, only keystrokes are recorded. There are several mouse options to choose from.

Relative to Screen

This option captures all mouse clicks and movements in relation to the cursor's position on the monitor.

Relative to Window

This option captures all mouse clicks and movements in relation to the application window being used.

Capture and Restore the Window Size

This option is only available if the Relative to Window option is selected. It remembers the size of the window when the capture occurred. When the macro plays back, the window is resized, if necessary, to the original size when the recording was made. This helps ensure that the mouse clicks and movements are accurate.

Relative to Position

This option captures the mouse clicks or movements based on the last position of the mouse cursor.

Only Capture the Mouse Clicks

Use this option to only capture the mouse clicks and the position of the mouse click. This option captures the position where the mouse is to click and also the mouse click itself. No other movements of the mouse are recorded.

Unless drawing or needing to record the dragging of an item, select the Only Capture the Mouse Clicks option. This option reduces the size of the macro considerably and makes it much easier to edit, if needed. It also increases the playback speed of the macro, since all of the interim mouse move commands and associated delays from the macro are eliminated.

Capture the Macro Speed At

This option actually refers to how fast the macro plays back in relation to the recording speed.

Speed	Description
Actual Speed	Plays the macro back at the same speed as the macro was recorded. Any hesitations, etc. play back at the same speed as recorded.
Half Speed	Slows down the macro playback to half of the recording speed.
Double Speed	Speeds up the macro playback to twice the actual recording speed.
Quadruple Speed	Speeds up the macro playback to four times the actual recording speed.
Fastest Speed (no delays)	Plays back the macro with no hesitations, delays, etc. Remove the check next to the Capture at a Specific Speed box to eliminate delays from the macro. **Caution:** Selecting this option may cause the macro to play back too fast and throw off the timing of the macro.

- ✓ **Note:** After the capture is complete, it is possible to modify the playback speed of the macro in the Script Editor. Use the Macro Playback Speed command to do this.

Display a Prompt before Capturing (Capture preference option)

After pressing the system macro to start a capture, this option displays a message indicating that the capture is about to start.

Begin Capture in Window (available when clicking on the Capture icon)

Select the program from the drop down menu in which the macro recording will start. The drop down menu lists the windows currently open on the computer. Or select the last active window option. This refers to the window that was open prior to opening Macro Express Explorer to start the capture.

Press the Capture button and the program selected is brought to the foreground so capturing may begin.

Capture a Temporary Macro

There is a quick way to capture a temporary macro for testing purposes. No assigning of hotkeys or checking settings is required. The macro is saved temporarily for playback. Open the application in which to record the macro and press the system macro hot key combination of Ctrl+Alt+Shift+T to start the capture. Next enter the keystrokes, mouse clicks and movements. When finished, press the system macro Ctrl+Alt+Shift+X to stop the capture.

The macro is not saved as a viewable macro in the Macro Express Pro - Explorer and cannot be edited in the Editor windows. The macro is stored in a file. Once another macro is captured using Ctrl+Alt+Shift+T, the previously captured macro is overwritten. The location of the file is displayed in Options > Preferences > File System - File Paths tab. Use the File Paths preference option to change the location of this file.

The capture of a temporary macro uses the capture settings defined in Options > Preferences > General - Capture tab.

Run the Captured Temporary Macro

Press the system macro hot key combination of Ctrl+Alt+Shift+S to start the play back of the temporary macro.

Multimedia Wizards

Below is a listing of each of the Multimedia wizards available.

Play Audio CD-ROM

Play an Audio CD.

- Play favorite music CDs
- Pause the CD to answer the phone
- Repeat a favorite song (go to previous, next or specific track)
- Open and close CD Tray

Play a Video Clip

Play a Video Clip.

- Display and watch a favorite video clip

Sounds and Volume

Play a sound or set audio volume, treble and bass controls.

- Play an audio clip to alert of an event
- Adjust volume down or up depending on music being played
- Mute the volume when on the phone
- Adjust treble and bass volume depending on music being played

Macro Express - Explorer > Quick Wizards Button > Uses Multimedia

Networking Wizards

Below is a listing of each of the Networking wizards available.

Send an Email

Send Email and attachments to one or more recipients.

- Send a set of instructions to a user. Prompt for user.
- Send a message to a specific set of friends or associates. Prompt for message.
- Send favorite pictures or other attachments
- Send data captured in the clipboard

Go to a Web Page

Go to a web site or initiate an ftp download.

- Go to a favorite web site
- Download the latest version of a software program

Work with a Network Drive

Connect, Disconnect or Toggle Network drives.

- Access the network server or another computer
- Make network connections temporary
- Bypass Network Neighborhood

Dial-Up Networking

Connect and Disconnect to the Internet and other computers using Dial-Up Networking.

- Quick one keystroke access to dialing up the Internet and hanging up the phone

Macro Express - Explorer > Quick Wizards Button > Networking

System and Files Wizard

Below is a listing of each of the System and File wizards available.

Launch or Activate a Program

Launch a program or a document. Or switch to a program that is already running.

- Switch to an E-Mail program if it is running; otherwise, launch and run it.
- Open a Word Document
- Open a spreadsheet document
- Launch a database program

Reboot or Shut Down

Logoff, Reboot or Shut Down the computer.

- Bypass explorer, task manager, etc.
- Shut down and power off with a single command

Copy, Delete or Rename a file(s)

Copy, Delete, Rename Files. Change current folder.

- Backup files
- Delete temporary files
- Change folder prior to using an open dialog box

Change the display resolution.

Change or set the system's display resolution.

- Maintain screen resolution preferences among multiple users
- Develop presentations or .html that is optimized for all screen resolutions
- View graphics in high color resolutions
- Maintain multiple settings for laptop and desktop

Change the Default Printer

Choose a Default Printer.

- Switch between color and non color printers
- Maintain printer preferences among multiple users
- Quickly fax documents

Change Desktop Wallpaper

Change the desktop wallpaper.

- Place favorite images on the desktop
- Place a family picture on the desktop
- Tile the image (multiple copies spread over the desktop)
- Stretch, Center or Position the image on the desktop

Move or Resize a Window

Resize or Position a Window.

- Automatically position windows anywhere on the screen.
- Resize windows to allow perfect alignment of Windows
- Position the top most window in the center, top corner or anywhere on the screen.

Maximize/Minimize a Window

Maximize, Minimize, Restore, Hide, Close Windows.

- Maximize or Minimize the Top Most Window with a keystroke
- Temporarily hide windows and then restore (unhide) them when needed
- Shut down the Top Most program with a keystroke.

Open an Explorer Folder

Open up an Explorer Folder.

- Go directly to a specified folder in Explorer
- Clean up a cookie folder
- Clean up the temp folder
- Rearrange the start menu folders

Desktop Goodies

Cascade, Tile, Minimize or Restore All Windows. Undock and suspend computer.

- Quickly unclutter the desktop
- Align all windows on the desktop with a keystroke
- Undock the computer from a docking station by pressing a key

Run Control Panel

Open up the Control Panel or run a Control Panel applet.

- Go directly to Internet Settings to make changes
- Go directly to Display properties to make changes
- Go directly to Network Options to make changes
- Go directly to System Properties to make changes
- Go directly to Accessibility Options to make changes

Macro Express - Explorer > Quick Wizards Button > Works with Systems and Files

Text Based Wizards

Below is a listing of each of the Text Based wizards available.

Create a Reminder

Create a reminder to be displayed at a specific time.

- Create a reminder for an appointment or meeting.
- Create reminders for birthdays or anniversaries
- Create reminders for a chore
- Create reminders of an upcoming event

Paste Text or Graphics

Paste Text or Graphic images into programs automatically.

- Insert commonly used phrases, paragraphs or boilerplate into a Word Processor, Email and other programs such as:

  Answers to frequently asked questions
  Sales quotes
  Legal, Medical, Real Estate, etc documents
- Insert graphic images into documents.

Type the Date or Time

Type the current, future or past date or time into programs.

- Time stamp documents with today's date and time
- Date and time stamp an email
- Place yesterday's or tomorrow's date into a document

Type a Symbol

Place an international character or a symbol in programs.

- Insert Western European Language characters including â è ö î ú Ö Ë Æ ß ñ and more
- Insert Western European Language punctuation such as ¡ ¿
- Insert Math Symbols such as ÷ × ‰ ± ¼ ½ ¾
- Insert Currency Symbols such as ¢ £ ¥ $
- Insert Legal Symbols such as ® © ™

Type Some Text

Have Text, Keystrokes and Symbols typed into programs automatically.

- Insert into a Word Processor, Email and other programs commonly used phrases such as:
  Name, Address, Phone
  Headers & Footers
  Email Taglines
- Insert common keystrokes such as tabs or ALT commands to invoke menu options.

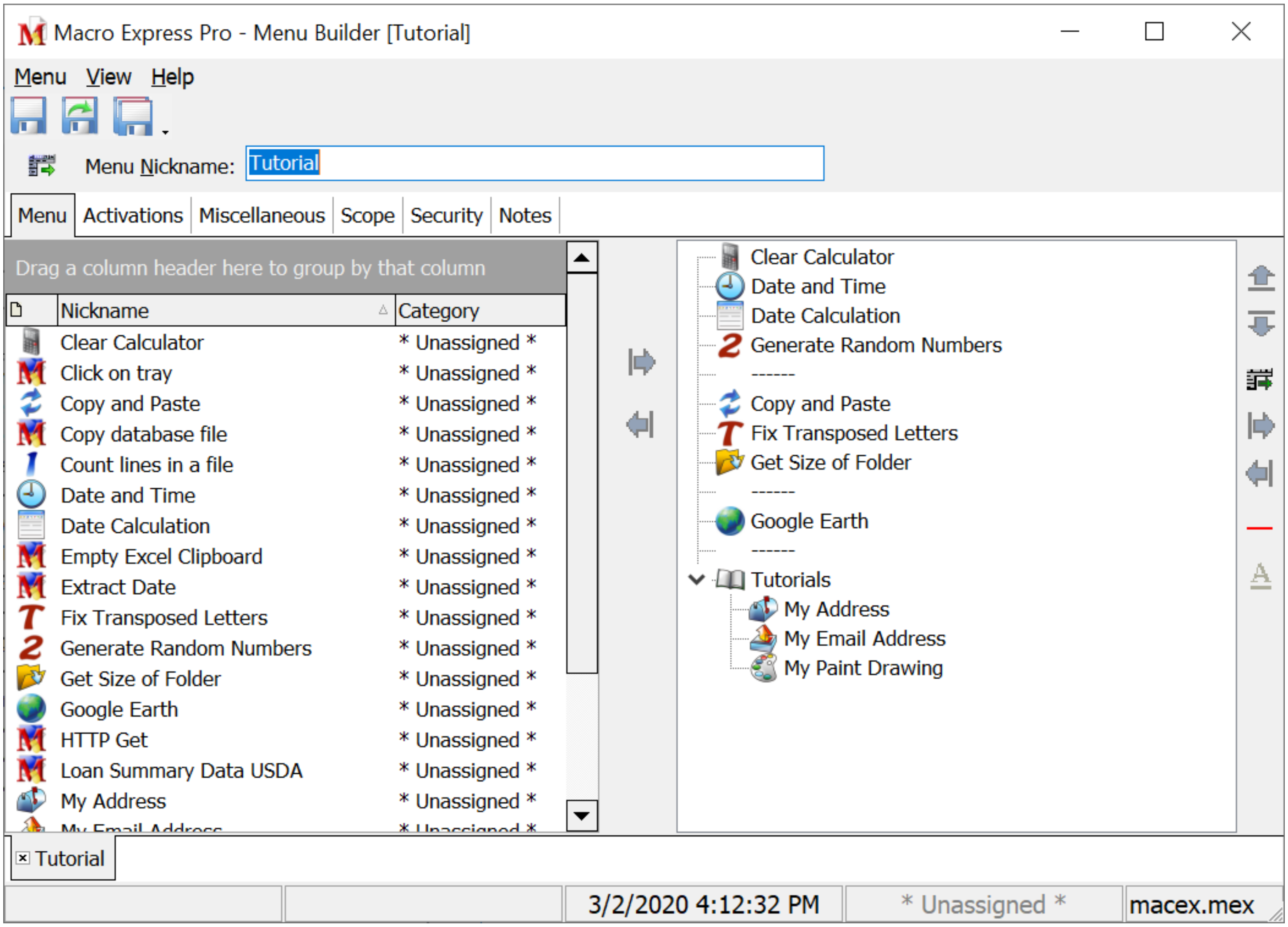
Macro Express - Explorer > Quick Wizards Button > Text-based Macros

Menu Builders

Menu Builder - Menu

The Macro Menu Builder creates popup menus or floating menus of the most frequently used macros or any other combination of macros. The completed popup menu can be activated by pressing a hot key or designating a notification area icon. The floating menu always stays on top of the desktop allowing continual access to the macros in the menu.

The Create a Popup Menu tutorial walks through the process of building a popup menu.



Macro Express Pro - Menu Builder [Tutorial]
Menu
View
Help
Menu
Activations
Miscellaneous
Scope
Security
Notes
Drag a column header here to group by that column

Menu Nickname: Tutorial

Nickname	Category
Clear Calculator	* Unassigned *
Click on tray	* Unassigned *
Copy and Paste	* Unassigned *
Copy database file	* Unassigned *
Count lines in a file	* Unassigned *
Date and Time	* Unassigned *
Date Calculation	* Unassigned *
Empty Excel Clipboard	* Unassigned *
Extract Date	* Unassigned *
Fix Transposed Letters	* Unassigned *
Generate Random Numbers	* Unassigned *
Get Size of Folder	* Unassigned *
Google Earth	* Unassigned *
HTTP Get	* Unassigned *
Loan Summary Data USDA	* Unassigned *
My Address	* Unassigned *
My Email Address	* Unassigned *

- Clear Calculator
- Date and Time
- Date Calculation
- Generate Random Numbers
- ------
- Copy and Paste
- Fix Transposed Letters
- Get Size of Folder
- ------
- Google Earth
- ------
- Tutorials
  - My Address
  - My Email Address
  - My Paint Drawing

3/2/2020 4:12:32 PM | * Unassigned * | macex.mex

Macro Express Explorer > Double click on existing Popup menu or Macro > Add Macro > Select Activation and Build as a Popup Menu > Click OK

Available Macros

All of the macros available for the menu are listed in the left hand column of the window. To include a macro in the menu, highlight the macro and click on the

Insert Macro icon to place the macro in the menu. Or highlight the macro and drag it to the right hand column to place it in the menu.

Macros in Menu

This column lists all of the macros included as part of the popup or floating menu. This is the order that they are displayed in the popup menu or the floating toolbar. To rearrange the order, highlight the macro to move and then click on the Up or Down buttons. The highlighted macro will be moved up or down in the list depending on the button selected.

To remove a macro from the menu, highlight the macro and then click on the Remove button. The macro is removed from the popup list and returned to the list of available macros. This macro can now be assigned to another menu, left alone or returned to the popup or floating menu later.

Menu Builder Navigation

This section describes the functionality of the buttons available to build and modify the menus.

Insert Macro

Highlight a macro from the list of available macros and click on the Insert Macro button to add the macro to the popup or floating menu.

Remove

Highlight a macro already placed in the menu and click on the Remove button to remove the macro from the menu. The macro will be displayed again as an available macro in the general menu.

Move Item Up

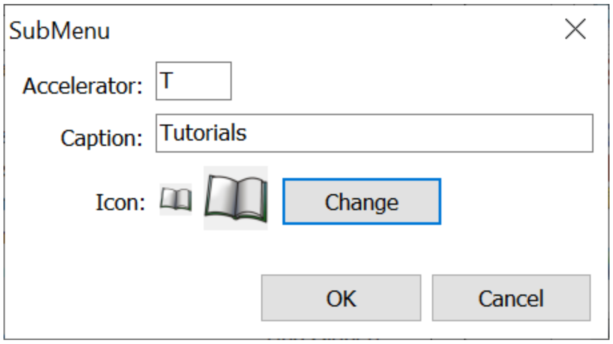
Highlight a macro in the menu and click on the Move Item Up button to change the position of the macro in the menu. Each click moves the macro up one spot in the menu.

Move Item Down

Highlight a macro in the menu and click on the Move Item Down button to change the position of the macro in the menu. Each click moves the macro down one spot in the menu.

Insert Submenu

Click on this button to open the SubMenu window as shown below. This option creates a submenu within the menu structure. Use the Up and Down arrows, or the Indent and Unindent buttons to move the submenu to its desired location.



Accelerator: T

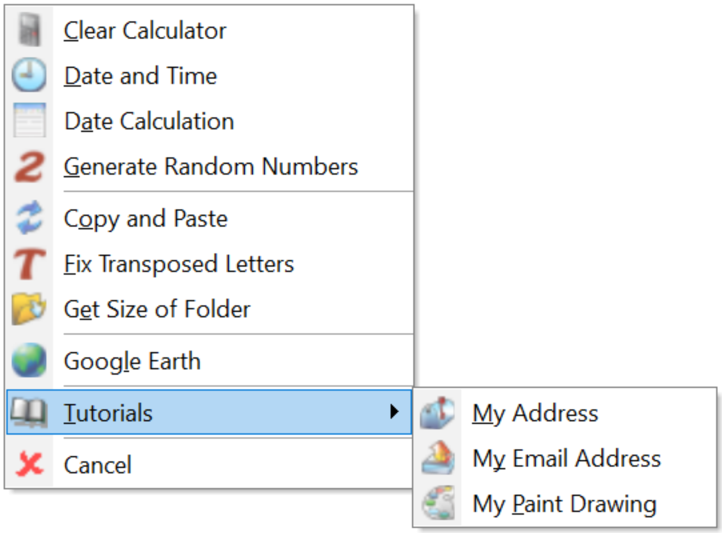
Caption: Tutorials

Accelerator: Assign any letter of the alphabet or a number from 0 to 9 as the submenu accelerator key. Pressing the assigned key expands the submenu. The assigned accelerator keys are not used with the Icon menus or the Windows Default style menu.

Caption: Assign a name for the submenu. This identifies the submenu.

Icon: Assign an icon to the submenu by clicking on the Change button. Locate an icon and select it for use.

Image of Popup Menu with Submenu Displayed



Indent

This option is available when using a submenu. Move the macro beneath the submenu using the Move Item Down button. Next click on the Indent button to move the macro so that it is now part of the submenu.

Unindent

This option removes the highlighted macro from the submenu and places it back in the general menu.

Insert Separator

This option inserts a separator line in the menu to separate two macros. This is strictly for visual effect.

Set Accelerator

Assign an accelerator key to the macro selected. Pressing the accelerator key plays back the macro when the menu is open. Highlight the macro and click on the Set Accelerator button for a prompt to assign an accelerator key. Choose from any letter of the alphabet or a number from 0 to 9 as the accelerator key.

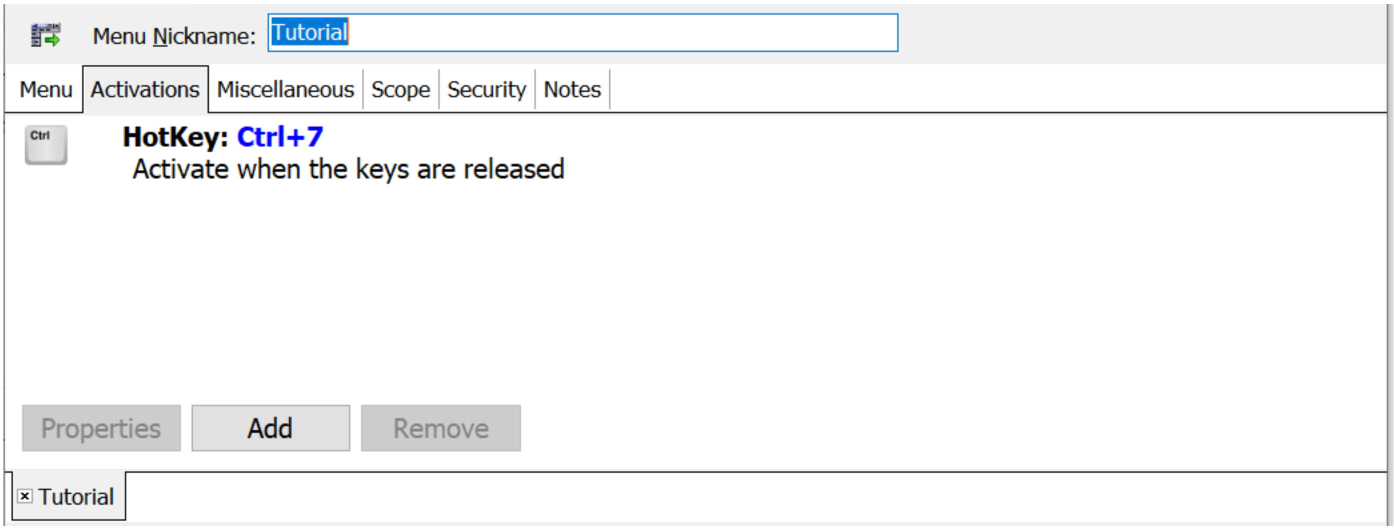
✓ **Floating Menu Hint:** If the floating menu has somehow been dragged far enough off the edge of the screen so that it cannot be drug back into position, follow one of these two procedures to correct the problem.

1. Right mouse click on the notification area icon. From the menu that opens left click on Tools > Restore Toolbars.
2. From the Macro Explorer, find the Floating Menu macro and double click on it to open the Menu Builder. Click on the Miscellaneous Tab, uncheck the Menu is Enabled box and click on the Save icon. Then place a check next to the Menu is Enabled box and click Save.

This restores the Floating menu so that it is again visible on the screen. The menu is placed at the 0,0 screen coordinates in the upper left hand corner of the screen.

Menu Builder - Activations

Typically the activation of a macro is assigned when first creating the macro. However, use this screen to change the macro activation or assign more than one activation method to a macro. The current menu activations and details about the activations are listed in this tab.


Menu Nickname: Tutorial

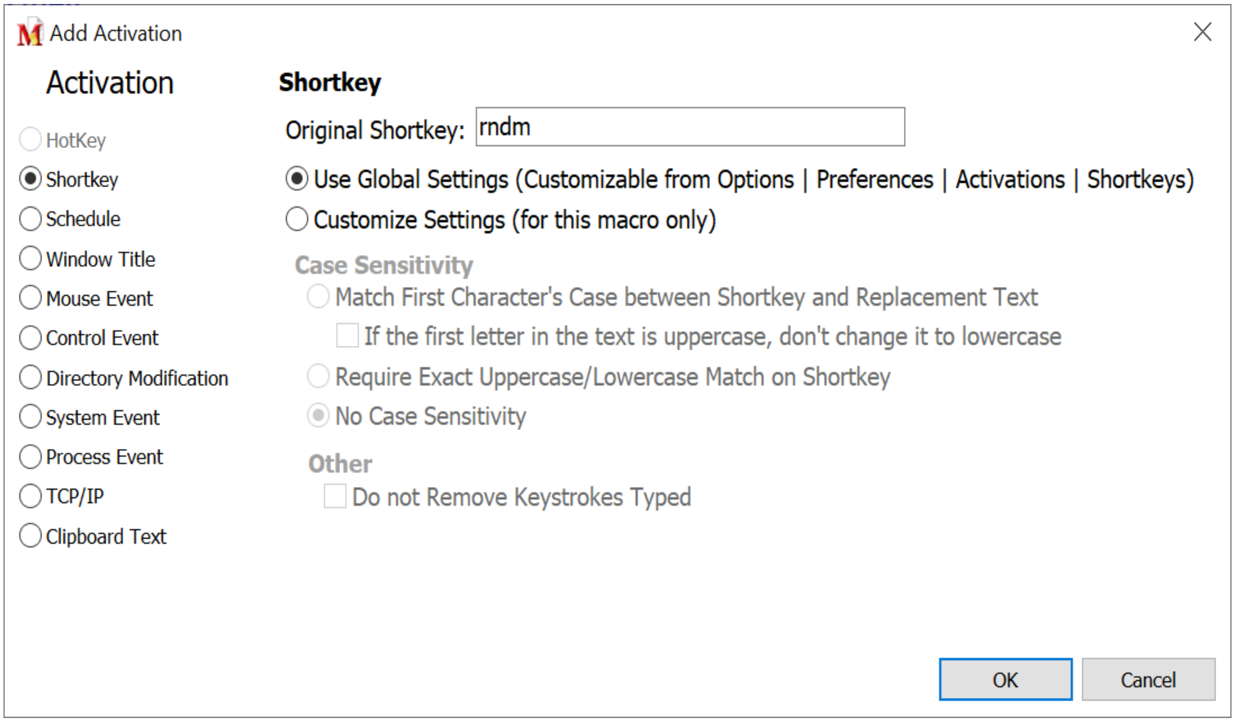
HotKey: Ctrl+7
Activate when the keys are released

Properties

To use the Properties option, first highlight an activation listed. This allows changes to be made to the highlighted activation. For example if the activation is a hot key of Ctrl+F, it may be assigned to another combination using this function.

Add

Click on the Add button to add another activation to the macro. The Add Activation window opens, as shown below, so that an additional activation may be selected.



Add Activation

Activation

- ○ HotKey
- ● Shortkey
- ○ Schedule
- ○ Window Title
- ○ Mouse Event
- ○ Control Event
- ○ Directory Modification
- ○ System Event
- ○ Process Event
- ○ TCP/IP
- ○ Clipboard Text

Shortkey

Original Shortkey: rndm

● Use Global Settings (Customizable from Options | Preferences | Activations | Shortkeys)

○ Customize Settings (for this macro only)

Case Sensitivity

○ Match First Character's Case between Shortkey and Replacement Text

□ If the first letter in the text is uppercase, don't change it to lowercase

○ Require Exact Uppercase/Lowercase Match on Shortkey

● No Case Sensitivity

Other

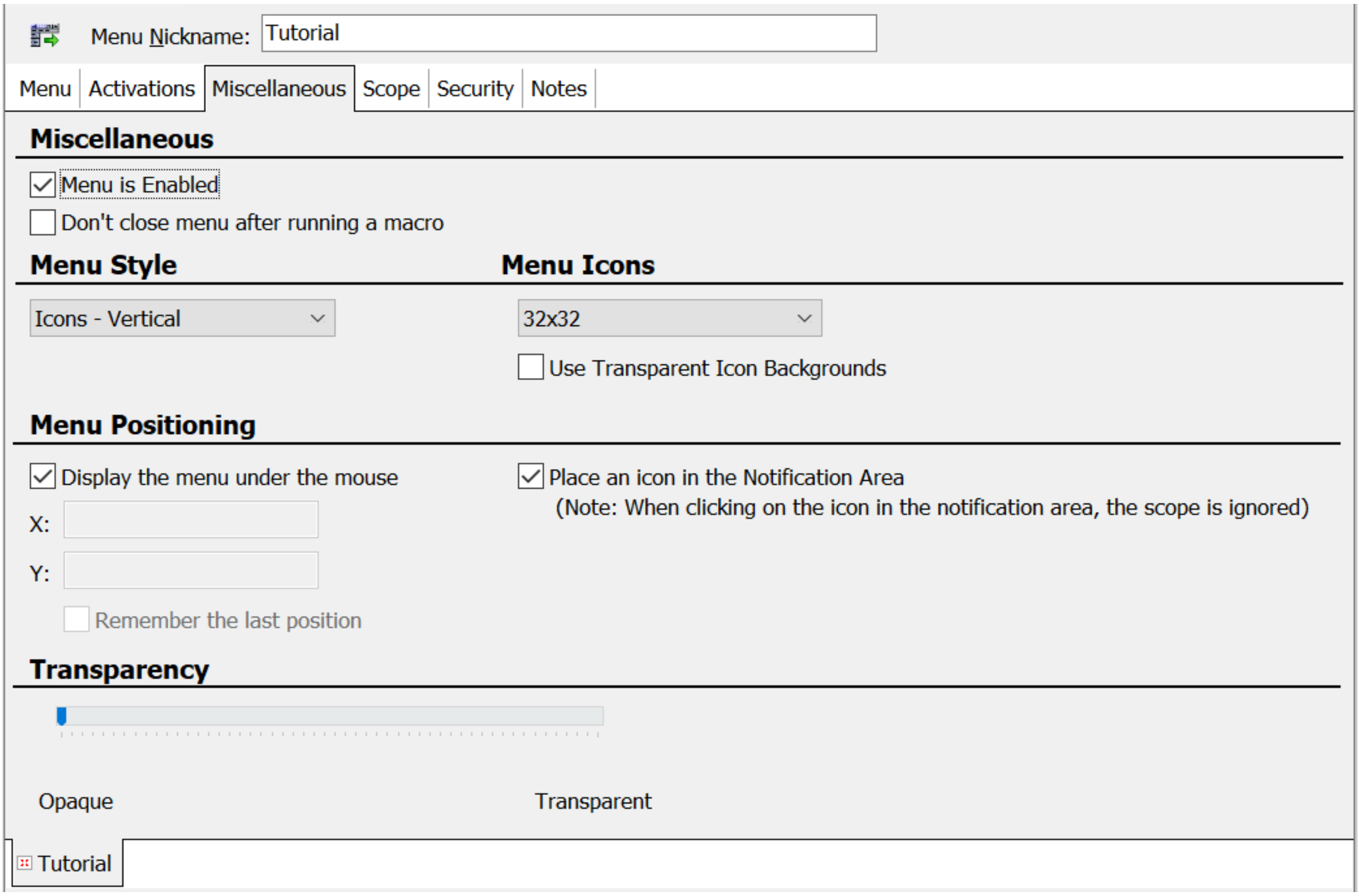
□ Do not Remove Keystrokes Typed

Remove

Highlight a macro activation and click on the Remove button to remove this activation from the macro.

Menu Builder - Miscellaneous

The Miscellaneous tab provides several different options for the menu being built. Select the menu style and where it displays on the screen, as well as other settings described below.



Menu Nickname: Tutorial

Miscellaneous

☑ Menu is Enabled

□ Don't close menu after running a macro

Menu Style

Icons - Vertical

Menu Icons

32x32

□ Use Transparent Icon Backgrounds

Menu Positioning

☑ Display the menu under the mouse

X:

Y:

□ Remember the last position

☑ Place an icon in the Notification Area
(Note: When clicking on the icon in the notification area, the scope is ignored)

Transparency

Miscellaneous

Menu is Enabled

This check box shows whether the menu has been disabled or is still an active menu. A check in the box signifies that the menu is enabled (active).

Don't close menu after running a macro (Popup menu only)

Select this option to keep the menu open after running a macro from the popup menu.

Don't allow menu to be closed (Floating menu only)

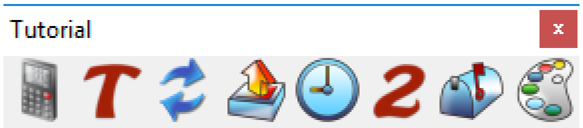
This option keeps the floating menu open with no chance to click on the red X or press Alt + F4 to close it. The menu may still be closed from the Macro Express - Explorer window by disabling the floating menu.

Menu Style

Use this option to select what information to display in the popup window. To more readily identify the macros placed in the popup menu, the macro information may be displayed in several different ways - Icons Only, First Part of the Macro, Nickname and Scope or Windows Default.

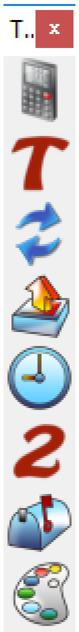
Icons - Horizontal

This option displays the icons assigned to each macro in the menu. The icons are displayed horizontally. The maximum number of icons displayed in a single row is 20.



Icons - Vertical

This option displays the icons assigned to each macro in the menu. The icons are displayed vertically. The maximum number of icons displayed in a single column is 20.



Menu Icons

When using menu icons, select from the drop down list the size of the icons to appear in the popup window or floating menu. The sizes are listed in pixels. The icon size options are 16x16, 24x24, 32x32, 48x48, and 64x64 pixels.

Use Transparent Icon Backgrounds

Select this option to display the backgrounds of the icons transparently.

Clicking on the icon in the menu starts the playback of the macro.

First Part of the Macro

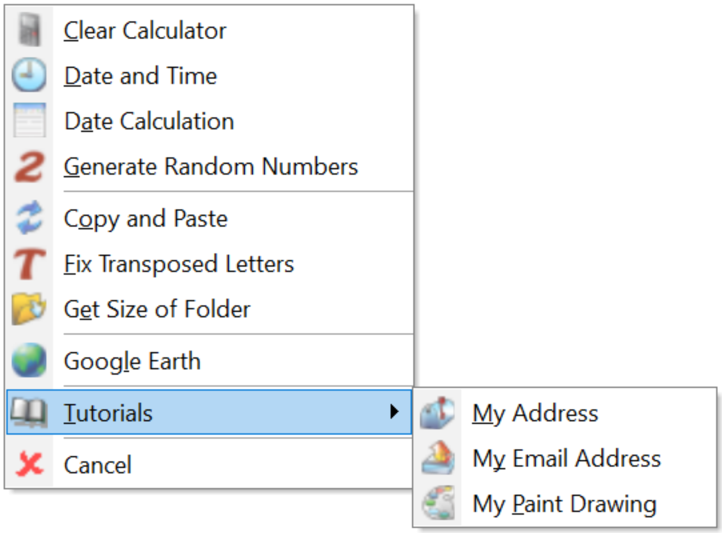
This option displays the first few words or commands of the macro script in the menu. When the popup window appears, either double click on the macro or press the number or letter associated with the macro to have it play back.

Nickname and Scope

The Nickname and Scope option displays the name assigned to the macro and the macro scope, such as a global, or program or window specific macro. When the popup window appears, either double click on the macro or press the number or letter associated with the macro to play it back.

Windows Default (Popup menu only)

The Windows Default menu displays the icon associated with the macro and the macro nickname. Click on the macro to start the playback. Also assign accelerator keys to the macros using the Set Accelerator A button. Pressing the A key activates the My Address macro in the example below. The accelerator keys are underlined and displayed to the left of the macro name.



Menu Positioning

Determine where to display the menu when activated.

Display the Menu under the Mouse

This option displays the menu underneath the current position of the mouse cursor. The upper left corner of the menu is positioned under the mouse cursor.

X, Y

Enter the X and Y screen coordinates where the popup menu should be displayed on the screen. The default is to display in the upper left hand corner of the screen with coordinates of 0,0.

Place an Icon in the notification area

This option is only available for the popup menu. The icon associated with the popup menu is placed in the notification area - on the bottom right of the screen near the clock. A left mouse click on the icon displays the popup menu.

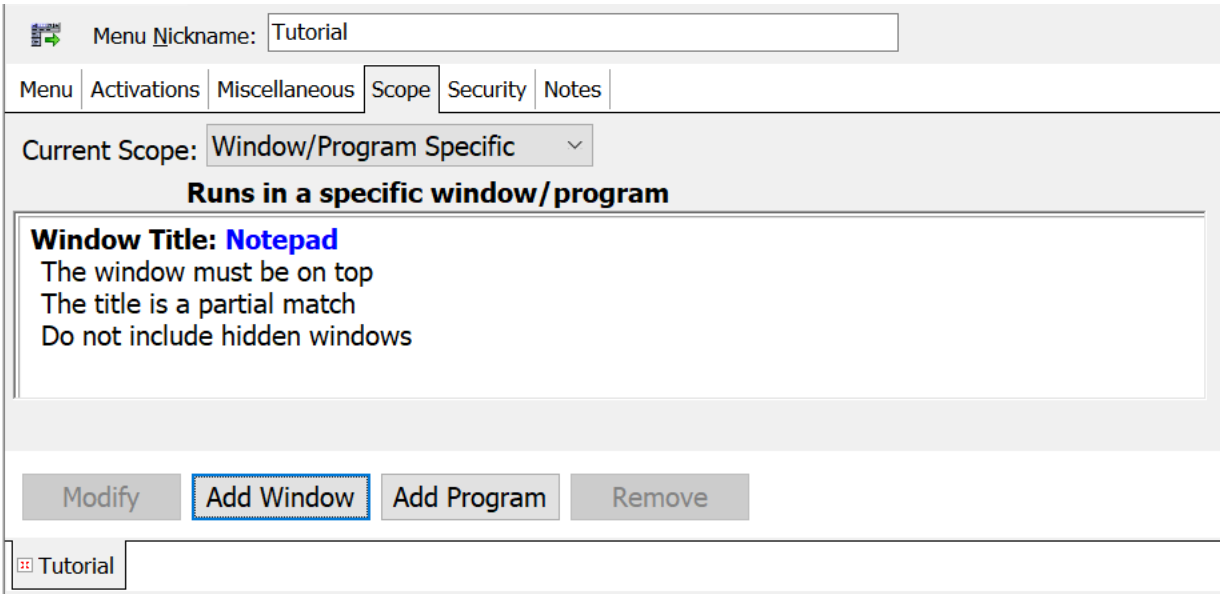
- ✓ **Note:** The scope of the menu is set to Global when this option is selected.

Transparency

When using any of the menu styles, other than Windows Default, optionally set the transparency level of the menu. The transparency setting enables viewing the text or image underneath the menu when it is displayed. Moving the slider to the right makes the menu image more transparent.

Menu Builder - Scope

The scope for a popup menu is similar to that of a macro. Assign the menu to only open in a specific window or program, in all windows or programs except for those chosen, or allow it to open in all applications. Floating menus always have a Global scope.



Menu Nickname: Tutorial

Current Scope: Window/Program Specific

Runs in a specific window/program

Window Title: Notepad
The window must be on top
The title is a partial match
Do not include hidden windows

Global

The global option allows the menu to open in any Windows application.

- ✓ **Note:** If opting to display the icon in the notification area, the scope is automatically Global. The icon in the notification area displays regardless of which program or window is open and behaves as a global macro. If a hot key is assigned to the popup menu, the hot key may be assigned to a scope.

Example
Assume that a popup menu icon is assigned to display in the notification area. But a hot key (CTRL Q) is also assigned to the same popup menu. The scope is set to Program Specific and is to only work with Wordpad.

So if is Wordpad active and CTRL Q is pressed, the popup menu appears. Open notepad and nothing happens after pressing CTRL Q.

But it is still possible to left click on the icon in the notification area to open the menu. The menu is always global when clicking on the icon in the notification area.

Global Except

This option allows the menu to open in all Windows applications, except the program or window selected. More than one window or application may be listed.

Window/Program Specific

With this option, the menu opens only if the specific window or program selected is active. If the window or application is not active the menu does not open.

Buttons

These four buttons are only available when using the Window/Program Specific or Global Except scopes.

Modify

Highlight an entry made in the scope window and click on the Modify button. This allows for changes to the window or program previously selected.

Add Window

Click on the Add Window button to view all windows that are currently running on the computer. One tab lists all visible windows currently running and a second tab lists all windows running that are hidden. Hidden windows are not displayed when an application is running, but are still active.

If the window requested is currently running, scroll through the list and click on the title. The window name is placed in the Window Title edit field. If the window is not currently running, type the window name in the Window Title edit field.

Exact Match/Partial Match

The exact match option requires that the Window title match exactly before the menu opens. A partial match will only require part of the window title in order to work. For example, "notep" would be sufficient to find the "notepad" window.

The title contains wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

A Window with this Title Must Be on Top

This option opens the menu only if the window is the current top-most window (the one that is showing on top of all other windows).

Include Hidden Windows in the Scope

By default the macro will only look for windows that are visible, not those that run hidden and are not seen by the user. Select this option to have the scope apply to the windows listed in the Hidden tab, as well as those that are visible.

After making a selection, click OK and the window is listed in the Scope tab. The process may be repeated to add additional windows if necessary.

Add Program

Click on the Add Program button to view all programs that are currently running on the computer. One tab displays the tree view of the programs in the Start Menu. The other tab displays a list of processes currently running. Scroll through the list or tree view and highlight the program to display the program name in the Executable Path edit box. If the program is not found in either tab, manually type the file name and path or click on the Browse button to find the file on the computer.

This Program Must Have a Window on Top

This option only opens the menu if the program is the current top-most program (the one that is showing on top of all other windows).

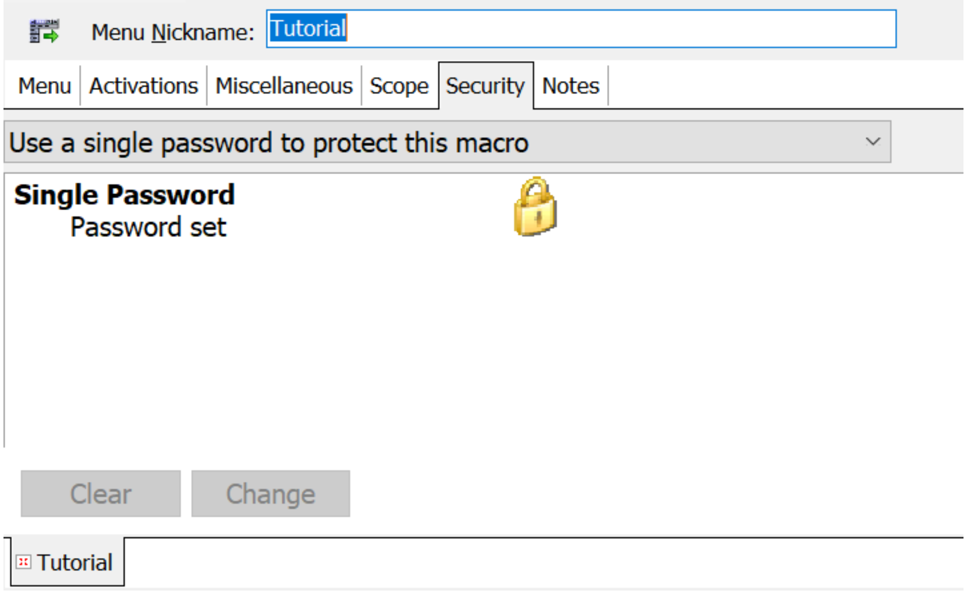
After making a selection, click OK and the program is listed in the Scope tab. The process may be repeated to add additional programs if necessary.

Delete

Highlight an entry made in the scope window and click on the Delete button. The scope entry previously made is removed.

Menu Builder - Security

Use the Security tab to set up a password that is required before the menu can be edited, played back or both. Three security options are available.



Menu Nickname: Tutorial

Use a single password to protect this macro

Single Password
Password set

Do Not Password Protect this Macro

This is the default setting. No password is required to view, edit or run the menu.

Use a Single Password to Protect this Macro

Select this option and click on the Change button to set up a password. With the password information defined, the password must be entered before viewing or editing the contents of the menu is possible. This same password is also required before running the menu.

Use Separate Passwords for Editing and Running This Macro

Select this option to create one password to edit the menu and another password to run the menu. Or choose to have a password for either the Edit or Run and not use a password for the other.

Edit Password

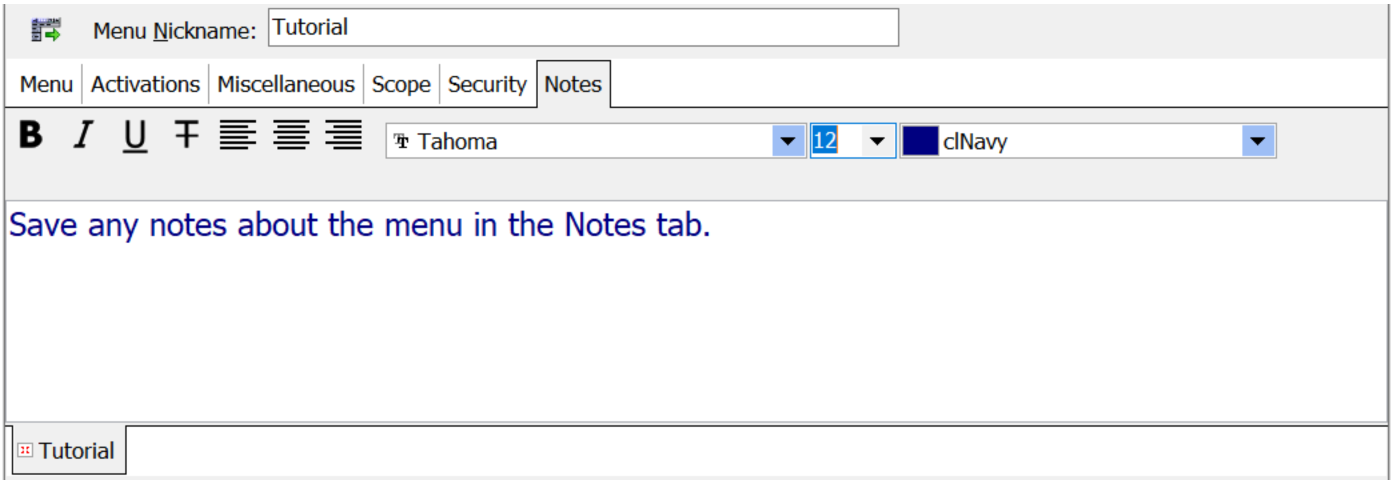
Highlight Edit Password and click on the Change button to set up a password. With the password information defined, then password must be entered before view the contents of the menu. This is one way to prevent others from viewing the macros or being able to alter them.

Run Password

Highlight Run Password and click on the Change button to set up a password. With this password information defined, the password must be typed before the menu opens.

Menu Builder - Notes

This tab opens the Notes field. This is a free form text editor for keeping notes about the macro. Use this option to make some notes about the menu, how it is to be used or how frequently. Essentially this field can be used to input any pertinent information about the menu.



Menu Nickname: Tutorial

Save any notes about the menu in the Notes tab.

The Notes field supports Rich Text editing, including the use of various font types, sizes and colors, plus bold text, italics, underlines and strikeouts.

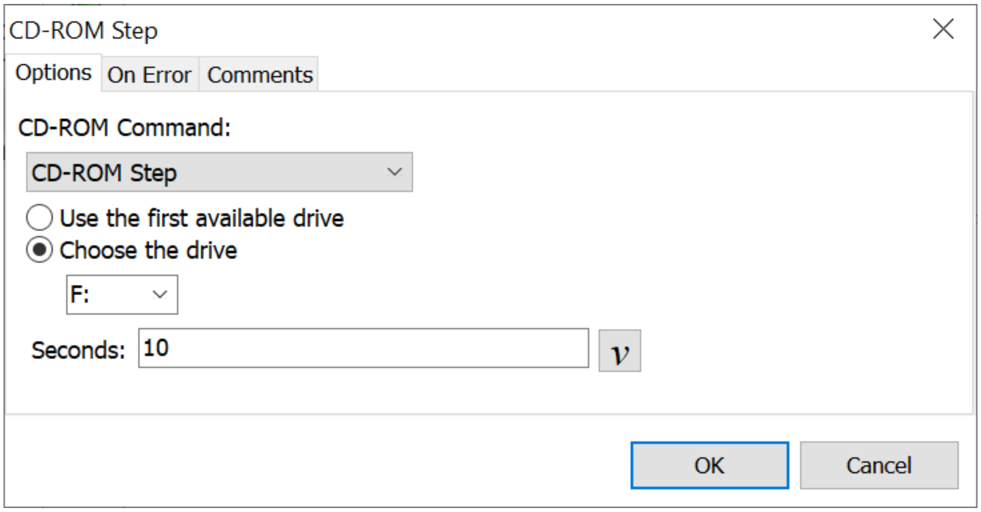
Macro Commands

CD-ROM

CD-ROM

*Commands deprecated - See note below

If you enjoy listening to audio CD's while working at the computer, you can create a number of macros to automate features found in the CD-Player application shipped with Windows. Create macros to replay a favorite track, open and close the tray, start, stop or pause the CD for an interruption, or skip tracks -- practically anything except physically inserting the CD into the drive.



CD-ROM Command: CD-ROM Step

○ Use the first available drive

● Choose the drive

F:

Seconds: 10

Script Editor > Expand CD-ROM Category > Select Choice

Command	Description
Play CD-ROM	Starts the play of a CD inserted in the CD-ROM drive.
Stop CD-ROM	Stops the play of the CD.
Pause CD-ROM	Pauses the playback of the CD. This is handy when the phone rings or some other type of interruption occurs.
CD-ROM Next Track	Skips forward to the next track of the CD.
CD-ROM Previous Track	Backtracks to the previous track of the CD.

CD-ROM Goto Track	Automatically forwards to the track number requested. A small box prompting for the desired track number appears. Enter the track number and click OK.
CD-ROM Step	Specify the number of seconds to fast forward through the CD. Enter the number of seconds to step forward in the Seconds edit box.
CD-ROM Back	Specify the number of seconds to backtrack through the CD. Enter the number of seconds to backtrack in the Seconds edit box.
Eject CD-ROM	Opens the CD tray to eject the CD.
Close CD-ROM Tray	Closes the CD tray.

If multiple CD-ROM drives exist on the computer, such as a standard drive and a CD-ROM burner, use these options to select which drive to use.

First Available

This option selects the first available CD-ROM drive.

Choose

Use this option to select which drive to use.

CD-ROM Drive Letter

Use this in conjunction with the Choose option. Use the drop down list to select which drive to use in the macro.

---

*Deprecated Commands

Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future. Where possible we recommend that you avoid the use of deprecated macro commands.

By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option found in Options, Preferences, General - Misc tab.

Clipboard Copy, Cut and Paste

The following clipboard commands do not open an edit window in the Script Editor. These commands are inserted directly into the script. No modifications are required. Expand the Clipboard category in the Script Editor and select the command needed.

Clipboard Copy

Clipboard Copy is the same as issuing a Ctrl + c in Windows. It copies to the clipboard whatever is selected (highlighted). This function should follow commands in the macro script that highlight the text or image.

Clipboard Cut

Clipboard Cut is the equivalent of issuing a Ctrl + x in Windows. It deletes the selected (highlighted) text or image and copies it to the Windows clipboard. Follow this command by saving the clipboard contents to a variable or issuing a clipboard paste command.

Clipboard Empty

This command clears or empties the clipboard of all its contents.

Clipboard Paste

Clipboard Paste is the same as issuing a Ctrl + v in Windows. The Paste command inserts the contents of the Windows clipboard into the application currently running.

Clipboard Start Copy / Clipboard End Copy

The Clipboard Start Copy command instructs the macro to consider anything that follows in the macro to be copied to the clipboard. It copies everything in the script to the clipboard until it finds the Clipboard End Copy command.

In most cases the Text Type command is much more convenient than using the Clipboard Start Copy and Clipboard End Copy commands. Insert the text into the Text Type command and select the Use Clipboard and Paste Text option.

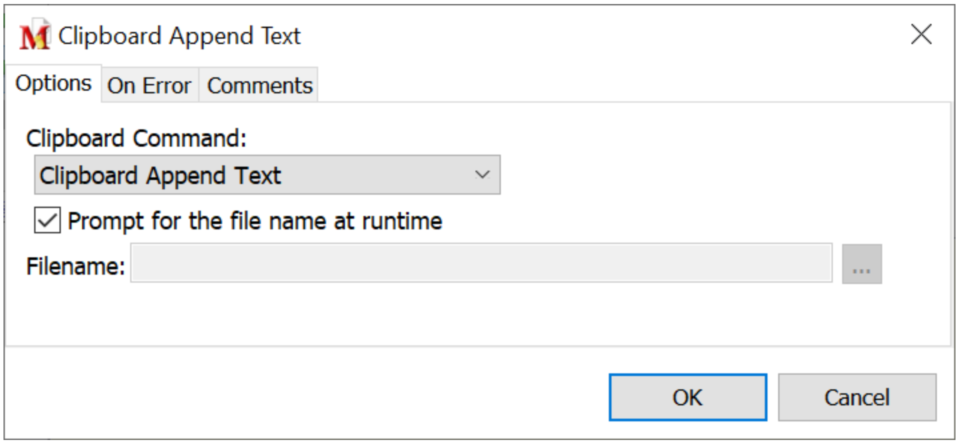
- ✓ **Note 1:** The Clipboard Copy, Cut and Paste commands will not work in applications that do not support the standard Windows shortcuts of Ctrl + c, Ctrl + x or Ctrl + v. Most applications support these Windows standards, but there are some that do not.

- ✓ **Note 2:** Windows requires a small amount of time to process a Clipboard function. If the macro is ignoring a copy, cut or paste during playback, the macro may be going faster than Windows can process the clipboard functions.

  To make sure the macro doesn't play too fast, click on Options > Preferences > Playback > Delays. Bump the default clipboard delay value from 250 milliseconds to a larger delay. Some experimentation may be necessary to determine the optimal value for the system. The other alternative is to insert a Delay after the clipboard command to allow Windows the necessary time.

Clipboard Commands

These clipboard commands save the clipboard contents to either a text file or a graphics file. Other options include appending text to an existing file or having the clipboard contents typed out into another application.



Clipboard Append Text

Clipboard Command: Clipboard Append Text

☑ Prompt for the file name at runtime

Filename:

Script Editor > Expand Clipboard Category > Select One of the Options in this Topic

Type Clipboard Text

This option types the clipboard text letter by letter into another application instead of pasting the entire contents. Although this method inserts the text slower, (especially large blocks of text), there are applications which do not accept standard Windows clipboard paste commands.

Ignore CR/LF

The Ignore CR/LF function instructs the macro to type the clipboard text without any carriage returns or line feeds that may have been copied to the clipboard.

- Select this option to ignore carriage returns and line feeds.
- Leave the box unchecked to include the carriage returns and line feeds in the text playback.

Clipboard Save Text

Clipboard Save Text saves the text contents of the clipboard to a text file. Any images, etc. are ignored.

Prompt for File Name at Macro Run Time

Select this option to be prompted at macro runtime to enter the file name to store the clipboard contents.

To not receive a prompt, enter a file name and the full path or click on the

Browse button to search for the desired text file. Click the OK button to save when finished.

Clipboard Append Text

The Clipboard Append Text option appends (adds) the text contents of the clipboard to the end of an existing text file.

Prompt for File Name at Macro Run Time

Choose this option to be prompted, when running the macro, to enter the file name that will receive the text from the clipboard.

To not receive a prompt, enter a file name and the full path or click on the Browse button to search for the desired text file. When finished click the OK button to save.

Clipboard Save Graphic

The Clipboard Save Graphic option saves the graphic content of the clipboard to a .PNG, .JPG, Bitmap, Windows Metafile or Enhanced Metafile file type.

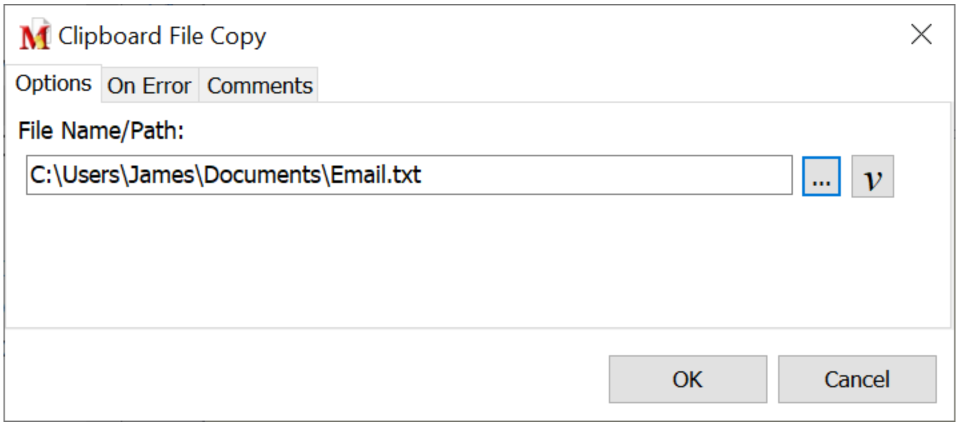
Prompt for File Name at Macro Run Time

Choose this option to be prompted, when running the macro, to enter the file name to store the graphic image from the clipboard.

To not receive a prompt, enter a file name and the full path or click on the Browse button to search for the desired text file. When finished click the OK button to save.

Clipboard File Copy

The Clipboard File Copy command copies the contents of a text file directly to the clipboard. Rather than opening the file, highlighting the text and copying to the clipboard, this command reads the contents directly from the file and places the contents into the clipboard. Issuing a subsequent Paste command pastes the contents of the text file into the application currently open.



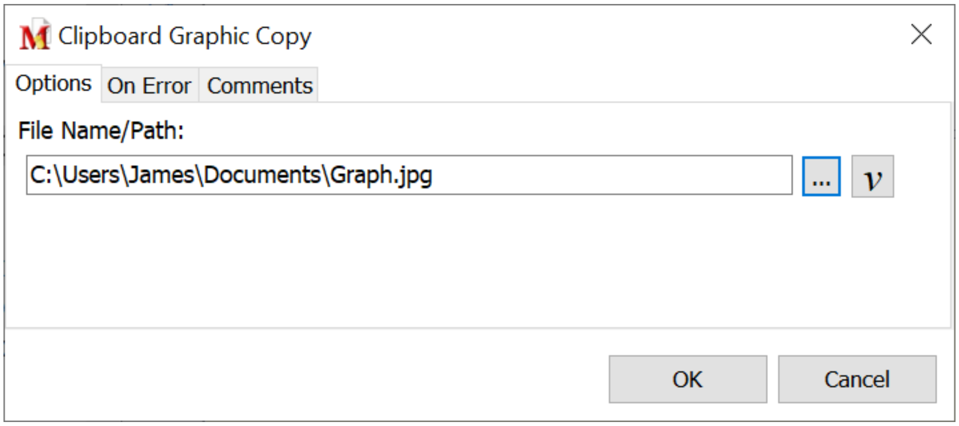
File Name/Path: C:\Users\James\Documents\Email.txt

Script Editor > Expand Clipboard Category > Clipboard File Copy

Enter a file name and the full file path or click on the Browse button to search for the desired text file. Click the OK button to save.

Clipboard Graphic Copy

The Clipboard Graphic Copy command copies a .PNG, .JPG, Bitmap, Windows Metafile or Enhanced Metafile image directly into the clipboard. Issuing a subsequent Paste command pastes the image into the application currently open.



File Name/Path: C:\Users\James\Documents\Graph.jpg

Script Editor > Expand Clipboard Category > Clipboard Graphic Copy

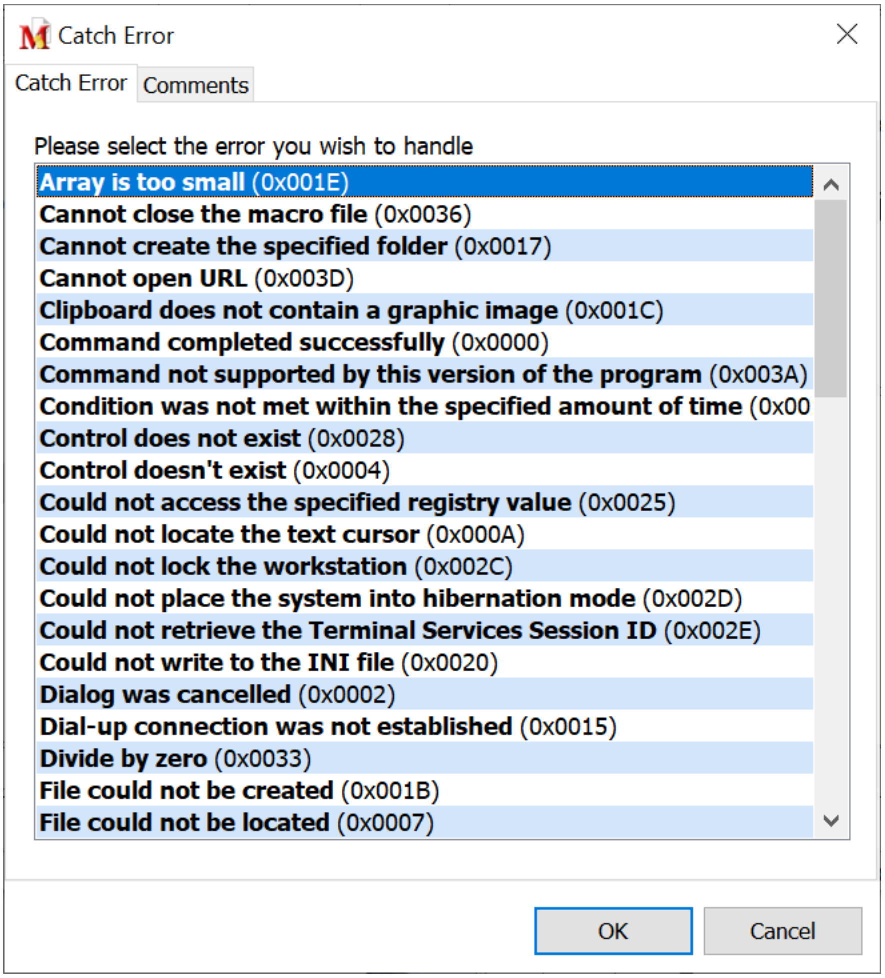
Enter a file name and the full file path or click on the Browse button to search for the desired image file. Click the OK button when finished to save.

Debug

Catch Error / End Catch Error

The Catch Error command is used to handle potential errors during the playback of a macro.

Practically all macro commands have an On Error tab. This tab provides several options for handling a potential error. The Catch Error command is available for use with either the Prompt for Action or Handle Error options. The other On Error tab options ignore the Catch Error command or halt the macro before encountering the Catch Error command in the script.



Please select the error you wish to handle

- **Array is too small** (0x001E)
- **Cannot close the macro file** (0x0036)
- **Cannot create the specified folder** (0x0017)
- **Cannot open URL** (0x003D)
- **Clipboard does not contain a graphic image** (0x001C)
- **Command completed successfully** (0x0000)
- **Command not supported by this version of the program** (0x003A)
- **Condition was not met within the specified amount of time** (0x00
- **Control does not exist** (0x0028)
- **Control doesn't exist** (0x0004)
- **Could not access the specified registry value** (0x0025)
- **Could not locate the text cursor** (0x000A)
- **Could not lock the workstation** (0x002C)
- **Could not place the system into hibernation mode** (0x002D)
- **Could not retrieve the Terminal Services Session ID** (0x002E)
- **Could not write to the INI file** (0x0020)
- **Dialog was cancelled** (0x0002)
- **Dial-up connection was not established** (0x0015)
- **Divide by zero** (0x0033)
- **File could not be created** (0x001B)
- **File could not be located** (0x0007)

Script Editor > Expand Debug Category > Catch Error

If there is a possibility that one of the commands in the macro may fail because of a network error, a file not found error, a window did not open failure, etc. then one option is to use the Catch Error command.

Of course another option is to simply allow the macro to stop if the error occurs.

But it may be preferable to have the macro take specific corrective action using the Catch Error command. The box below illustrates how this command is used in the Script Editor. This set of commands is placed immediately after the command chosen to handle the error.

Between the Catch Error and End Catch Error commands insert the macro commands to handle the error. This may include a corrective action to allow the macro to continue. Or insert an entire alternate script and end the macro before it can finish the entire script by using the Macro Stop command.

Sample Macro - Catch Error Syntax

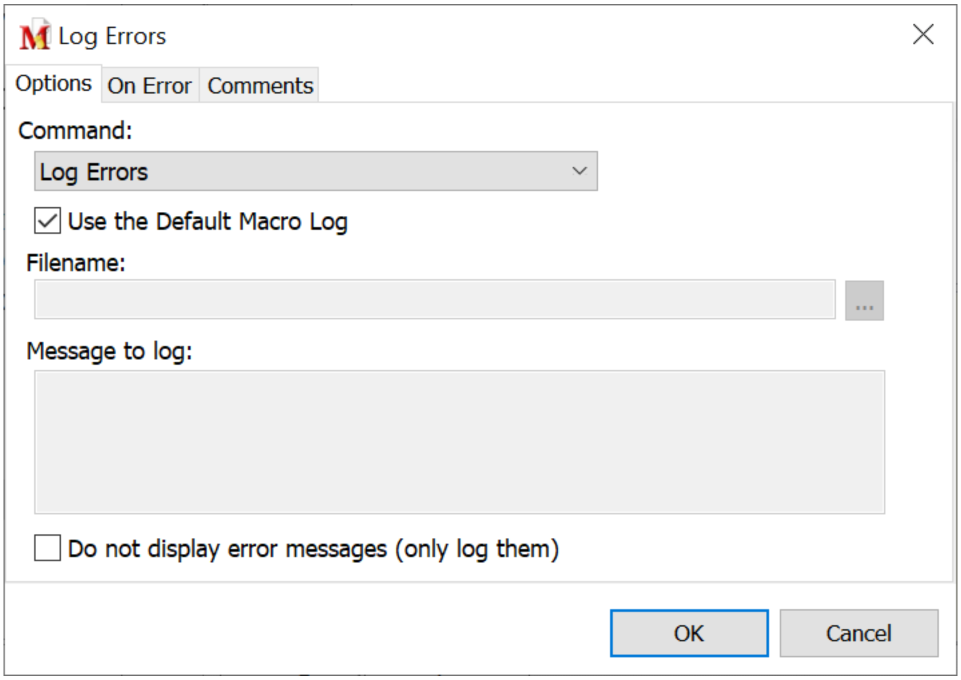
```
Window Activate: xyz
On Error
  Catch Error: The window could not be found
    Insert macro commands here to handle the error.
  End Catch Error
End Error
```

List of errors available for handling

Array is too small
Cannot close the macro file
Cannot create the specified folder
CD command could not be executed
Clipboard does not contain a graphic image
Command completed successfully
Condition was not met within the specified amount of time
Control doesn't exist
Could not access the specified registry value
Could not locate the text cursor
Could not lock the workstation
Could not place the system into hibernation mode
Could not retrieve the Terminal Services Session ID
Could not write to the INI file
Dialog was cancelled
Dial-up connection was not established
Divide by Zero
File could not be created
File could not be located
File could not be opened
File format is unsupported
File is not the correct format
Files required for Dial-Up Networking are not installed
Invalid Date/Time value
Invalid Input
Label does not exist
Macro does not exist
Macro file is already loaded
Macro is locked by another user
Macro was halted
Maximum is smaller than the minimum
Message could not be sent
Network error
Notification area icon does not exist
Operation was not completed
Path does not exist
Process could not be terminated
Process is not running
Program could not be launched
Registry key cannot be created
Registry key cannot be deleted
Registry value does not match the variable type
Requested feature is not supported
Resolution is unsupported

Log Errors

This command logs any errors that occur during the execution of a macro to a text file. When this command is encountered, an entry is immediately made in the error log file to indicate that a macro has started. Place this macro command as the first step in the macro script to ensure that the entry is made when the macro runs.



Command: Log Errors

☑ Use the Default Macro Log

Filename:

Message to log:

☐ Do not display error messages (only log them)

Script Editor > Expand Debug Category > Log Errors

A macro completion entry is made in the log file when the macro completes, indicating that the macro completed successfully. All entries are date/time stamped. This command is especially useful for unattended scheduled macros.

Optionally turn off the display of any error messages so that Macro Express is not stuck waiting for user intervention in cases where the macro runs unattended.

Use the Default File Log

Select this option to have the error messages sent to the default file log. The default error log folder can be set from Options > Preferences > File System - File Paths tab. Each macro is assigned its own individual log file.

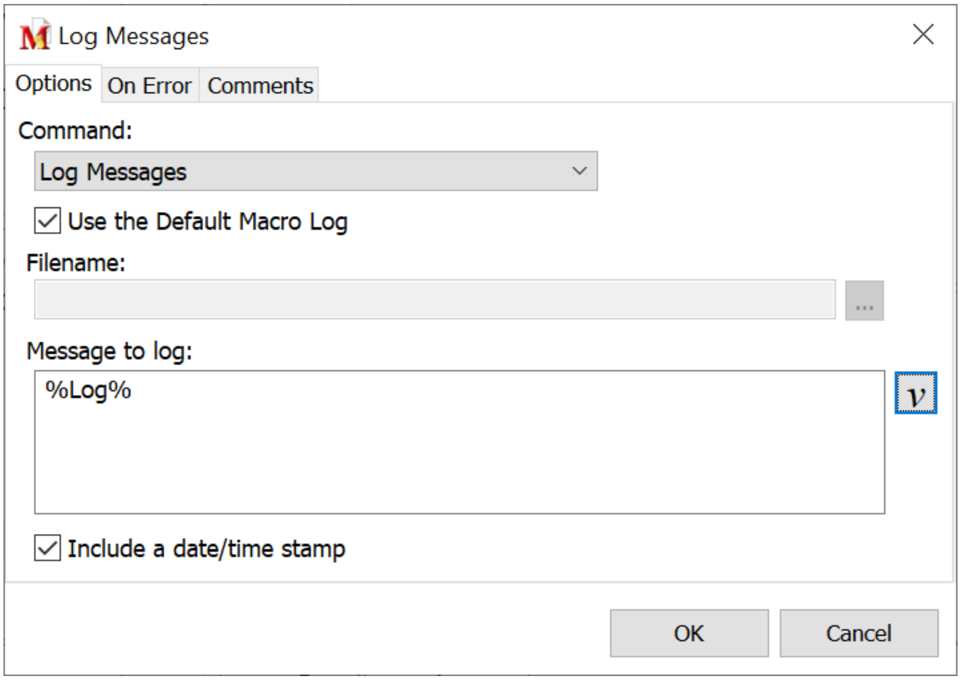
If not using the Default File Log, then insert the file name and path of the file to log the errors. Or click on the Browse button to locate the file.

Do Not Display Error Messages (Only Log Them)

This option suppresses the display of Macro Express error messages during the playback of a macro. An error message typically requires user intervention to clear the message from the screen. The error message is sent to a log file for later review.

Log Messages

The Log Messages command logs messages to a file. This could be used to log that the macro has progressed to a certain point in the macro and then log subsequent messages about the macro's progress.



Command: Log Messages

☑ Use the Default Macro Log

Filename:

Message to log: %Log%

☑ Include a date/time stamp

Script Editor > Expand Debug Category > Log Messages

Use the Default File Log

Select this option to have the messages sent to the default file log. The default log file can be set from Options > Preferences > File System - File Paths tab.

If not using the Default File Log, then insert the file name and path of the file to log the messages. Or click on the Browse button to locate the file.

Message to Log

Insert the message to be entered into the log file.

Include a Date/Time Stamp

Check this box to include a date and time stamp of when the entry was made to the log. The date/time stamp is based on the date and time of the system clock.

On Error / End Error

These commands are inserted directly into the script as there is not an edit window associated with them. From the Script Editor click on the Debug category of commands to expand the category. Then select either the On Error or End Error command to insert into the script.

The **On Error** command signals to the macro that action needs to take place if an error occurs with the macro command immediately preceding the On Error command during playback. This command is used in conjunction with the Catch Error message. Inserting the On Error and End Error commands without using the Catch Error command does nothing during macro playback. The commands will be ignored.

- ✓ **Note:** The On Error command must immediately follow the macro command that it is checking for errors. For example, a comment is not allowed between that command and the On Error command.

The **End Error** command closes the error handling sequence and must be paired with the On Error command. The commands in the Script Editor would look similar to the example illustrated below.

Sample Macro - On Error Syntax

```
On Error
  Catch Error: The window could not be found
    Insert macro commands here to handle the error.
  End Catch Error
End Error
```

Desktop

The Desktop commands provide a number of handy shortcuts for arranging the desktop. These commands are the same as those found in Windows by right clicking on a blank area of the Task Bar and selecting a function.

There is not an edit window available for these commands in the Editors. The commands are inserted directly into the script. From the Editor expand the Desktop category and double click on the command required.

Command	Description
Cascade Desktop Windows *Command deprecated	Layers all open windows to display the caption bar of each window.
Tile Desktop Windows Horizontally *Command deprecated	Arranges all open windows horizontally on the screen.
Tile Desktop Windows Vertically *Command deprecated	Arranges all open windows vertically on the screen.
Minimize All Windows	Minimizes all active windows so that the desktop is visible.
Restore All Windows	This restores all windows to their original size from a minimized state.

*Deprecated Commands

Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future. Where possible we recommend that you avoid the use of deprecated macro commands.

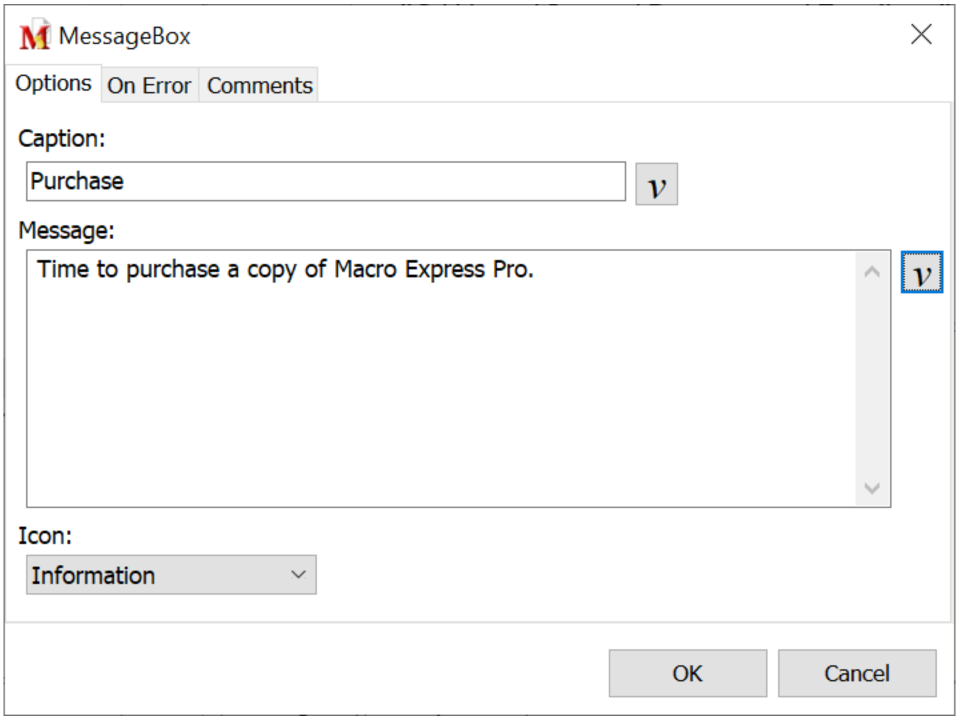
By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option found in Options, Preferences, General - Misc tab.

Dialogs

Message Box

Use the Message Box to display messages during the playback of the macro. Use this function to create reminder messages. Set the macro to run on a schedule to remind of an important event. Or generate the message box to instruct users of what to do next in the macro.

This option is similar to the Text Box Display command, but with several differences. The message box displays one of several message icons. Several of the icons produce an accompanying Windows sound when the message box is displayed. But there are no options for dictating the length of time that the box should appear or rich text features.



Caption: Purchase

Message: Time to purchase a copy of Macro Express Pro.

Icon: Information

Script Editor > Expand Dialogs Category > Message Box

Caption

Enter the caption in this field. This entry is displayed as the window title of the message box that opens during macro playback.

Message

Insert the text of the message in this edit field.

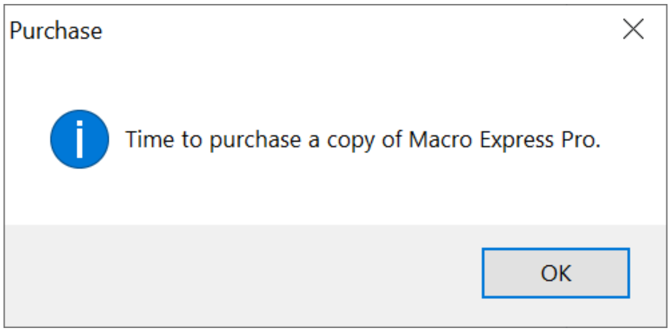
Icon

Select an icon to display next to the text of the message. Or choose to display no icon by selecting the None option. Each of the icon options are shown below.

⚠	Warning
ℹ	Information
?	Question
⊗	Error

Sample image

Below is an example of what the Message Box looks like during macro playback. This is based on the entries we made in the MessageBox image above.

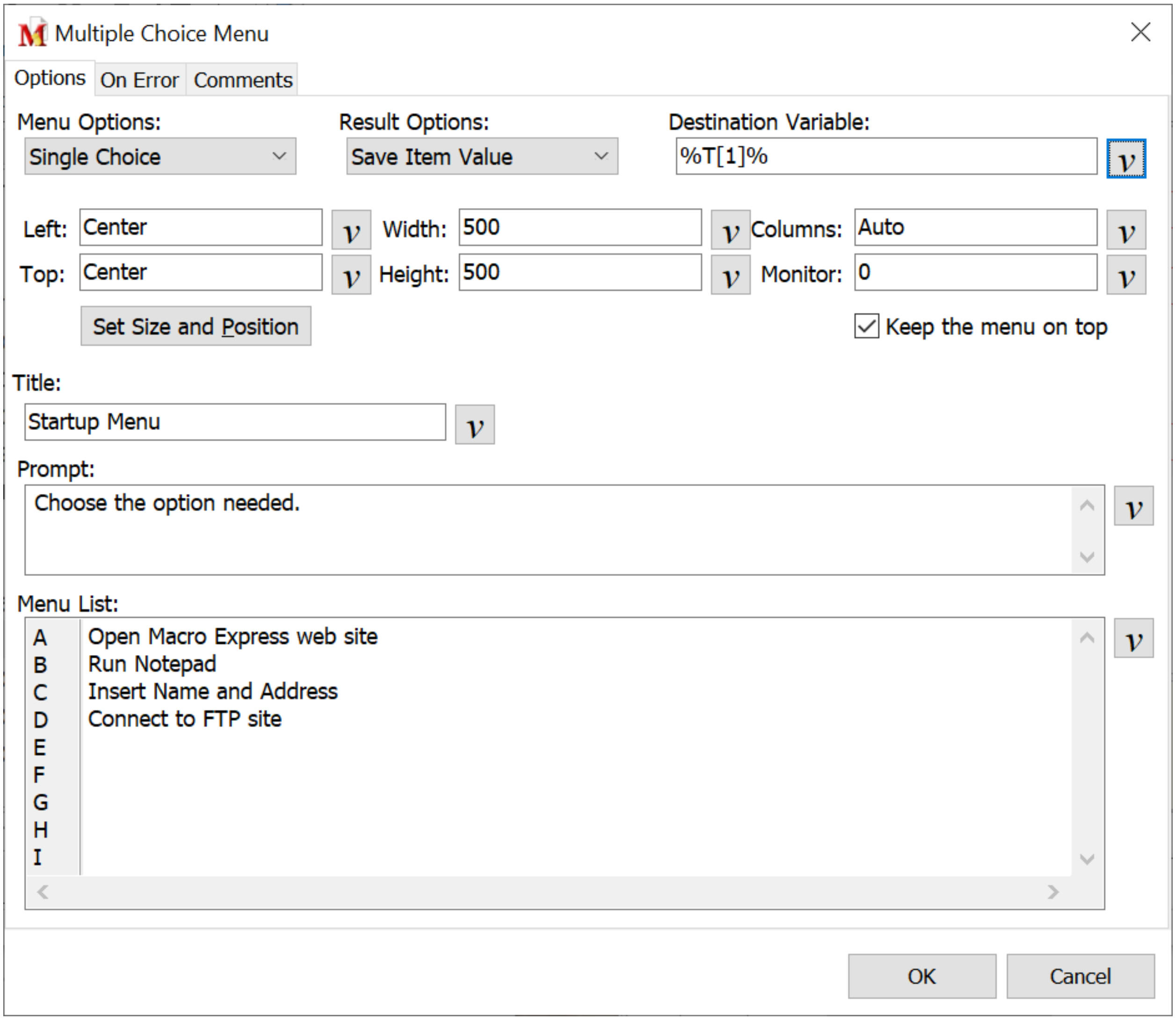


Time to purchase a copy of Macro Express Pro.

Multiple Choice Menu

The Multiple Choice Menu command displays a menu of options during macro playback. Choices can be displayed as either radio boxes (one selection out of the group) or as check boxes (select as many as wanted). The selection results are stored in a text string variable which can be tested for conditional processing of the macro.

- ✓ **Note:** To create Yes/No, True/False, or OK/Cancel type menu options, use the If Message command.



Menu Options: Single Choice

Result Options: Save Item Value

Destination Variable: %T[1]%

Left: Center

Top: Center

Width: 500

Height: 500

Columns: Auto

Monitor: 0

☑ Keep the menu on top

Title: Startup Menu

Prompt: Choose the option needed.

Menu List:

A	Open Macro Express web site
B	Run Notepad
C	Insert Name and Address
D	Connect to FTP site
E	
F	
G	
H	
I	

Script Editor > Expand Dialogs Category > Multiple Choice Menu

Menu Options

Choose whether to allow the user to choose only one valid selection or multiple selections.

Single Choice

With this option selected the user may only select one choice from the displayed list.

Sample Macro

This sample macro is based on the screen shot of the Create a Multiple Choice Menu window displayed in this topic. Essentially the macro will display a menu of four choices.

Select the desired option and click OK. The macro then continues based on the choice made. For example if A is selected, then the default web browser would load the Macro Express home page.

Macro Script

```
Multiple Choice Menu: Prompt
If Variable %T[1]% Equals A
  Web Site, "https://www.macros.com", using Default Web Browser
End If
If Variable %T[1]% Equals B
  Launch Program and Activate Window: Program "NOTEPAD", Window "notepad"
End If
If Variable %T[1]% Equals C
  Text Type (Simulate Keystrokes): Insight Software Solutions, Inc.
End If
If Variable %T[1]% Equals D
  FTP Site Connect to www/wintools.com
End If
```

Multiple Choice

The Multiple Choice option allows the user to select more than one choice from the displayed list. The correct logic must be inserted into the macro so that it can correctly handle more than one selection. A sample of how this would be done is shown below.

Sample Macro

```
Multiple Choice Menu: Prompt
If Variable %T[1]% Contains A
  And
If Variable %T[1]% Contains B
  Insert commands to perform
End If
If Variable %T[1]% Contains A
  And
If Variable %T[1]% Contains C
   Insert commands to perform
End If
If Variable %T[1]% Contains A
  And
If Variable %T[1]% Contains D
   Insert commands to perform
End If
If Variable %T[1]% Contains B
  And
If Variable %T[1]% Contains C
  Insert commands to perform
End If
If Variable %T[1]% Contains B
  And
If Variable %T[1]% Contains D
   Insert commands to perform
End If
If Variable %T[1]% Contains C
  And
If Variable %T[1]% Contains D
  Insert commands to perform
End If
```

Result Options

Save Item Text

Selecting this item requires entering the menu list name used when writing the macro. Based on the example above, use If Variable T[1] = Go to Macro Express web site, If Variable T[2] = Run Notepad, etc. to create the rest of the Multiple Choice macro.

Save Item Value

Selecting this item requires entering the letter associated with the option. Based on the example above, use If Variable T[1] = A, If Variable T[2] = B, etc. to create the rest of the Multiple Choice macro.

- ✓ **Note:** Number of Menu Items Allowed
  The number of menu choices permitted varies depending on which combination of Single/Multiple Choice and Item Value/Text is selected. Below are the various combinations and the possible number of menu items allowed.

- ✓ **1. Single Choice / Item Value** - No limit to the number of items in menu.
  2. Single Choice / Item Text - No limit to the number of items in menu.
  3. Multiple Choice / Item Value - Menu limited to 36 items.
  4. Multiple Choice / Item Text - No limit to the number of items in menu.

Destination Variable

This is the variable that receives the result of the user choices from the menu. The "If Variable" command is used to test the value of this variable to determine what choice has been selected by the user. The variable T[1] was selected in the window displayed above.

To ensure the menu functions correctly use the command "If Variable T[1] Equals A" to see if the user chose the first item in the Menu List. (Use "If Variable T[1] Equals B" to test for the second item in the Menu List, and so on.)

Position and Size the Multiple Choice Window

Choose from the options below to size, position and format the window.

Left/Top

Position the upper left hand corner of the Multiple Choice window "x" number of pixels from the top and left edges of the screen. Enter the pixel values to position the window.

Width/Height

Enter the pixel values to manually set the size of the Multiple Choice window. For example, entering a width of 500 and a height of 600 would generate a window size of 500 x 600 pixels. Insert the word Auto in the fields to have the window auto-sized, based on the contents of the menu.

Columns

Choose the number of columns to display in the Multiple Choice window. The default setting is Auto, which adds a second column when more than 10 items are listed in the menu.

Monitor

This option applies to systems that utilize more than one monitor. If a system has a single monitor, then leave the monitor setting to zero (0). Zero represents the single monitor or the left monitor if there is more than one monitor. One (1) represents a second monitor or the monitor to the right of the left monitor. Similarly, two (2) represents the third monitor or two monitors to the right of the left monitor, and so on.

Set Position Manually

Use this option to manually position the Multiple Choice window to the desired location. Also manually set the size of the Multiple Choice window. The Window Position and Size topic explains these options in detail.

Keep the Menu on Top

This option keeps the Multiple Choice menu visible on the screen (on top of other running applications) until the menu is closed.

Title

The Title contains the text that appears in the caption bar of the Window displaying the menu.

Prompt

The Prompt contains the text that appears right above the menu choices. This text should help the user understand why they are being prompted to make a choice. For example, if the menu contains a list of income ranges, the prompt could simply consist of something like "Please choose your income bracket".

Menu List

The Menu List contains the list of choices for the menu. Each corresponding letter (A, B, C...) represents a different choice in the menu. They are displayed in the order shown on this screen. If only 3 choices are entered (i.e. an entry for A, B and C), then the menu only shows the 3 entries.

Do not leave gaps in the entries or the menu will end up with blank items. In other words, use the first 3, 5 or however many entries needed. For example, don't just create entries for items A, D, E and G or the menu will display blank items for items B, C and F.

- ✓ **Note:** Text Variables may be placed in the Title, Description/Prompt and Menu List input fields. For example, place the variable T[1] (you would actually enter %T[1]%) in the "Description/Prompt" edit box.

-- Review the Multiple Choice Menu with Defaults sample macro in the samples.mex file, which is supplied with the program, for a working example of the Multiple Choice Menu. --

Multiple Choice Tips

Make one of the Multiple Choice selections the Default

Let's say you want option B to be your default selection. Use the Variable Set String command and select the Set Value Now option. Set the value of variable T[1] to B. Insert this command before your Menu command in the macro script. When the macro runs, option B will display as the default selection.

Allow users to select a menu item with the keyboard instead of clicking with the mouse

Let's look at the Multiple Choice Menu image above. There are four entries in the Menu List. We want to set this up so that when the menu is displayed, the user can press a keystroke rather than click on their choice with the mouse. To do so we would modify the Menu List to look like the following:

A	&Go to Macro Express web site
B	&Run Notepad
C	&Insert email tagline
D	&Connect to FTP site

You'll notice that an ampersand "&" has been placed in front of each item in the menu list. When the list is displayed during playback of the macro, the letter following the ampersand "&" will be underlined (set as the accelerator key). The user can then press the underlined letter for the choice they want. This will highlight their choice. Pressing the Enter key will perform the macro selection chosen.

The ampersand may be placed anywhere in the text string. It doesn't need to be placed before the first letter as in the example above. For example, we could set item A to read as follows: Go to &Macro Express web site. In this case the M is underlined and pressing the "M" key during playback will highlight this menu option.

Note 1: Be careful not to duplicate the underlined letters. Otherwise the first duplicate letter will be the default choice each time the letter is pressed.

Text Box Display

Use the Text Box Display feature to pop up text windows during macro playback. There are a number of possible uses for this command of which a few are listed below.

Reminders

Set up reminder messages on a monthly, weekly, daily or a one time occurrence. Create reminders to pop up on screen to make an important phone call at a specific time or to pick up a few items at the store after work. Create the macro as a scheduled macro to play back at the appointed time.

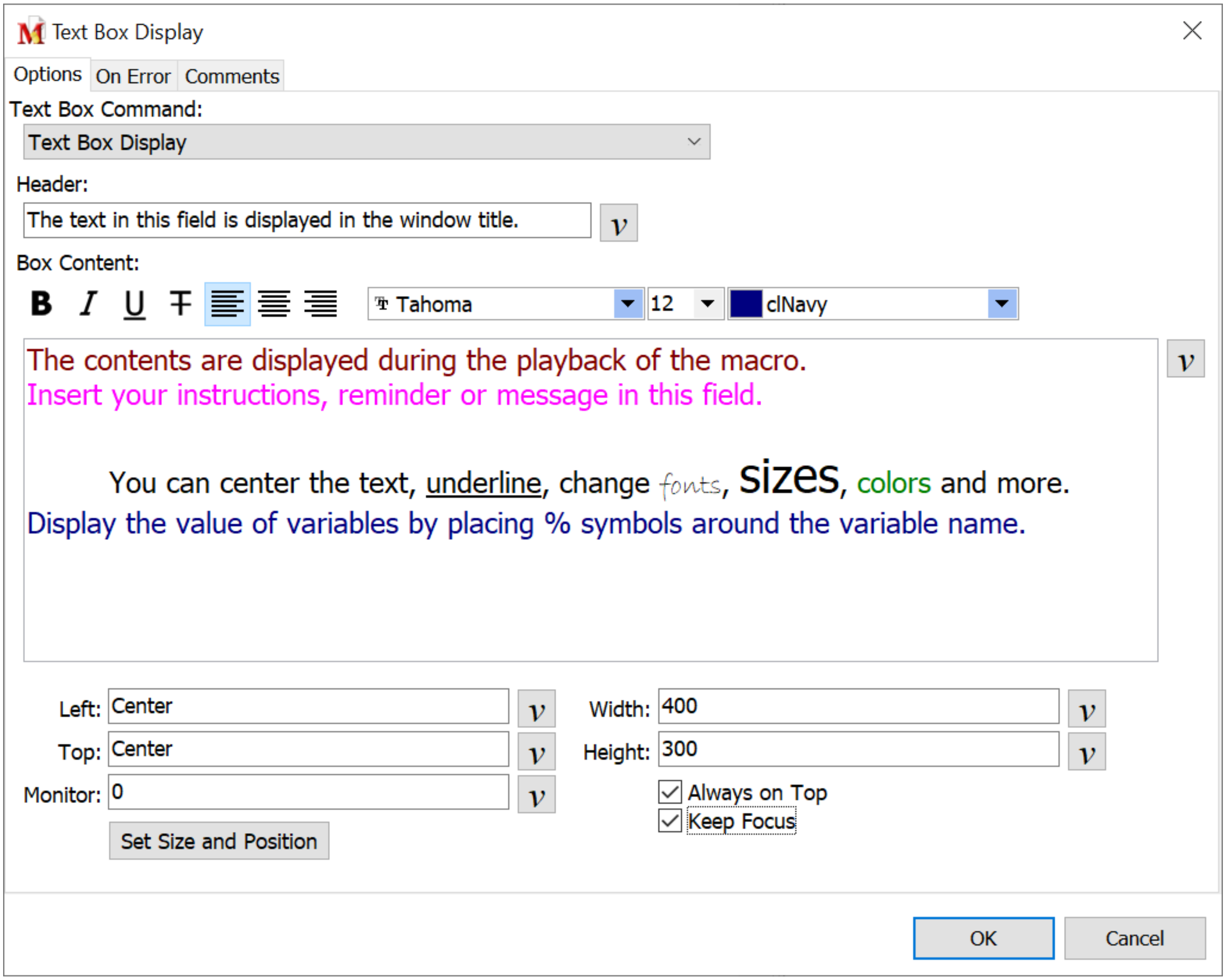
Display Instructions

Use the Text Box Display to provide instructions for a user to follow during macro runtime.

Debugging

This command is also useful for debugging macros, such as displaying the contents of variables. Make sure the values match what they are expected to be.

For example, to display the contents of T[1], insert %T[1]% into the Box Content field. When playing back the macro, the value for T[1] displays in the text box during macro run time and shows exactly what is being saved to the variable.



Text Box Command: Text Box Display

Header: The text in this field is displayed in the window title.

Box Content: Tahoma 12 clNavy

The contents are displayed during the playback of the macro.
Insert your instructions, reminder or message in this field.

You can center the text, underline, change fonts, sizes, colors and more.
Display the value of variables by placing % symbols around the variable name.

Left: Center
Top: Center
Monitor: 0
Width: 400
Height: 300
☑ Always on Top
☑ Keep Focus

Script Editor > Expand Dialogs Category > Text Box Display

Text Box Command

Select one of the four text box options available from the drop down list.

Text Box Display

The Text Box Display option pauses the macro. The user is required to click on the "OK" button to close the text box and allow the macro to continue.

Timed floating box or end of macro

The execution of the macro continues when this option is selected. The text box remains floating on top until either the specified time passes or the macro finishes. The Timed floating box does not stop or delay the macro. All the Floating Text Boxes are closed when the macro finishes. Add a Delay or Timed Delay command after the Text Box command in order to keep the box open for a specific amount of time.

Stay on top for _____ seconds

Specify the amount of time to display the text box by inserting the number of seconds in this field. The field is large enough that the box can stay open for hours. However, the box closes if the macro finishes before the number of seconds selected elapses.

Floating Box until "Close Box" command or end of macro

With this option checked the execution of the macro continues. The text box remains on top until either the macro terminates, a "Close Text Box" command inserted in the macro is reached or the box is manually closed.

Text Box Close

The Text Box Close command closes an existing text box that is being displayed. The Header must match exactly (uppercase/lowercase sensitive) in order for the text box to close. If no match is found on the header, the macro halts by default, unless one of the other On Error options is selected.

- ✓ **Note:** All floating text boxes are closed at the end of the macro run.

Header

Use this option to create a title for the display window. The Header appears in the window caption bar and is required in order to use the Close Text Box option.

Box Content

The Box Content field provides space to place a more detailed message or instructions. This field contains the message displayed during playback of the macro.

Rich Text

The edit box supports rich text. Use this option to create messages with different fonts, sizes, colors, etc.

Display text with bold, underline, italics and strike through features.

Left, center or right justify the text.

Use different font colors, sizes or font types.

Colors such as clWindowText, clWindowFrame, clScrollBar are assigned to specific parts of the text box window. For example, on different computers clWindowText may be be black, dark gray or white depending on the theme or custom color choices.

Always on Top

With the Always on Top option selected the text box remains visible on the screen (on top of other running applications) until the box is closed.

Keep Focus

Select this option to give focus to the Text Box Display window. If not selected, focus returns to the application that had focus before the display window opened.

Size and Position

These options position and size the Text Box Display window on the screen. The Window Position and Size topic explains these options in detail.

Quick Wizard Option

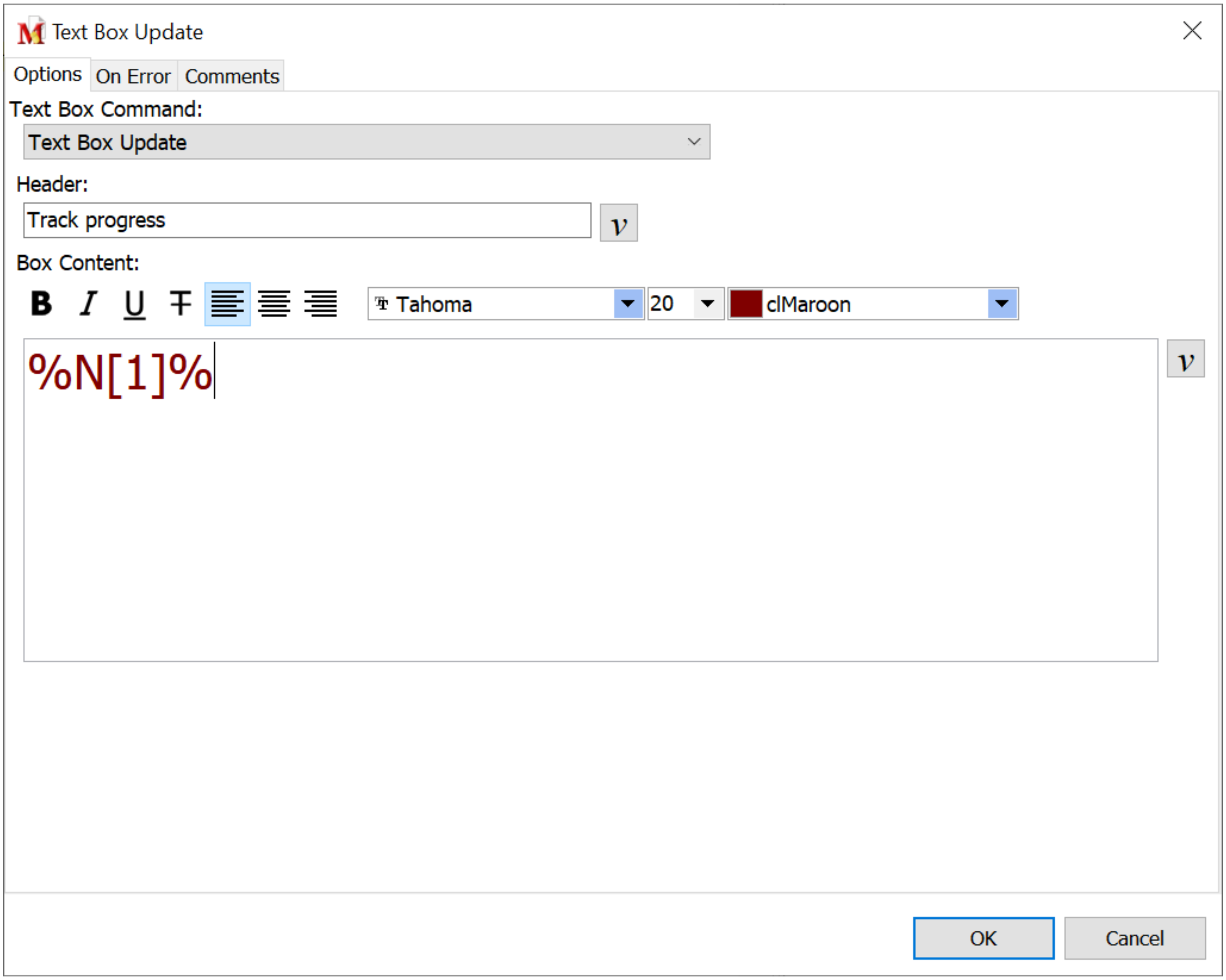
Using the Quick Wizard for creating a reminder provides a couple of additional options. If creating a one-time reminder, there is an option to automatically delete the macro after playback. Also choose to tack on a beep or a sound file to the macro to alert when a message is displayed. The scheduler is built into the wizard to make the whole process quick and easy.

Sounds may also be added to reminder macros when not using the wizard. Use the Sound File command to inert a sound into the macro script.

Text Box Update

Use this command to update a Text Box being displayed in the macro with a new value, further instructions, etc. Also use this function to display and update the number of passes through a Repeat Loop or a Text File Process.

- ✓ **Note:** The Update Textbox command is to **only** be used with the **Timed floating box or end of macro** or **Floating Box until "Close Box" command or end of macro** options.



Text Box Command: Text Box Update

Header: Track progress

Box Content:

%N[1]%

Script Editor > Expand Dialogs Category > Update Textbox

Header

Insert the name of the Text Box to be updated during the playback of the macro. The Header name must match exactly with the header of the Text Box previously defined in the macro script.

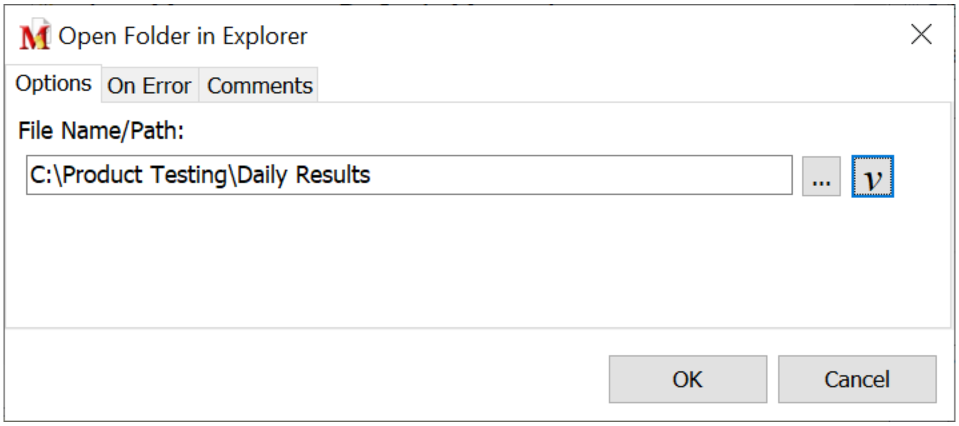
Box Content

Enter the information to be displayed or updated in the Text Box into the Box Content field. This text will replace any information that was previously displayed in the Text Box.

Explorer

Open Folder in Explorer

This option provides a shortcut to specific folders found in Windows Explorer. Set up a hot key or other trigger to instantly go several layers deep in Windows Explorer.



File Name/Path: C:\Product Testing\Daily Results

Script Editor > Expand Explorer Category > Open Folder in Explorer

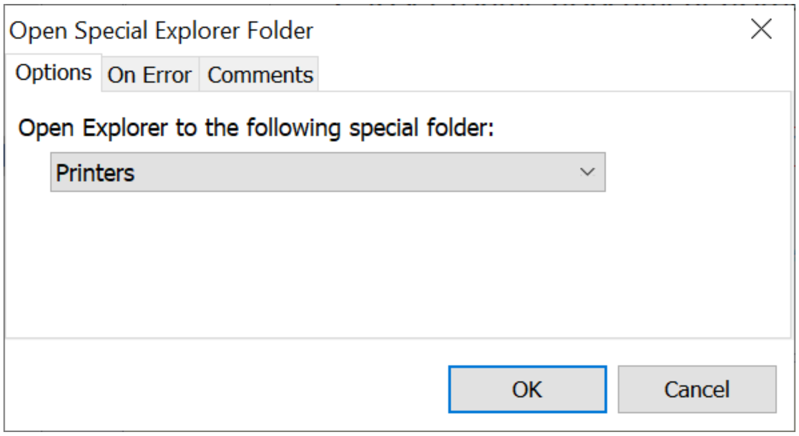
One example might be to create a shortcut to the Startup folder to periodically make adjustments. Or create a shortcut to the Cookies folder to see how it has grown after a day of Internet usage.

Another option would be to monitor a Temporary folder full of old downloads or email attachments that might require a periodic cleaning. In short, this option provides a quick way to get to the folders most frequently accessed.

Enter a folder name and the full path or click on the Browse button to search for the desired folder. Click the OK button when finished to save.

Open Special Explorer Folder

This command provides a quick way to get to some of the most frequented folders or dialog boxes. Other Explorer folders not listed below can be directly accessed with a macro using the Open Folder or Open Folder in Explorer command.



Open Explorer to the following special folder: Printers

Script Editor > Expand Explorer Category > Open Special Explorer Folder

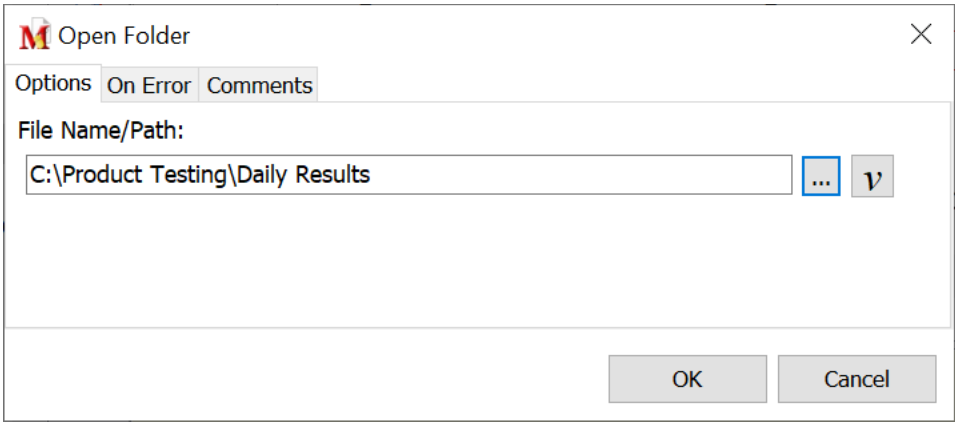
Use this option to quickly open any of the fourteen Explorer folders listed below. Select the folder and click OK to save.

Available Folders

Control Panel
Desktop Folder
Desktop Programs
Favorites
Fonts
My Computer
Network Neighborhood
Printers
Programs Folder
Recent Documents
Recycle Bin
Start Menu
Startup Folder
User Documents

Open Folder

This command is similar to the Open Folder in Explorer command. "Open Folder" opens a window with only the contents of a folder displayed. "Open Folder in Explorer" opens a window with two panes that displays the Folder tree view on the left and the contents of the selected folder on the right.



File Name/Path:
C:\Product Testing\Daily Results

Script Editor > Expand Explorer Category > Open Folder

The Open Folder command provides a shortcut to folders on the computer. Set up a hot key or other trigger to instantly go several layers deep in the File/Folder structure.

One example might be to create a shortcut to the Startup folder to periodically make adjustments. Or create a shortcut to the Cookies folder to see how it has grown after a day of Internet usage.

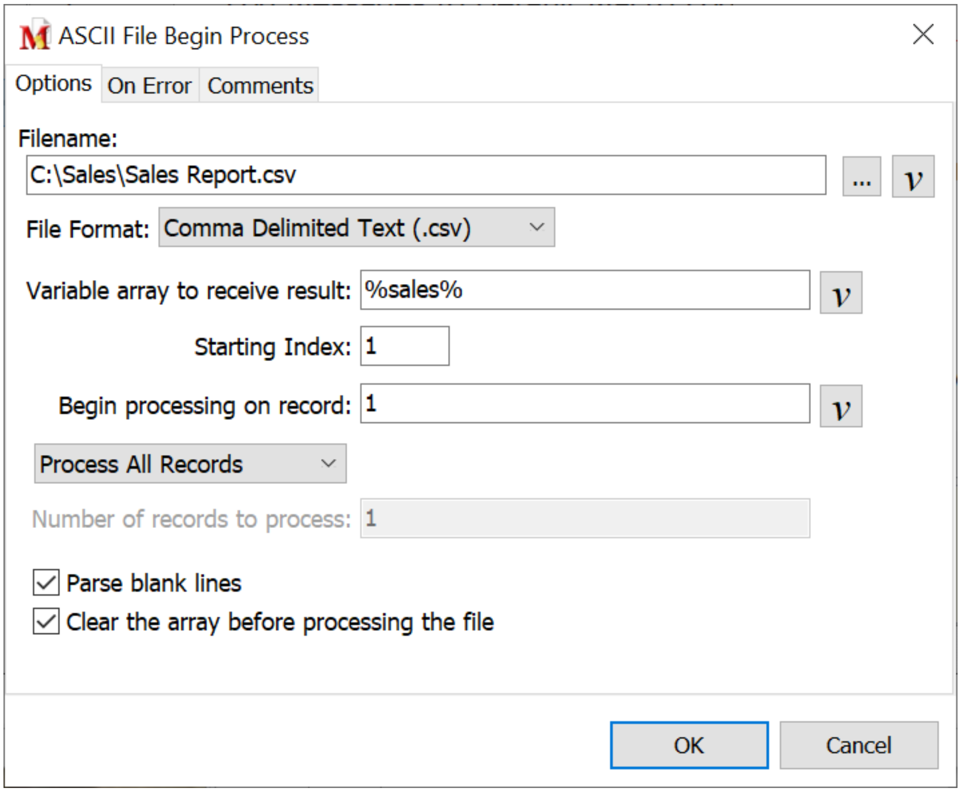
Another option would be to monitor a Temporary folder full of old downloads or email attachments that might require a periodic cleaning. In short, this option provides a quick way to get to the folders most frequently accessed.

Enter a folder name and the full path or click on the Browse button to search for the desired folder. Click the OK button when finished to save.

ASCII File Process

The ASCII File Process is a powerful command that extracts data from text files, one line at a time. The values of each record are saved to variables for later use, such as inserting the extracted values into other applications. This is an efficient method for transferring data from one file to another.

- **Cautionary Note:** Do not manually edit or have another program edit the file during the processing, as this may cause unpredictable results on the computer. If you need to edit or update the file while the ASCII File Begin Process command is running, then have the macro make a copy of the file and process the file copy.



ASCII File Begin Process

Filename: C:\Sales\Sales Report.csv

File Format: Comma Delimited Text (.csv)

Variable array to receive result: %sales%

Starting Index: 1

Begin processing on record: 1

Process All Records

Number of records to process: 1

☑ Parse blank lines

☑ Clear the array before processing the file

Script Editor > Expand Files or Folders Category > ASCII File Process Command

File Name

Enter the name of the file to be opened and processed in this field. Click on the Browse button to locate the file and path on the computer.

File Format

The ASCII File Process processes three commonly used Delimited Text file formats. Most spreadsheet and database programs can export information in at least one of these file formats.

Comma Delimited Text

In a Comma Delimited Text file every field in each record is separated by a comma. Any field that contains a comma must be surrounded by a double quote (").

Comma Delimited Text file sample

```
84031,UT,HANNA,DUCHESNE,435,UTC -7:00,Y
84032,UT,HEBER CITY,WASATCH,435,UTC -7:00,Y
84033,UT,HENEFER,SUMMIT,435,UTC -7:00,Y
84034,UT,IBAPAH,TOOELE,435,UTC -7:00,Y
84035,UT,JENSEN,UINTAH,435,UTC -7:00,Y
84036,UT,KAMAS,SUMMIT,435,UTC -7:00,Y
84037,UT,KAYSVILLE,DAVIS,801/385,UTC -7:00,Y
84037,UT,FRUIT HEIGHTS,DAVIS,801/385,UTC -7:00,Y
84038,UT,LAKETOWN,RICH,435,UTC -7:00,Y
84039,UT,LAPOINT,UINTAH,435,UTC -7:00,Y
84040,UT,LAYTON,DAVIS,801/385,UTC -7:00,Y
```

Tab Delimited Text

In a Tab Delimited Text file every field in each record is separated by a Tab character. For visual effect, <tab> is displayed in the example to represent where a Tab character should be placed.

Tab Delimited Text file sample

```
84031<tab>UT<tab>HANNA<tab>DUCHESNE<tab>435<tab>UTC
-7:00<tab>Y
84032<tab>UT<tab>HEBER CITY<tab>WASATCH<tab>435<tab>UTC
-7:00<tab>Y
84033<tab>UT<tab>HENEFER<tab>SUMMIT<tab>435<tab>UTC
-7:00<tab>Y
84034<tab>UT<tab>IBAPAH<tab>TOOELE<tab>435<tab>UTC -7:00<tab>Y
84035<tab>UT<tab>JENSEN<tab>UINTAH<tab>435<tab>UTC -7:00<tab>Y
84036<tab>UT<tab>KAMAS<tab>SUMMIT<tab>435<tab>UTC -7:00<tab>Y
84037<tab>UT<tab>KAYSVILLE<tab>DAVIS<tab>801/385<tab>UTC
-7:00<tab>Y
84037<tab>UT<tab>FRUIT HEIGHTS<tab>DAVIS<tab>801/385<tab>UTC
-7:00<tab>Y
84038<tab>UT<tab>LAKETOWN<tab>RICH<tab>435<tab>UTC -7:00<tab>Y
84039<tab>UT<tab>LAPOINT<tab>UINTAH<tab>435<tab>UTC
-7:00<tab>Y
84040<tab>UT<tab>LAYTON<tab>DAVIS<tab>801/385<tab>UTC
-7:00<tab>Y
```

ASCII Delimited Text

In an ASCII Delimited Text file every field in each record is surrounded by double quote marks (") and separated by a comma. If the text contains a double quote it must use two double quotes in a row or it will be interpreted as the end of the field.

ASCII Delimited Text file sample

```
"84031","UT","HANNA","DUCHESNE","435","UTC -7:00","Y"
"84032","UT","HEBER CITY","WASATCH","435","UTC -7:00","Y"
"84033","UT","HENEFER","SUMMIT","435","UTC -7:00","Y"
"84034","UT","IBAPAH","TOOELE","435","UTC -7:00","Y"
"84035","UT","JENSEN","UINTAH","435","UTC -7:00","Y"
"84036","UT","KAMAS","SUMMIT","435","UTC -7:00","Y"
"84037","UT","KAYSVILLE","DAVIS","801/385","UTC -7:00","Y"
"84037","UT","FRUIT HEIGHTS","DAVIS","801/385","UTC -7:00","Y"
"84038","UT","LAKETOWN","RICH","435","UTC -7:00","Y"
"84039","UT","LAPOINT","UINTAH","435","UTC -7:00","Y"
"84040","UT","LAYTON","DAVIS","801/385","UTC -7:00","Y"
```

Each field in a record (a record is a line of the text file) is loaded into a variable which can be used throughout the macro. Optionally specify from which line to start in the text file and to either process all the records from that point or just a specific number of records. Use of the variables and processing will take place between the command to open the file and the command to end the file processing.

Example

This example uses any one of the example data files shown above and the following macro:

```
ASCII File Begin Process: "test.txt" (Tab Delimited Text )
    Text Type (Simulate Keystrokes): %T[1]%
      %T[2]%
      %T[3]%
      %T[4]%
      %T[5]%
      %T[6]%
      %T[7]%
ASCII File End Process
```

For this example, T is entered in the field labeled "Variable Array to Receive Result" in the ASCII File Begin Process command. The value of 1 is entered as the Starting Index.

The first line of the file is processed with the following variable values:

```
 T[1] = 84031
 T[2] = UT
 T[3] = HANNA
 T[4] = DUCHESNE
 T[5] = 435
 T[6] = UTC - 7:00
 T[7] = Y
```

Next, the second line of the file is processed with the following variable values:

```
 T[1] = 84032
 T[2] = UT
 T[3] = HEBER CITY
 T[4] = WASATCH
 T[5] = 435
 T[6] = UTC - 7:00
 T[7] = Y
```

The third line of the file is processed with the following variable values:

```
 T[1] = 84033
 T[2] = UT
 T[3] = HENEFER
 T[4] = SUMMIT
```

Variable Array to Receive Result

This is the text variable array that will receive the first field in a record (or line of text). Sequential text variables will receive subsequent field numbers. For example, if T is selected as the starting text variable, the first field in the record is saved to T[1], the second field in the record is saved to T[2] and so on.

Or instead of using T as the variable, assign a name to the variable such as "sales" and create as an array.

These variables may then be used and processed by other macro commands. For example, to type out the first two records into another application, use the Text Type command. Insert the variables to play back. In this case insert %sales[1]% %sales[2]%. This plays back the values saved to the first two variables.

Starting Index

The Starting Index defines which element in the array to start with. In the example above the variables used in the Process an ASCII Text File would start with T[1]. If the starting index is set to 5 then the first variable used would be T[5].

If other variables in the macro were previously assigned to T[1], T[2], etc. and will be used later, then start the index with a higher value in order to not overwrite the values saved to these variables.

- ✓ **Note:** The "T" variable is automatically set up as an array. The variables are assigned as T[1], T[2], T[3], etc. There is not a limit as to the number of elements in the array. If using a named variable, first create the variable, assign it as an array and select the number of elements needed. Array elements start with [1] and not [0].

Begin Processing on Record ____

Use this option to select which record to begin processing in the file. To process the entire file, enter 1 in the value field to indicate starting with the first record. Or use an Integer variable when specifying which record to start processing and the number of records to process.

What To Process

Process All Records

Select this option to process every record in the file, if electing to start with record #1. Or it will start with the record selected and process all subsequent records.

Process Partial Records

Choose the "Process Partial Records if only needing to process a certain range of records (or lines of text).

Number of Records to Process

Enter in this field the number of records to process.

Parse Blank Lines

Use this option to process any blank lines in the file. The variable values will be empty when the blank line of the file is processed.

Leave this option unchecked so that the macro skips any blank lines in the file.

Clear the Array before Processing the File

Select this option to clear the values of all elements in the array selected to receive the results before the file processing begins. In the example above, the T variable array is selected. If a value were previously set to the array element T[10] before the processing begins, this value would be cleared when the file processing starts.

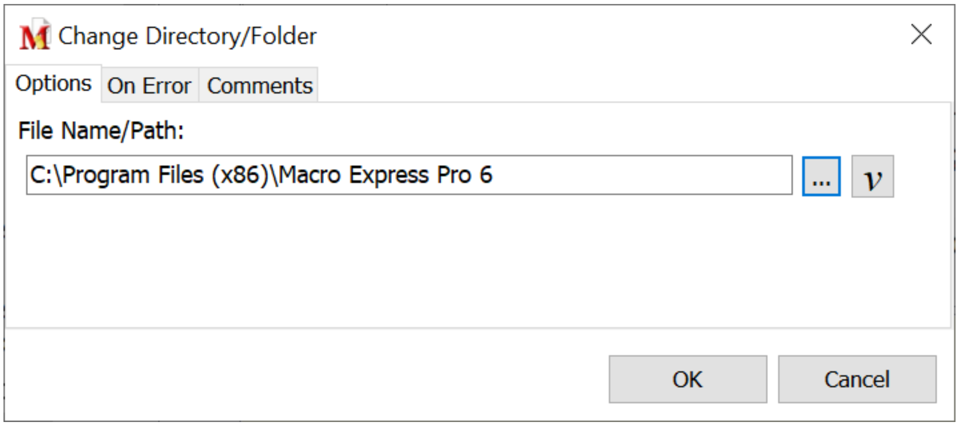
ASCII File End Process

Each ASCII File Begin Process command requires a corresponding ASCII File End Process command. All commands in the script between the Begin and the End Process commands will be repeated the number of times instructed.

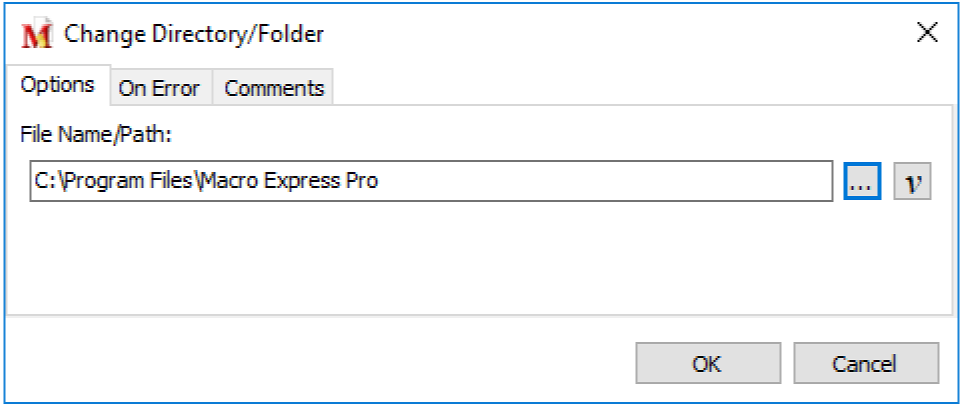
If all records have been read out of the text file and processed, then control goes to the macro command that immediately follows the ASCII File End Process command.

Change Directory/Folder

Some programs require a specific default folder to be set before they are run. The Change Directory/Folder allows for this situation.



File Name/Path: C:\Program Files (x86)\Macro Express Pro 6



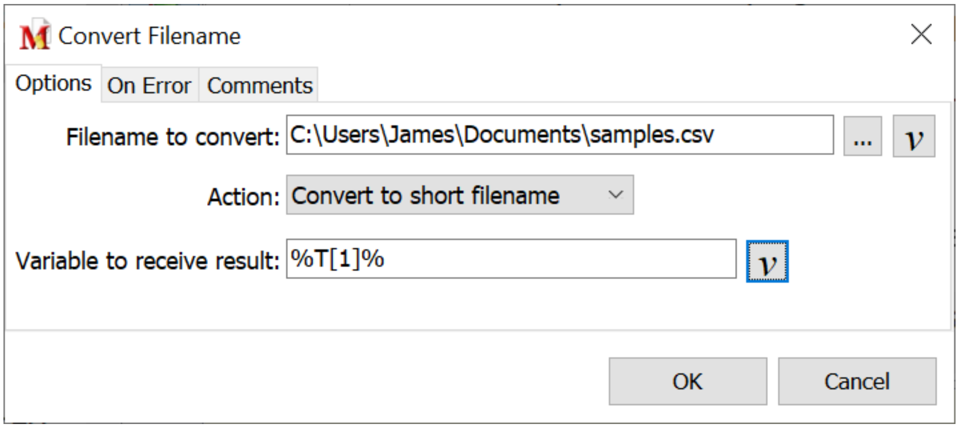
File Name/Path: C:\Program Files\Macro Express Pro

Script Editor > Expand Files or Folders Category > Change Directory/Folder

Enter the name of the path and folder or use the Browse button to locate a specific folder. Click OK to save the setting.

Convert File Name

Some programs or systems may display file names and paths in a truncated format, shortening any names in the path to eight characters or less. Select the file and path and choose whether to convert it to a long or short file name.



Filename to convert: C:\Users\James\Documents\samples.csv

Action: Convert to short filename

Variable to receive result: %T[1]%

Script Editor > Expand Files or Folders Category > Convert File Name

File Name to Convert

Select the file name to be converted. The folder and entire path must be entered. Or click on the Browse button to navigate to the desired file. The file name must be a valid file in order for the conversion to work properly.

Action

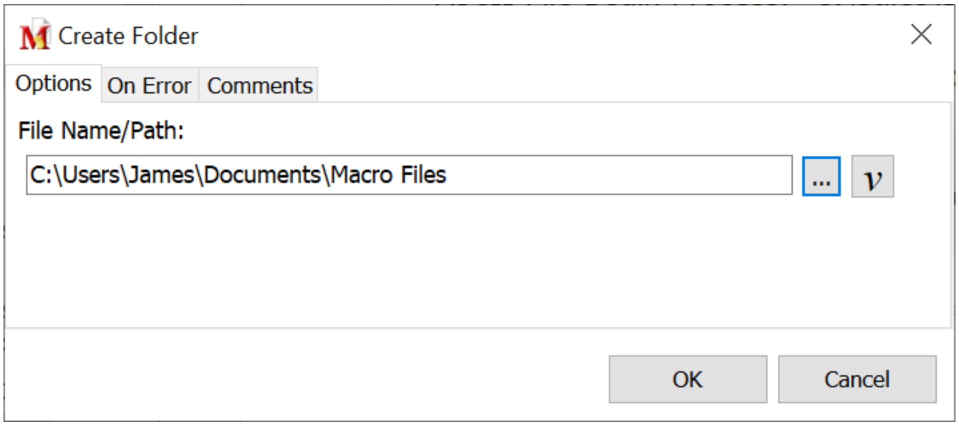
Select from the drop down list whether to convert the file to a short name or a long name.

Variable to Receive Result

The converted file name is saved to a Text String variable for later use in the macro.

Create Folder

Use this option to create a new folder anywhere on the computer.



File Name/Path: C:\Users\James\Documents\Macro Files

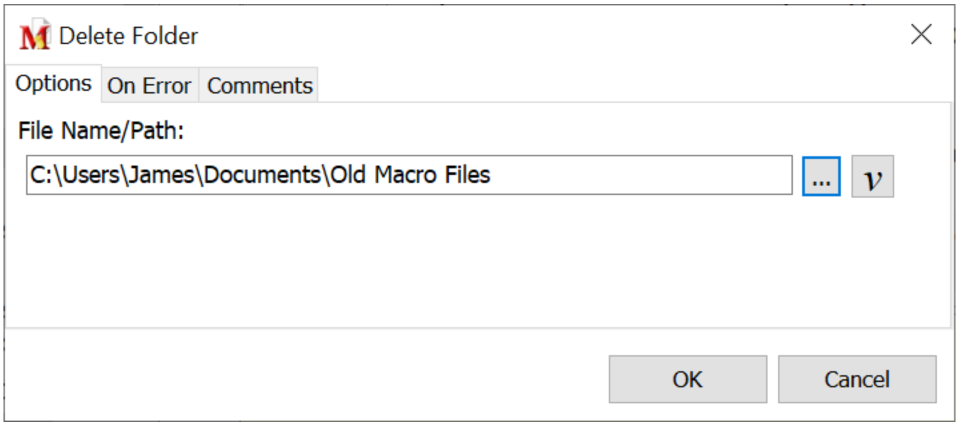
Script Editor > Expand Files or Folders Category > Create Folder

File Name/Path

Insert the new folder name and path. Or click on the Browse button to fill in the path information and insert the new folder name.

Delete Folder

Use this option to delete a folder and the contents of that folder from the computer.



File Name/Path: C:\Users\James\Documents\Old Macro Files

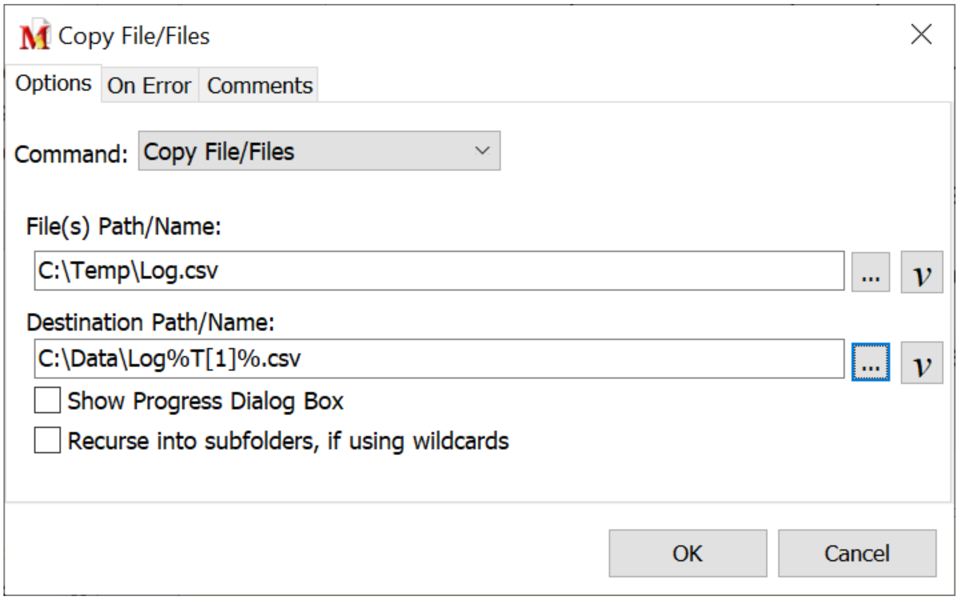
Script Editor > Expand Files or Folders Category > Delete Folder

File Name/Path:

Insert the folder name to be deleted and path. Or click on the Browse button to select the folder to be deleted.

File Manipulation

Use the file commands to change, create or delete folders as well as copy, rename, delete or move files. Select the desired function from the Command drop down list. Note: Wild cards may be used in the copy, rename, move and delete "File" commands. Use *.* to perform actions on all files in the folder, *.doc for all .doc files, fa.* for all files that start with an f, etc.



Command: Copy File/Files

File(s) Path/Name: C:\Temp\Log.csv

Destination Path/Name: C:\Data\Log%T[1]%.csv

□ Show Progress Dialog Box

□ Recurse into subfolders, if using wildcards

Script Editor > Expand Files or Folders Category > Click on one of commands discussed in this topic.

Delete File or Files

This option deletes the selected file or files. It can be particularly helpful when creating temporary files and deleting them when finished.

File(s) Path/Name

Insert the file name and path or click on the Browse button to locate the file to be deleted.

Delete Files Permanently

Deletes the file so that it bypasses the Windows Recycle Bin. Once deleted it is not recoverable through the Recycle Bin.

Copy File or Files

Make copies of existing files on the computer.

File(s) Path/Name

Insert the file name and path or click on the Browse button to locate the file to be copied.

Destination Path/Name

Insert the new file name to assign to the copy. Optionally click on the Browse button to copy over top of an existing file or to insert the file path and edit the file name.

- ✓ **Note:** This function may also be used for copying folders and the contents of the folder. Insert the path and the name of the folder in the edit field. Or use the browse button to select a file in the folder to be copied. Remove the file name from the edit field so that just the path and folder name remain.

Rename File or Files

Rename existing files on the computer.

File(s) Path/Name

Insert the file name and path or click on the Browse button to locate the file to be renamed.

Destination Path/Name

Insert the name to assign to the renamed file. Or click on the Browse button to insert the file path and edit the file name.

Move File or Files

Move files on the computer to a new location.

File(s) Path/Name

Insert the file name and path or click on the Browse button to locate the file to be moved.

Destination Path/Name

Insert the name to assign to the moved file. Or click on the Browse button to insert the file path and edit the file name.

- ✓ **Note:** This function may also be used for moving folders and the contents of the folder. Insert the path and the name of the folder in the edit field. Or use the browse button to select a file in the folder to be moved. Remove the file name from the edit field so that just the path and folder name remain.

  If creating a new folder - the Destination Path/Name, the original folder will be deleted and all of the contents moved to the new folder. If moving the files to an existing folder, the original folder is not deleted but the files are moved into the existing folder.

Rename Folder

Use this option to rename an existing folder.

Old Filename

Insert the folder name and path or click on the Browse button to locate the folder to be renamed.

New Filename

Insert the name you want to give the folder. Or click on the Browse button to insert the file path and edit the folder name.

Show Progress Dialog Box

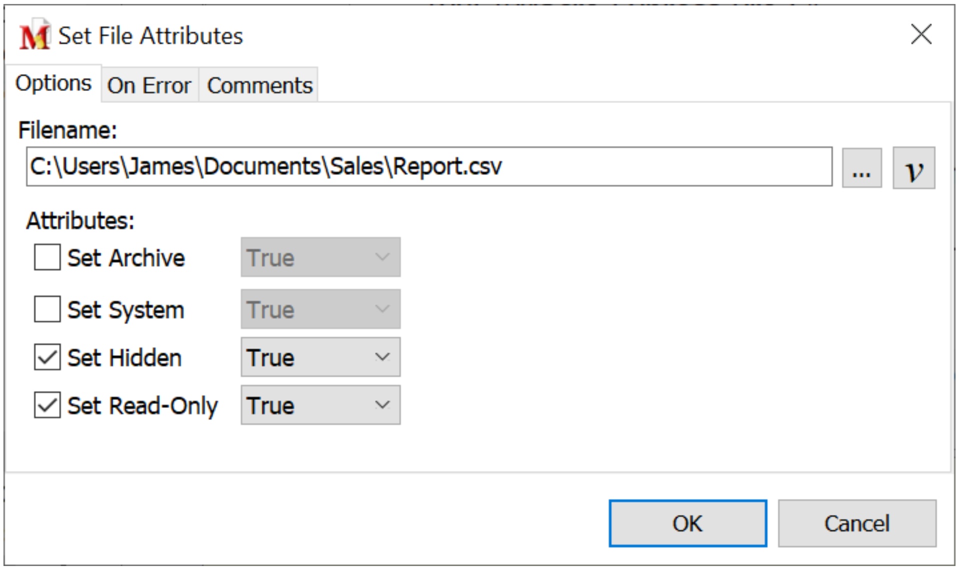
If this is checked, then a standard Windows dialog box opens displaying each file being copied. If the action takes only a very short time to perform, then the dialog box does not appear.

Recurse into subfolders, if using wildcards

This allows action to be taken on any folder found within the folder selected, if that folder matches the wildcard. For example, if using *.*, a match occurs on every folder name. If the command is to copy *.* and this option is selected, then all subfolders and their contents will be copied as well.

Set File Attributes

When using My Computer or Windows Explorer, right click on a file name to open the menu. Left mouse click on the Properties menu item to see more details about the file. At the bottom of the Properties window the following File Attributes may be displayed - Archive, System, Hidden and Read Only.



Filename:
C:\Users\James\Documents\Sales\Report.csv

Attributes:

- □ Set Archive True
- □ Set System True
- ☑ Set Hidden True
- ☑ Set Read-Only True

Script Editor > Expand Files or Folders Category > Set File Attributes

The Set File Attributes command changes the file attributes of files without having to navigate through My Computer or Windows Explorer.

Filename

Insert the file name and path or click on the Browse button to locate the file to be changed.

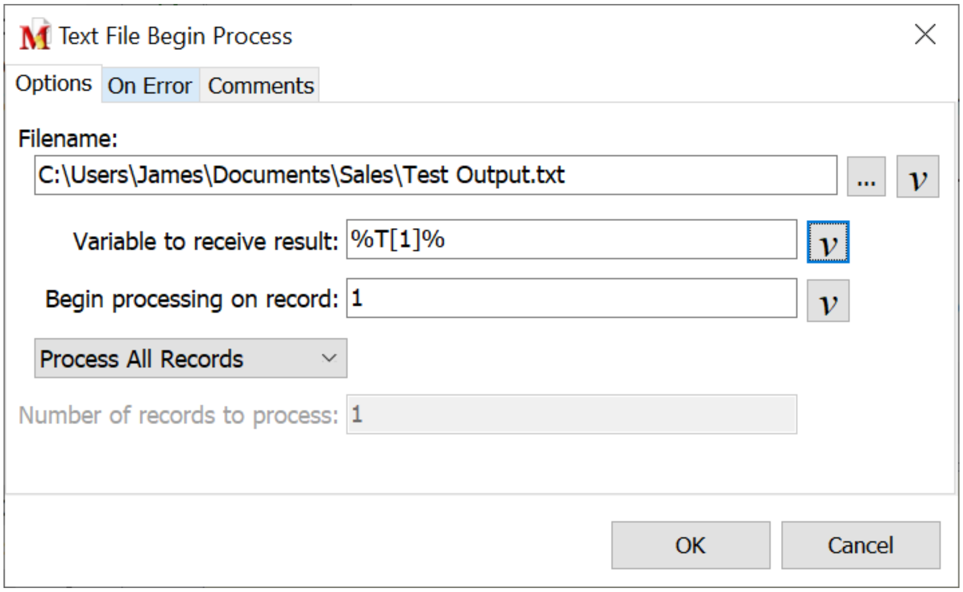
Attributes

Select the attribute or attributes to be turned on or off. True activates the attribute. False removes the attribute from the file.

✓ **Note:** The If File Attribute command is available to first test the state of that file.

Text File Process

The Text File Process command reads each line of a text file into a variable. The entire line of text is saved into a single variable.



Filename: C:\Users\James\Documents\Sales\Test Output.txt

Variable to receive result: %T[1]%

Begin processing on record: 1

Process All Records

Number of records to process: 1

Script Editor > Expand Files or Folders Category > Text File Process

Each line of the text file is loaded into a variable which can be used later in the macro. Select which line of the text file to begin with and either process all the lines from that point or just process a specific number of lines. Use of the variable and processing take place between the command to open the text file and the command to end the text file processing.

- **Cautionary Note:** Do not manually edit or have another program edit the file during the processing, as this may cause unpredictable results on the computer. If you need to edit or update the file while the Text File Begin Process command is running, then have the macro make a copy of the file and process the file copy.

Example

If a text file contains the following data:

```
FMC,"98 SE, 2000",,Talon,750 MHz,256 B,,83
H,95 B,5.00.3314.2101,P II,333 MHz,65 B
H,"98 SE, 2000",5.00.3314.2101,P III,550 MHz,384 B,,,"17""
Flat"
```

and variable T[1] is used as the "Variable to Receive First Field".

The first time through the Process loop the variables will be:
  T[1] = FMC,"98 SE, 2000",,Talon,750 MHz,256 B,,83

The second time through the Process loop the variables will be:
  T[1] = H,95 B,5.00.3314.2101,P II,333 MHz,65 B

The third time through the Process loop the variables will be:
  T[1] = H,"98 SE, 2000",5.00.3314.2101,P III,550 MHz,384 B,,,"17"" Flat"

Filename

Insert the name of the text file to be opened and processed. Click on the Browse button to locate the file and path or enter manually.

Variable to Receive Result

This is the text variable that receives the value of the text line in the file. This variable may then be used and processed by other macro commands.

Begin Processing on Record ____

Use this option to select which record to begin processing in the file. To process the entire file, enter 1 in the value field to indicate starting with the first record. Or use an Integer variable when specifying which record to start processing and the number of records to process.

What to Process

Process All Records

Select this option to process every record in the file, if electing to start with record #1. Or it will start with the record selected and process all subsequent records.

Process Partial Records

Choose the "Process Partial Records if only needing to process a certain range of records (or lines of text).

Number of Records to Process

Enter in this field the number of records to process.

Text File End Process

Each Text File Begin Process command requires a corresponding Text File End Process command. All commands in the script between the Begin and the End Process commands will be repeated the number of times instructed.

- ✓ **Note:** If all records have been read out of the text file and processed, then control goes to the macro command that immediately follows the Text File End Process command.

Related Topics

Email Send

Sending messages by email can be automated and activated by the touch of a key. Send to one or multiple recipients, carbon copy (cc), and blind carbon copy (bcc). Select recipients at macro design time or at macro run time. A built in email address manager maintains a list of frequent recipients.

Generate the email message contents at macro design time, when the macro runs, from the contents of the clipboard or from a text file. One or multiple attachments may be included with the email. Attachments may also be specified at macro design time or at macro play back time.

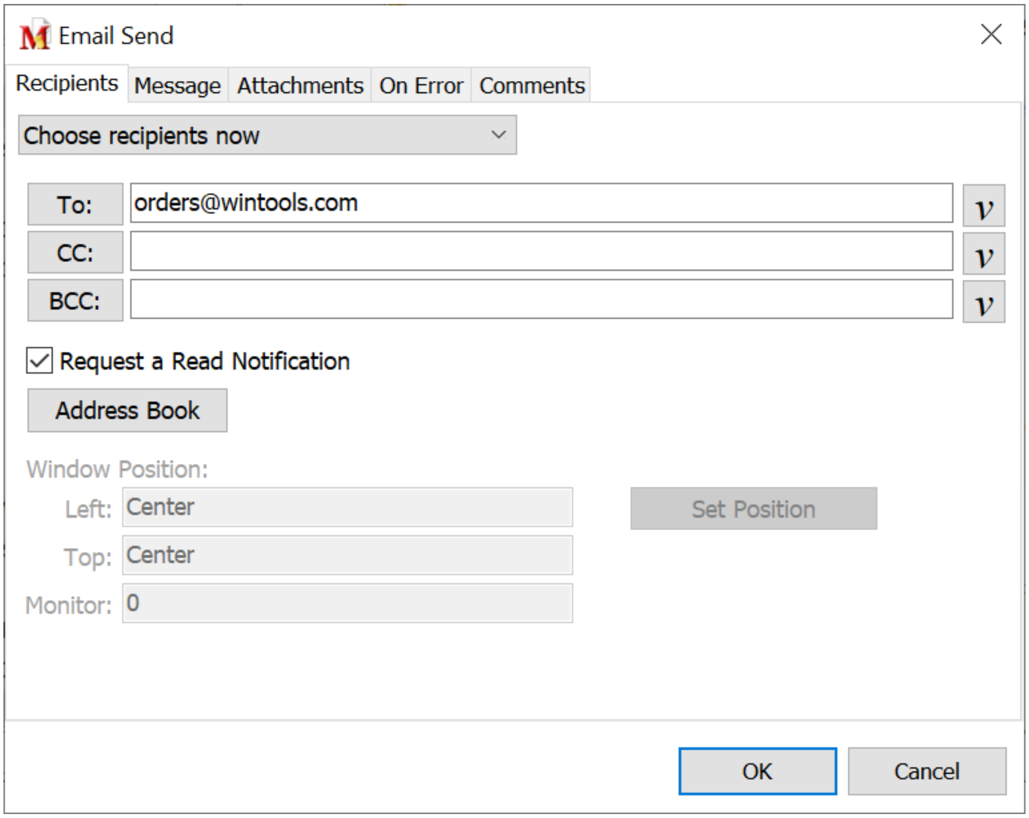
Optionally maintain or browse a log of all email messages sent. The log file path is set from Options > Preferences > File System > File Paths.

- ✓ **Note:** In order for the Email Send command to work properly first set up the default email information. Click on Options > Preferences > Email to enable the email functions of Macro Express.

The Email Send command is divided into three main sections - Recipients, Message and Attachments. These are each discussed below.

Recipients

The first step is to determine whether to hard code the recipients into the macro or choose the recipients at macro runtime. Select the option preferred from the drop down list.



Choose recipients now

To: orders@wintools.com

CC:

BCC:

☑ Request a Read Notification

Left: Center

Top: Center

Monitor: 0

Script Editor > Expand Internet Category > Email Send > Recipients Tab

Choose Recipients Now

Select the name of the person(s) to receive the email.

To: Click on the To button and select the person to receive the email from the Address book that opens.

CC: Click on the CC button and select a person from the address book to receive a carbon copy of the message.

BCC: Click on the BCC button and select a person from the address book to receive a blind carbon copy of the message.

Request a Read Notification

Select this option to have a small dialog box open during macro playback notifying that the email was sent successfully.

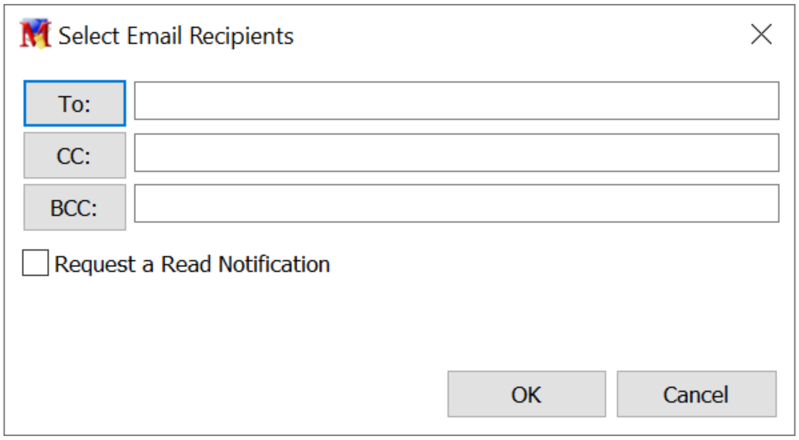
Address Book

Click on the Address button to add, remove or modify any names on the email recipient list. Or edit the Address Book from Options > Preferences > Email.

More than one person (address) may be inserted in any of these fields.

Prompt for Recipients at Runtime

The Prompt for Recipients at Runtime option displays a window similar to the one shown below during playback of the macro. Enter the email address or addresses of those to receive the email in the appropriate fields. Or click on the To, CC or BCC buttons to insert an address from the Address Book.

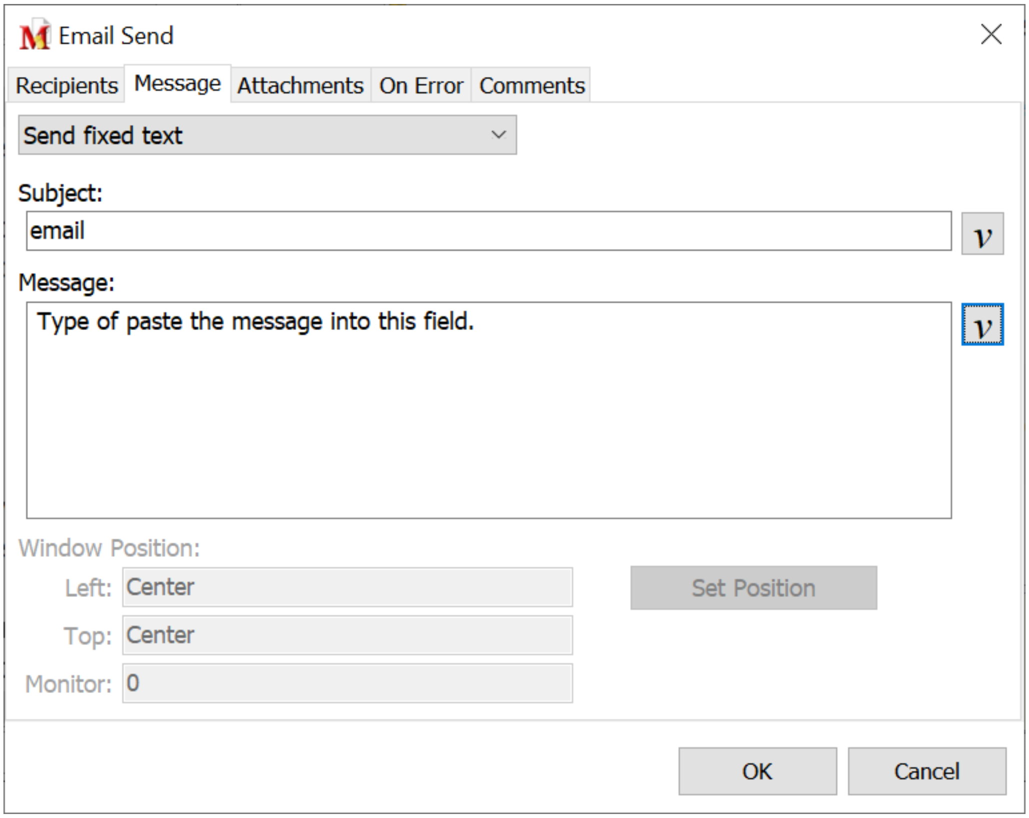


Request a Read Notification

Select this option to have a small dialog box open during macro playback signaling that the email was sent successfully.

Message

The four message options available are explained below. Select the option preferred from the drop down list.



Send fixed text

Subject: email

Message: Type of paste the message into this field.

Window Position:

Left: Center

Top: Center

Monitor: 0

Script Editor > Expand Internet Category > Email Send > Message Tab

Fixed Text

The fixed text option sends the same fixed text email message each time the macro is invoked. Enter the message subject in the appropriate field and the email message into the Message edit box.

Prompt for Subject & Text

This option prompts during playback of the macro to include a text message. An edit window opens prompting to enter a subject heading and message. Click on the OK button to send the email or Cancel to not send the message.

Window Position

Use this option to specify where the prompt window should display. See the Window Position and Size topic for complete details on using this option.

Send Text From Clipboard

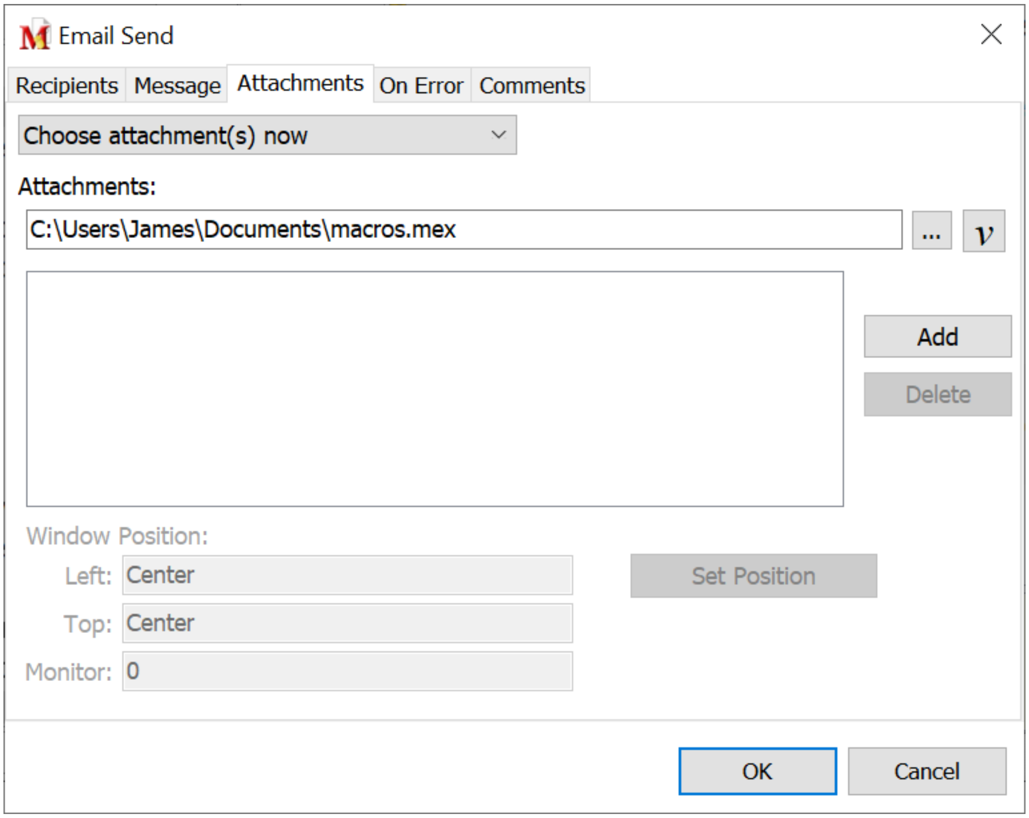
Use this option to send whatever is currently saved to the clipboard in an email message. Copy something to the clipboard before running the macro or have the macro make the clipboard copy and then send the email. The clipboard contents are sent via email to the designated recipients).

Send Text From File

This option sends the contents of the selected text file. Click on the Browse button to locate the text file to send or enter the file name and path in the File field.

Attachments

This tab provides three options for sending attachments with the email message. Select the option preferred from the drop down list.



Email Send

Attachments: C:\Users\James\Documents\macros.mex

Window Position:

Left: Center

Top: Center

Monitor: 0

Script Editor > Expand Internet Category > Email Send > Attachments Tab

No Attachment

Click on this option to not include an email attachment with the message.

Choose Attachment Now

Use this option to select the email attachment during macro design time. Click on the Browse button to locate and select the attachment file. Then click on the Add button to add to the list of attachments. Variables may also be used as part of the attachment file name.

This option allows for multiple attachments. After selecting the first attachment and it is placed in the list, start again to add another file. If a mistake is made in selecting the wrong attachment, highlight it from the list and click on the Delete button to remove the attachment.

Prompt For Attachment

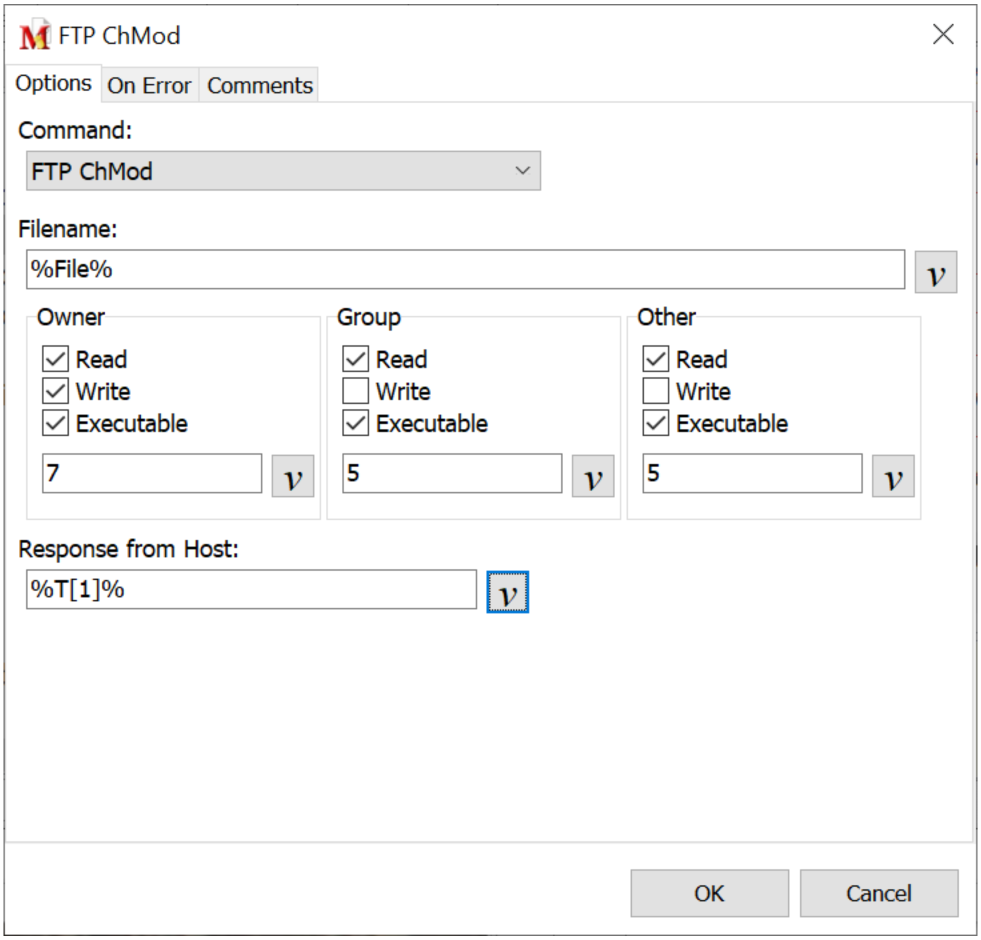
This option prompts to select an attachment during macro playback. A Select Attachments window opens to browse the computer to find the correct attachment.

This option also allows for multiple attachments. After selecting first attachment and it is placed in the list, click on the Add button again to add another. If a mistake is made in selecting the wrong attachment, highlight it from the list and click on the Delete button to remove the attachment.

Once attachments are added click on the OK button to send the email.

FTP ChMod

This command assumes familiarity with operating FTP servers and clients.



Command: FTP ChMod

Filename: %File%

Owner
☑ Read
☑ Write
☑ Executable
7

Group
☑ Read
□ Write
☑ Executable
5

Other
☑ Read
□ Write
☑ Executable
5

Response from Host: %T[1]%

Script Editor > Expand Internet Category > FTP ChMod

Filename

Select the filename to be edited.

You, Group, Other

Select the attributes for the file selected. Select file rights for yourself, your group and all others. Elect to make the file readable so that any of these three may have read rights to the file. Also elect to make this file writable, so that it may be edited by one or any of these three. Or make certain files executable so that others may be able to access the file and run it.

Place checks in the boxes or assign resulting numbers that would apply. When selecting the box, a number is automatically assigned below.

Log Results To

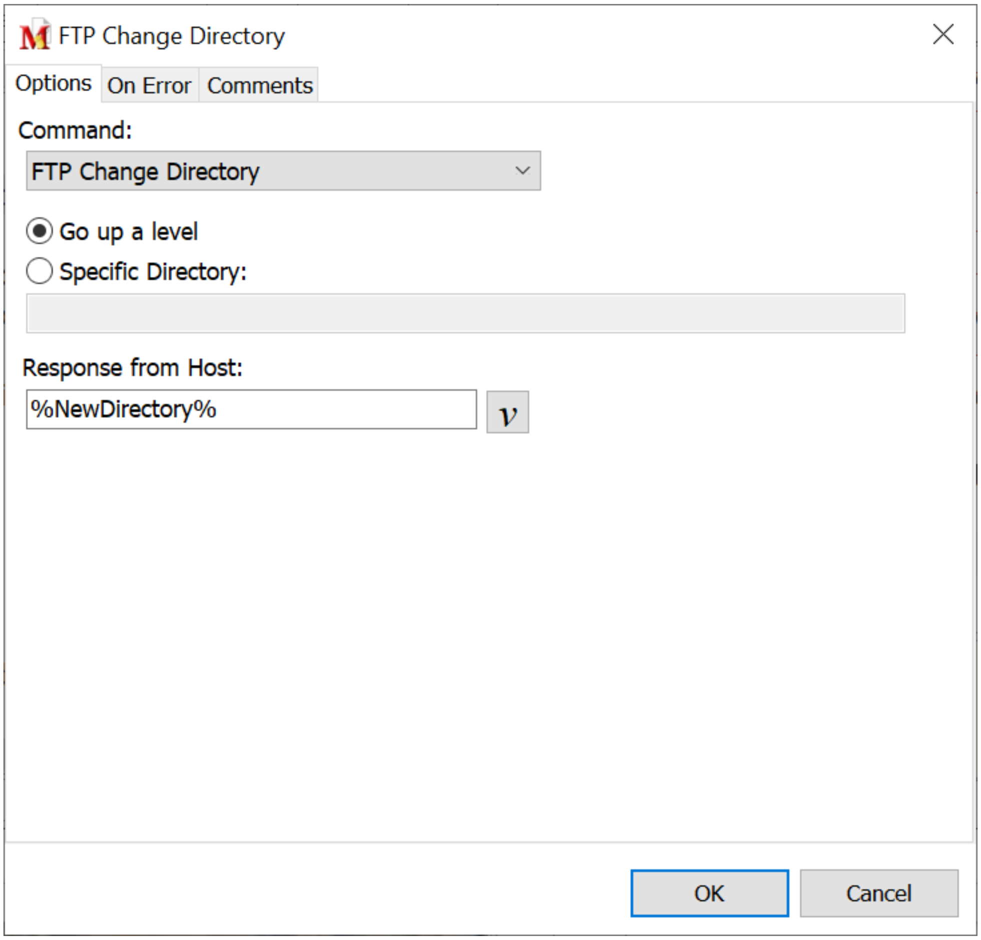
The message generated by the FTP server, whether a success or failure message, is saved to the variable assigned. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email.

✓ **Note:** Before using this function, the macro script first requires the FTP Site Connect command to connect to the site. When all FTP actions have been completed, use the FTP Site Disconnect command to disconnect from the site.

FTP Directory Commands

The following commands all affect the FTP directory.



Command: FTP Change Directory

● Go up a level

○ Specific Directory:

Response from Host: %NewDirectory%

Script Editor > Expand Internet Category > FTP Directory Commands

FTP Change Directory

The FTP Change Directory command changes the current working directory on the FTP site. The user can select a specific directory to change. This can be an absolute path or a relative path. This functionality is identical to the Change Directory/Folder command that applies to folders on a computer drive. However, this command also adds the ability to go up a level and switch to the previous directory in the hierarchy.

- ✓ **Note:** Path delimiters in an FTP server require the forward slash '/', not the backslash '\' that Windows uses.

Go Up a Level

This options moves up one folder in the FTP server folder hierarchy.

Specific Directory

This option moves to a specific folder in the FTP server.

FTP Delete Directory

This command deletes a directory from the FTP server.

Directory

Insert the directory to be deleted from the FTP server.

FTP Create Directory

This command creates a new directory on the FTP Server.

Directory

Insert the new directory name to be created.

Log Results To

This option is available for the FTP Change, Delete and Create Directory commands. The message generated by the FTP server, whether a success or failure, is saved to the variable assigned. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

FTP Get Current Directory

The FTP Get Directory command retrieves the present working directory on the FTP Server. This information is saved to the variable selected. If for some reason the retrieval was not successful the error message information is saved to the variable.

FTP List Directory

This command retrieves a listing of all files in the current directory on the FTP Server.

File Mask

Specifies the files being looked for in the directory. Wildcards are permitted.

Detailed List

This returns all values about the files in the current directory, such as file size, permissions and the owner.

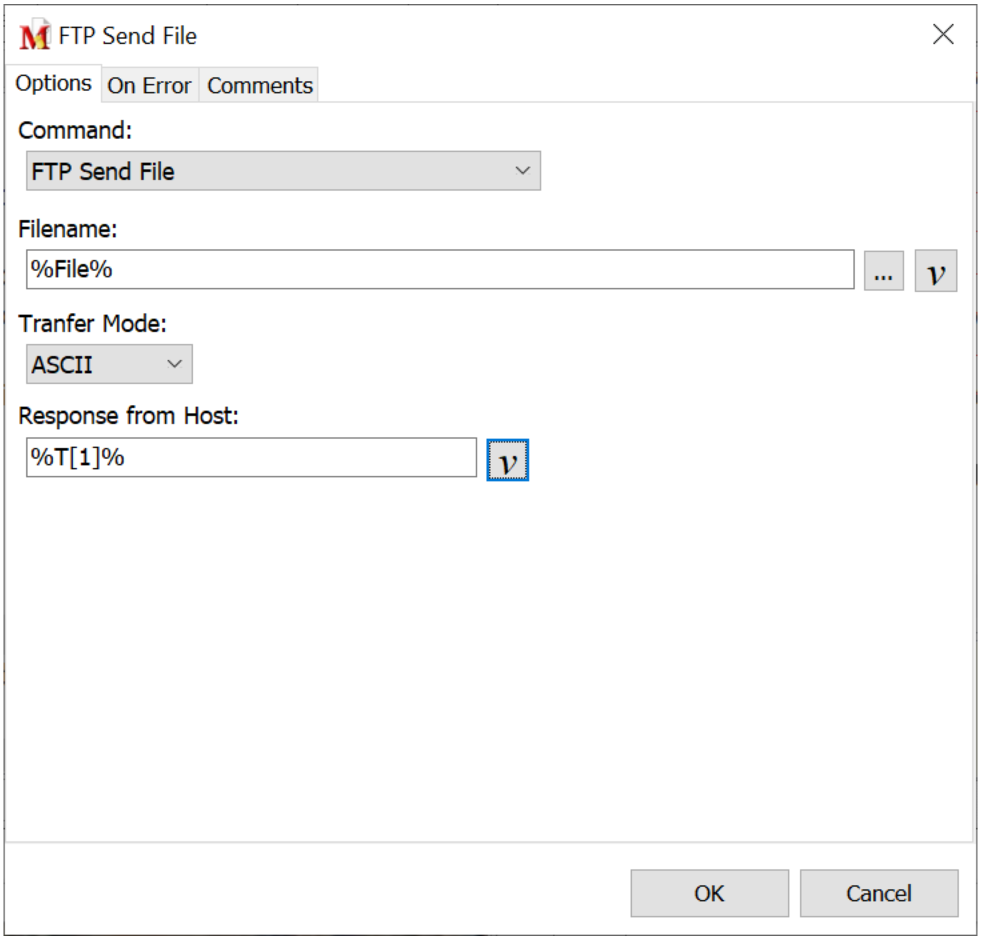
Variable to Save Result to

The information retrieved is stored in the variable selected.

- ✓ **Note:** Before using any of these functions, the macro script first requires the FTP Site Connect command to connect to the site. When all FTP actions have been completed, use the FTP Site Disconnect command to disconnect from the site.

FTP File Commands

Each of these commands pertain to files that are transferred via FTP.



Command: FTP Send File

Filename: %File%

Tranfer Mode: ASCII

Response from Host: %T[1]%

Script Editor > Expand Internet Category >Select one of FTP File commands listed in this topic

FTP Send File

Sends the selected file from the user's computer to the FTP server.

File Name

Insert the name of the file to send or click on Browse to locate the file.

Transfer Mode

Determine whether the file being sent or received is a Binary file or an ASCII file.

Log Result To

The message generated by the FTP server, whether a success or failure, is saved to the assigned variable. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

FTP Get File

Retrieves the selected file from the FTP server to the destination folder on the user's computer.

Remote File Name

Enter the name of the file found on the FTP server.

Local Destination Directory

Enter the folder name and path of the file to be saved on the computer. Or click on the Browse button to locate the directory.

Transfer Mode

Determine if the file being sent or received is a Binary file or an ASCII file.

Log Results To

The message generated by the FTP server, whether a success or failure, is saved to the assigned variable. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

✓ **Note:** The Scroll Lock + Pause or other methods available to stop the macro do not work during the playback of the FTP Send File and FTP Get File commands.

FTP Rename File

This command renames a FTP file.

Original Filename

Insert the name or path and name of the file to be renamed.

New Filename

Insert the new filename or new path and name of the file.

Variable to Receive Server Result

The message generated by the FTP server, whether a success or failure, is saved to the assigned variable. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

FTP Delete File

This command deletes the selected FTP file. The FTP Delete File command also accepts wildcards, such as *.*, in the filename edit box.

File Name

Enter the name of the file to be deleted from the FTP server.

Log Server Response to

The message generated by the FTP server, whether a success or failure, is saved to the assigned variable. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

Get Filesize

This command returns the size of a FTP file.

- ✓ **Note:** A return value of -1 means that the FTP server does not support the Size command.

Filename

To obtain the file size, first Insert the name or path and name of the file in question.

Destination Variable

The file size is stored in an integer variable. Select the variable to store this information.

Transfer Mode

Select whether the FTP file is a Binary file, an ASCII file or unknown. Binary is the default selection.

Store the Server Response Code In

The message generated by the FTP server, whether a success or failure, is saved to the assigned variable. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

✓ **Note:** Before using any of these functions, the macro script first requires the FTP Site Connect command to connect to the site. When all FTP actions have been completed, use the FTP Site Disconnect command to disconnect from the site.

FTP Keep Alive

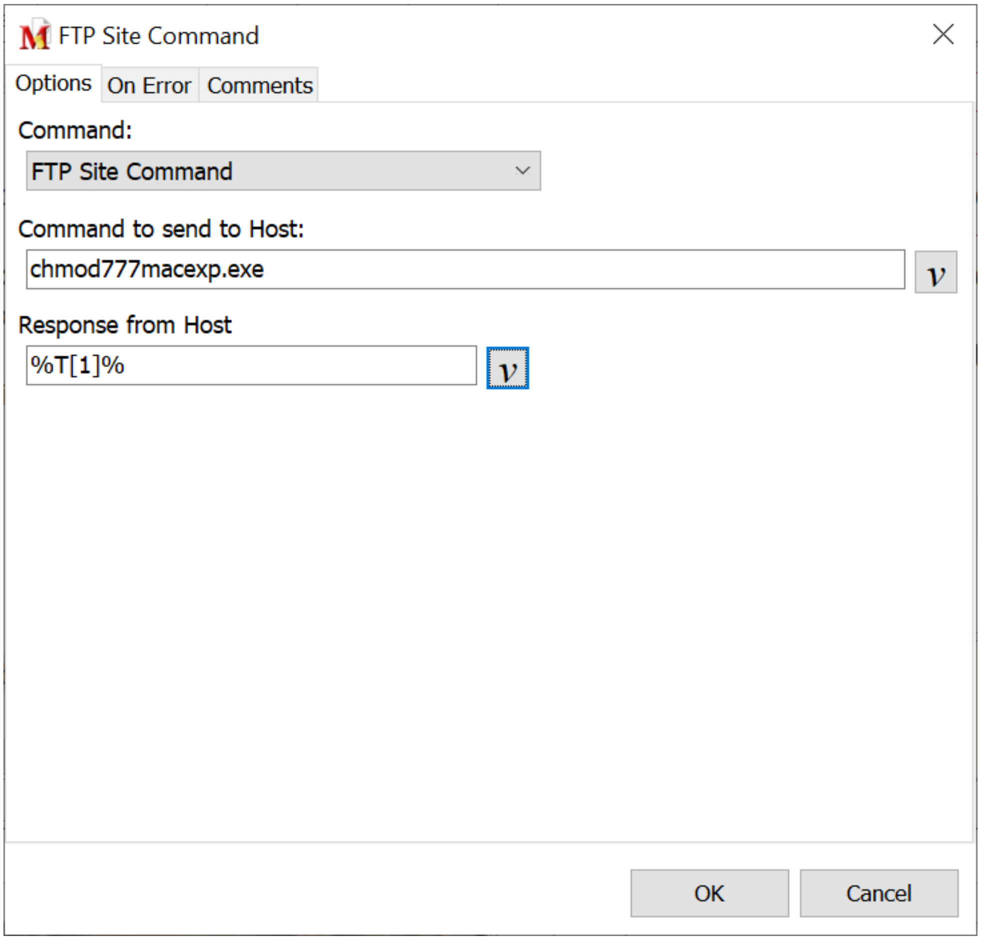
The FTP Keep Alive command sends a signal during macro playback to keep the FTP connection alive.

There is no window for this command. From the list of commands in the Script Editor or Direct Editor click on the Internet folder to expand the list of commands. Double click on the FTP Keep Alive command to insert into the macro.

- ✓ **Note:** Before using this function, the macro script first requires the FTP Site Connect command to connect to the site. When all FTP actions have been completed, use the FTP Site Disconnect command to disconnect from the site.

FTP Site Command

This command assumes familiarity with operating FTP servers and clients. This option sends a FTP command, one that we otherwise wouldn't support, to the FTP server.



Command: FTP Site Command

Command to send to Host: chmod777macexp.exe

Response from Host: %T[1]%

Script Editor > Expand Internet Category > FTP Site Command

Command to Send to FTP Server

This is essentially a free form field to send a command to the FTP server. There are a large number of FTP commands available. It is necessary to know the correct format of the command to send to the FTP server. This field is case sensitive. It is also necessary to make sure that the FTP server supports the command to be sent.

FTP Library Help Definition of FTP Site

"Site is a procedure used to send a site-specific FTP command directly to the FTP server. Site is used by an FTP server to provide services specific to the host system that are essential to file transfer, but not sufficiently universal to be included as commands in the FTP protocol."

In short, the FTP Site command will not send "normal" commands like "S", "get" or "put". The commands that are sent have a syntax of "SITE something".

Log Results To

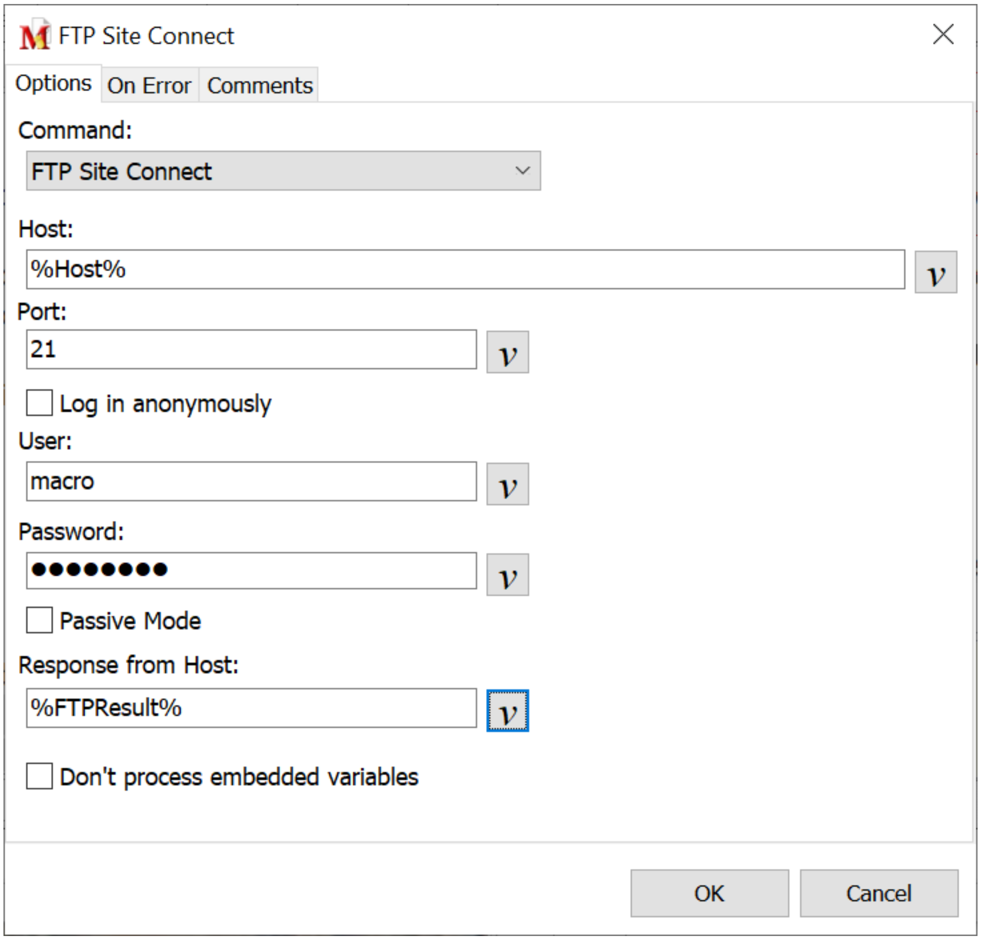
The message generated by the FTP server, whether a success or failure, is saved to the assigned variable. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

✓ **Note:** Before using this function, the macro script first requires the FTP Site Connect command to connect to the site. When all FTP actions have been completed, use the FTP Site Disconnect command to disconnect from the site.

FTP Site Connect or Disconnect

Only one FTP connection can be used at any given time. These commands assume familiarity with operating FTP servers and clients.



FTP Site Connect

Options | On Error | Comments

Command: FTP Site Connect

Host: %Host%

Port: 21

□ Log in anonymously

User: macro

Password: ●●●●●●●●●

□ Passive Mode

Response from Host: %FTPResult%

□ Don't process embedded variables

Script Editor > Expand Internet Category > FTP Site Connect or Disconnect

FTP Site Connect

The options shown in the window above pertain to connecting to a FTP site.

Host

Insert the address of the FTP server to be connected.

Port

Select the port used to connect to the FTP server. This is generally port 21.

Log in Anonymously

Use this option to connect to the FTP server anonymously. Most servers allow these kinds of connections, though they restrict the user to specific directories.

User

The login name used to connect to the FTP server.

Password

Insert the password associated with the login name.

Passive Mode

This option is used if the user is behind a firewall. Most firewalls do not allow users to connect to FTP servers unless they are in passive mode. This is primarily for security reasons.

Response from Host

Any error or success messages are stored in the specified variable. It is up to the user to determine if the connection to the FTP server was successful. There are several ways to monitor the variable to determine the status of the operation.

1. Place the variable in a Text Box Display command to display the results at the end of the macro.
2. Use the Variable Modify String command to save the variable results to a text file for later review.
3. Send the results via email to an email account.

Don't process embedded variables

When using a variable in any of the fields of this command, the Don't process embedded variables option ignores all % signs found within the variable value and does not attempt to treat the % signs as an embedded variable. It will just expand the % sign as a standard character during macro runtime.

Example

Assume for example a variable of %T[1]% with a value of ftp://ftp.macros.com/abc%23%/macexp.exe.

1. With this option selected, the macro will connect to ftp://ftp.macros.com/abc%23%/macexp.exe.

2. If the option is not selected, then the macro will assume that there is a variable of %23% and look to expand its value.

    a. If the variable %23% does not exist, then the macro will generate an error because it could not find the value for this variable.

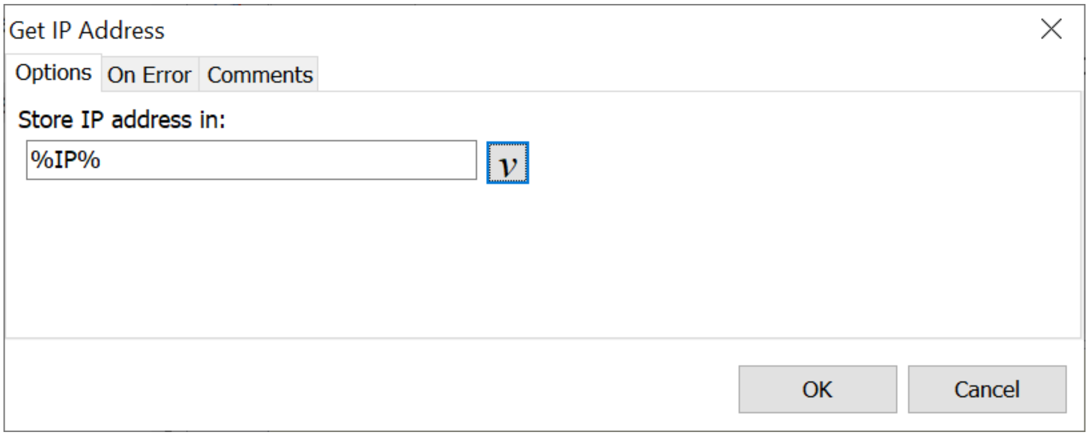
    b. Otherwise it will expand the value of %23% within the main variable.

FTP Site Disconnect

Disconnects a user from an FTP server. This command does not have any effect if the user is not connected when the FTP Site Disconnect command is executed.

Get IP Address

The Get IP Address command finds the IP (Internet Protocol) address of the computer running the macro and saves this address to a variable. This command may also help in determining if an online connection is active or not.



Store IP address in: %IP%

Script Editor > Expand Internet Category > Get IP Address

IP Address

An IP address is an identifier for a computer on a TCP/IP network. Networks using the TCP/IP protocol route messages based on the IP address of the destination. The format of an IP address is a 32-bit numeric address written as four numbers separated by periods. Each number can be from zero to 255. An example IP address is 192.14.122.216.

Within an isolated network, IP addresses can be assigned at random as long as each one is unique. However, connecting a private network to the Internet requires using registered IP addresses (called Internet addresses) to avoid duplicates.

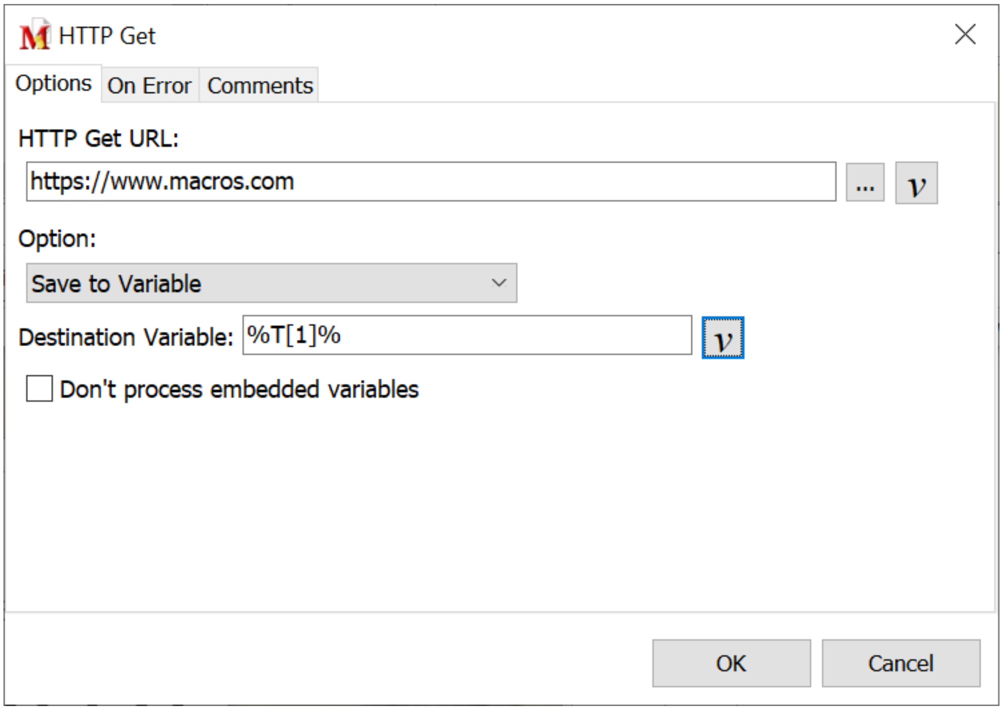
The Get IP Address command identifies the IP address of the network card on the computer - the IP address of the computer within a private network. Or if connecting to the Internet, this command identifies the IP address used to connect to the Internet. For a dial up connection, this IP address is assigned by the Internet Service Provider. It may change each time the computer is connected to the Internet.

Store IP Address In

The IP address is stored to the selected variable.

HTTP Get

The HTTP Get command gets the results for a specific web page. It is an HTTP command to request a file from a server or used for querying databases on web pages. The results may then be saved to a text string variable or to a file.



HTTP Get URL: https://www.macros.com

Option: Save to Variable

Destination Variable: %T[1]%

□ Don't process embedded variables

Script Editor > Expand Internet Category > HTTP Get

HTTP Get URL

Enter the URL desired.

Option

Save to Variable

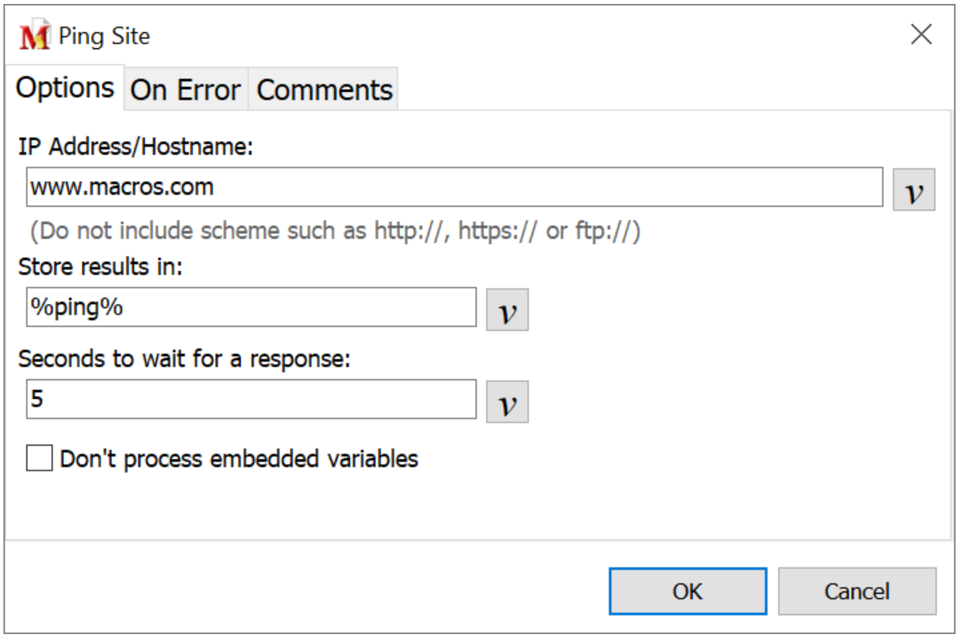
Choose a Text String Variable to store the results of the query.

Save to File

The results are saved to a file. Select a location and name the file.

Ping Site

This command pings a web site or IP address.



IP Address/Hostname:

www.macros.com

(Do not include scheme such as http://, https:// or ftp://)

Store results in:

%ping%

Seconds to wait for a response:

5

□ Don't process embedded variables

Script Editor > Expand Internet Category > Ping Site

IP Address/Hostname

Insert the address of the site to ping. This can be an address such as www.macros.com or an IP address.

Store results in

The results of the ping are saved to an assigned variable. The results include information such as if the ping was successful, or one of a number of error messages if it was not successful.

Seconds to wait for a response

Insert the number of seconds to wait for a response.

Don't process embedded variables

When using a variable in the address field, this option ignores any % signs found within the variable value and will not attempt to treat the % signs as an embedded variable. It will just expand the % sign as a standard character during macro runtime.

Example

Assume for example a variable of %T[1]% with a value of www.%DomainName%.com

1. With this option selected, the macro will ping the site found at www.%DomainName%.com

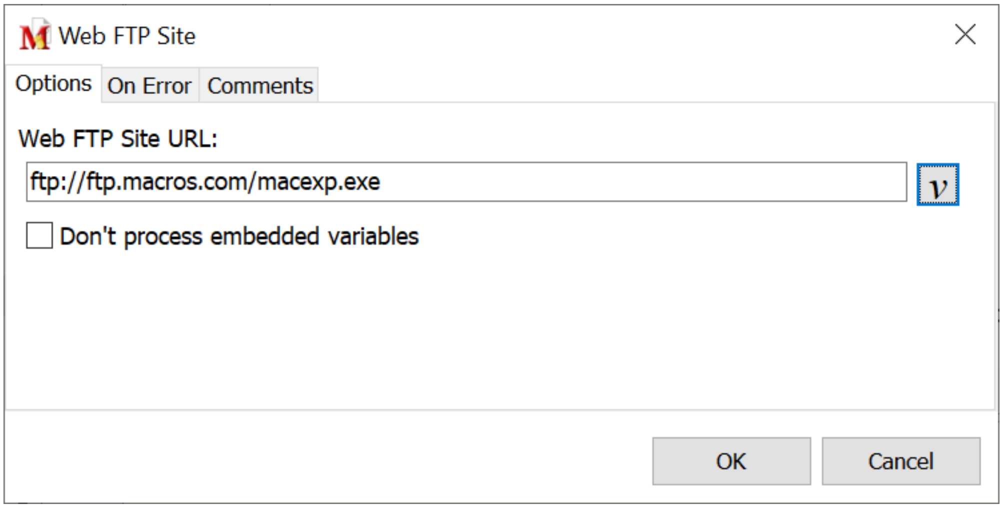
2. If the option is not selected, then the macro will assume that there is a variable of %DomainName% and look to expand its value.

   a. If the variable %DomainName% does not exist, then the macro will generate an error because it could not find the value for this variable.

   b. Otherwise it will expand the value of %DomainName% within the main variable.

Web FTP Site

The Web FTP Site command initiates an FTP transfer. Just insert the FTP address and click the OK button. An FTP transfer is initiated when the macro is executed.



Web FTP Site URL: ftp://ftp.macros.com/macexp.exe

☐ Don't process embedded variables

Script Editor > Expand Internet Category > Web FTP Site

Don't process embedded variables

When using a variable in the FTP edit field, this option ignores any % signs found within the variable value and will not attempt to treat the % signs as an embedded variable. It will just expand the % sign as a standard character during macro runtime.

Example

Assume for example a variable of %T[1]% with a value of ftp://ftp.macros.com/abc%23%/macexp.exe.

1. With this option selected, the macro will initiate the FTP transfer from ftp://ftp.macros.com/abc%23%/macexp.exe.

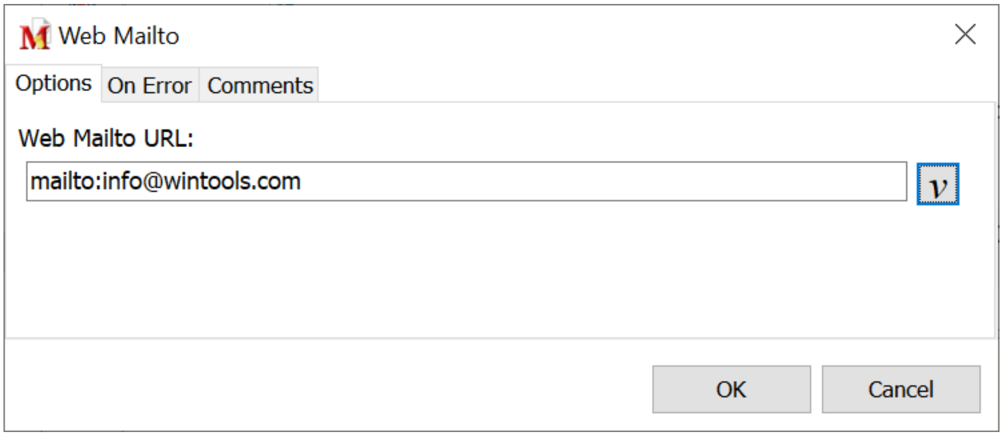
2. If the option is not selected, then the macro will assume that there is a variable of %23% and look to expand its value.

   a. If the variable %23% does not exist, then the macro will generate an error because it could not find the value for this variable.

   b. Otherwise it will expand the value of %23% within the main variable.

Web Mailto

This command provides a quick method for opening the default email program and inserting an email address in the To field of a message to be sent.



Web Mailto URL: mailto:info@wintools.com

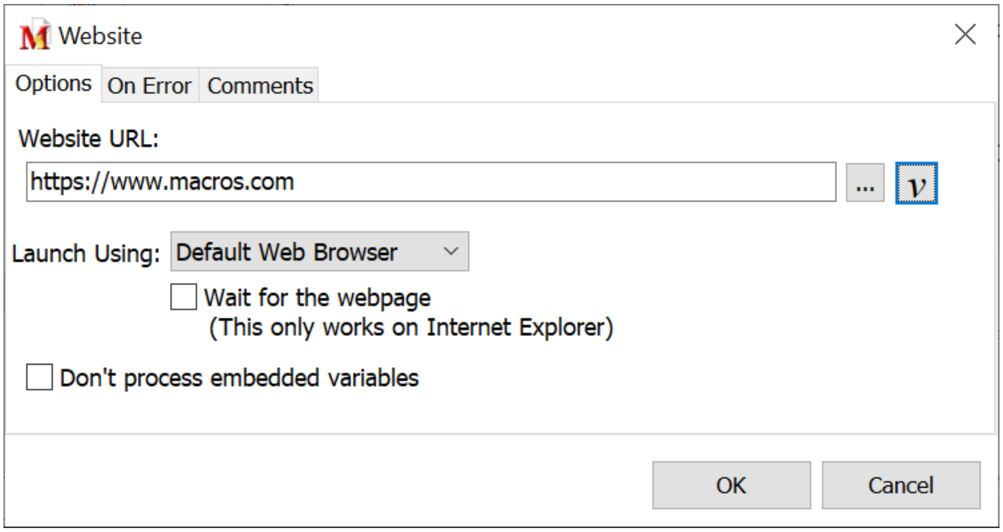
Script Editor > Expand Internet Category > Web Mailto

Insert the email address of the recipient in the address field, as shown above. The macro will open the computer's default email program to the New Message or equivalent window and insert the email address provided in the To: field.

This action is similar to clicking on a web page link to send an email to the web host. The action opens the email client and inserts the email address in the To: field. Most email programs support this function, though not all.

Web Site

Use this command to open a web page. Just enter the web site address and click the OK button. The web site entered is launched when the macro is run. If the browser is not open, the macro also opens the browser.



Website URL: https://www.macros.com

Launch Using: Default Web Browser

□ Wait for the webpage
(This only works on Internet Explorer)

□ Don't process embedded variables

Script Editor > Expand Internet Category > Web Site

Please Enter the Website

Insert the URL that is to be opened. The default setting is https:// for secure web sites. If opening a non-secure site, remove the "s" from the default to just show http://.

Favorites

Rather than type in a URL, click on the Favorites button to open the list of URL's in the Favorites folder. Select the URL, highlight it and click OK to insert the URL into the address field.

Launch Using

Select which browser to load and open the web page.

Default Browser

This option loads the browser that currently is designated by Windows as the default web browser.

Internet Explorer

The other option is to select Internet Explorer as the browser. If Internet

Explorer is already the default browser, then either selection is valid. But if another browser is currently set as the default and macro timing is critical, consider selecting Internet Explorer to take advantage of the Wait for Web Page option.

Wait for the Web Page

When selected, the macro pauses until the web page has fully loaded. After the page has fully loaded, the macro continues. The Wait for Web Page option is only available for Internet Explorer.

Don't process embedded variables

When using a variable in the website field, this option ignores any % signs found within the variable value and will not attempt to treat the % signs as an embedded variable. It will just expand the % sign as a standard character during macro runtime.

Example

Assume for example a variable of %T[1]% with a value of https://www.macros.com/abc%23%/support.htm.

1. With this option selected, the macro will load the web page found at https://www.macros.com/abc%23%/support.htm.

2. If the option is not selected, then the macro will assume that there is a variable of %23% and look to expand its value.

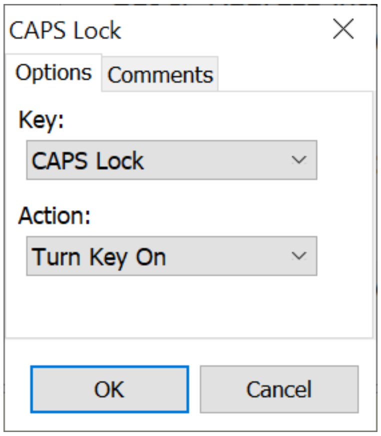
   a. If the variable %23% does not exist, then the macro will generate an error because it could not find the value for this variable.

   b. Otherwise it will expand the value of %23% within the main variable.

Keyboard

Caps, Num and Scroll Lock Keys

This command activates or releases the Caps Lock, Num Lock or Scroll Lock keys.



Key: CAPS Lock

Action: Turn Key On

Script Editor > Expand Keyboard Category > Caps Lock, Num Lock or Sroll Lock Options

For example, to capitalize a section of text (all caps), insert the Caps Lock On command before the text that is to be typed. To revert back to regular text, place a Caps Lock Off command before the next section of text. The Num Lock option is only available in Windows 2000 and later.

Key Action

Select which key to use - the Caps Lock, Num Lock or Scroll Lock key.

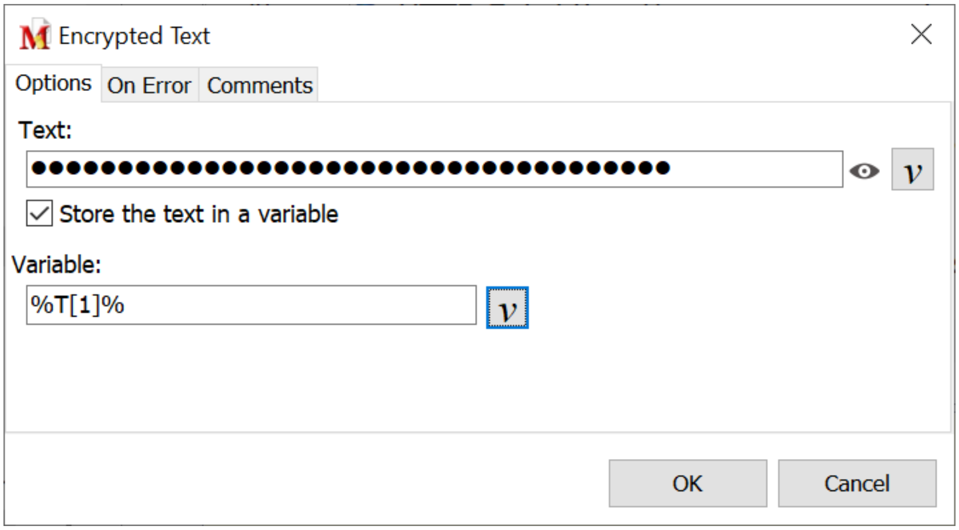
Action

Select whether to turn the key on or off.

- ✓ **Note:** If the Caps Lock, Num Lock or Scroll lock is turned on during macro playback, these keys will remain locked after the macro has completed, unless a corresponding Turn Key Off command is also issued playback.

Encrypted Text

When viewing or editing the contents of the macro, the encrypted text displays as encrypted text - *****. The command is handy for passwords and the like so that people snooping through the program cannot readily uncover passwords. Use this along with the Password command for additional security.



Text:
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

☑ Store the text in a variable

Variable:
%T[1]%

Script Editor > Expand Keyboard Category > Encrypted Text

Enter the text to be encrypted into the text edit field. Whatever is typed into this field appears as above, a series of dots. Each dot represents a character input. The actual text is typed when the macro is played back, not the encrypted dots.

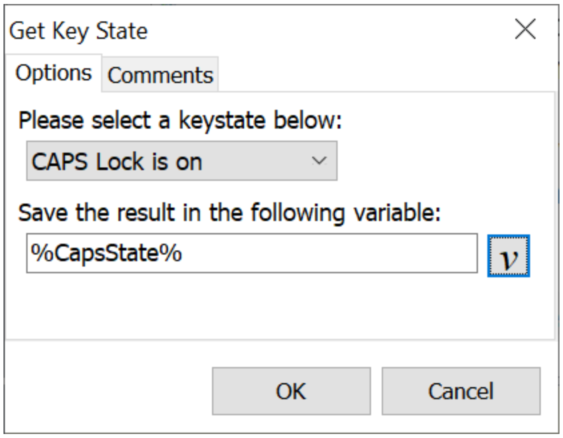
Store the Text in a Variable

Place the encrypted text into a variable with this option. Enter the variable name or select from the list of variables previously defined in the macro.

Get Key State

This command saves the status of specific keys to a Boolean variable - true or false statement. In the example below we are checking the state of the CAPS lock key. The assumption is made that the CAPS Lock key is on. The result is saved to a Boolean variable. In this case the variable is named Key State. If the CAPS lock key is on, the Boolean variable stores a True value. If the Caps Lock key is off, the Boolean variable stores a False value.

Create If Statements based on the results to turn the CAPS lock key on if it isn't already, etc.



Get Key State

Please select a keystate below: CAPS Lock is on

Save the result in the following variable: %CapsState%

Script Editor > Expand Keyboard Category > Get Key State

Please select a keystate below:

Select one of eight key states from the drop down list. The keystate options are listed below:

- Caps Lock is on
- Scroll Lock is on
- Num Lock is on
- Shift is pressed
- Ctrl is pressed
- Alt is pressed
- Left Win key is down
- Right Win key is down

Save the result in the following variable:

Select or insert the variable that will store the value. The variable selected must be a Boolean variable.

Lock/Unlock Keyboard and Mouse

These commands are inserted directly into the script as there is not an edit window associated with them. From the Script Editor click on the Keyboard category of commands to expand the category. Then select either the Lock Keyboard and Mouse or the Unlock Keyboard and Mouse command to insert into the script.

Lock Keyboard and Mouse

The lock Keyboard and Mouse command is used to prevent the user from typing or moving the mouse during playback of a macro, so that the macro is not disrupted. Once the macro has completed, keyboard and mouse control are returned to the user.

Unlock Keyboard and Mouse

It is not necessary to insert this command at the end of a macro to unlock the keyboard and mouse. This occurs automatically when the macro has finished playback.

Use this option when the Lock Keyboard and Mouse command is active and the macro contains a prompt for the user to input information, navigate to a file, etc. Insert the Unlock Keyboard and Mouse command into the script before the prompt to allow the user to take appropriate action. Lock the keyboard and mouse again or leave them unlocked for the remainder of the macro playback.

- ✓ **Note:** This command will not work in Windows Vista or later unless running Macro Express in Administrator mode.

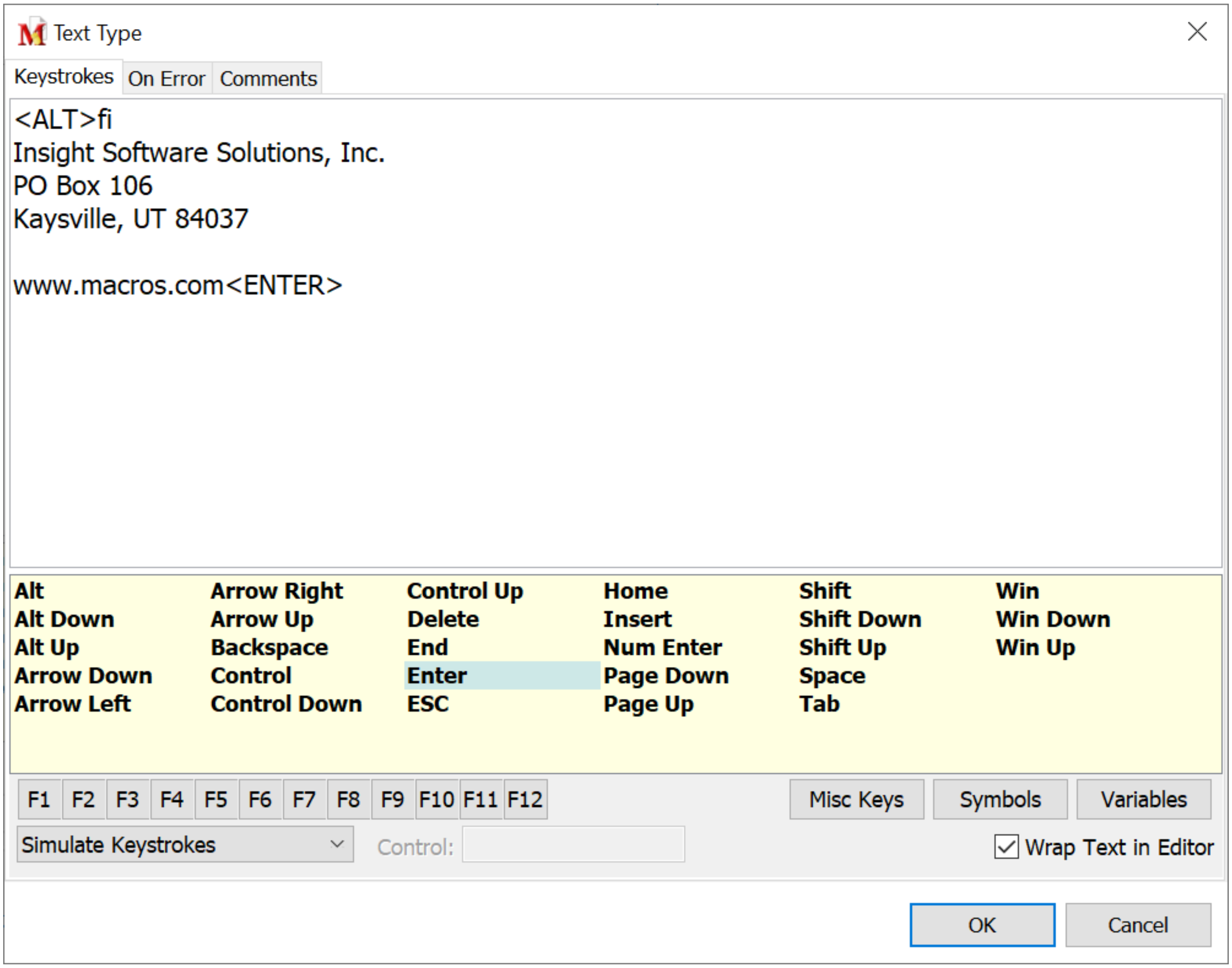
Caution: This is a Windows function that completely locks the keyboard and mouse. Use with caution. The keyboard and mouse remain locked if the macro gets stuck or an unexpected dialog appears during playback.

In these situations, the key to regaining control of the mouse and keyboard is pressing the Ctrl + Alt + Delete keys. The lock keyboard command does not prevent this key combination from working. Depending on the version of Windows being used, one or two additional steps may be required.

Press (Ctrl + Alt + Delete) to display the screen to lock the computer, open the task manager, etc. Press the Cancel button or Cancel arrow and control of the mouse and keyboard is returned.

Text Type

When in the Script Editor, use this window to enter text and a variety of other keystrokes as part of the macro script. The text or keystrokes play back into the application that is open when running the macro. The Text Type command is not necessary when using the Direct Editor. Text and the keystrokes displayed below may be input directly to the Direct Editor.



```
<ALT>fi
Insight Software Solutions, Inc.
PO Box 106
Kaysville, UT 84037

www.macros.com<ENTER>
```

Simulate Keystrokes

☑ Wrap Text in Editor

Script Editor > Expand Keyboard Category > Text Type

Inserting Keys and Text

Type the text to include in the macro script directly into the edit box as shown above. Click on one of the entries, such as Delete, in the yellow area to insert it into the edit box. Or tab and arrow to the entry and press Enter to insert the item into the macro script. Clicking on one of the F1 - F12 buttons inserts the selected F1 - F12 key into the edit field.

- ✓ **Note:** When using the Alt key or Ctrl Key to automate menu items, accelerator keys etc, use lower case letters after the key, such as in the example above - <Alt>fi. If <ALT>FI were used instead, the program would interpret the keys as <ALT><SHIFT>f<SHIFT>i, generating unpredictable results.

Hold a Key Down

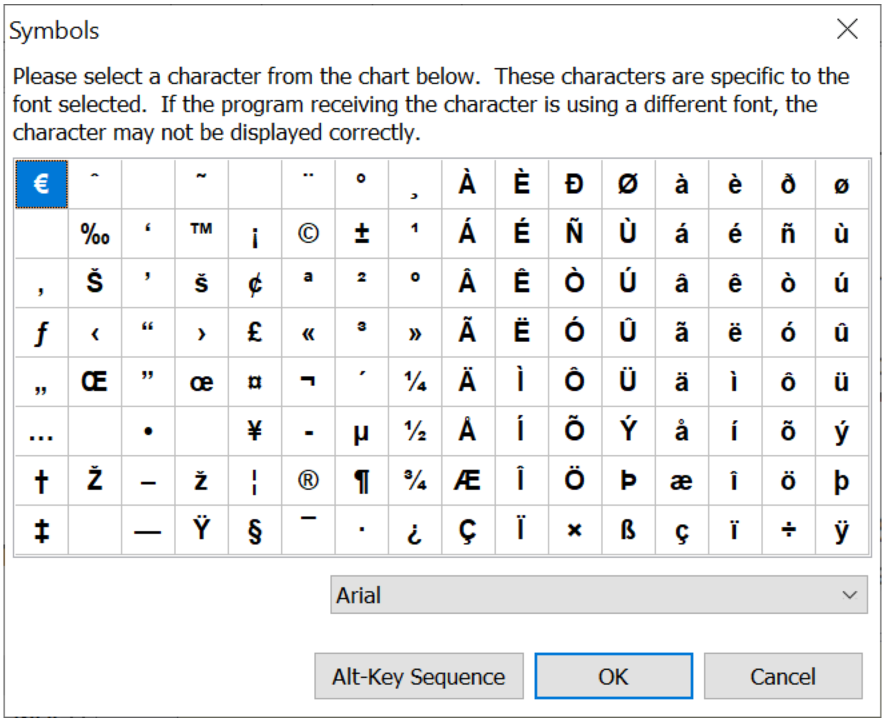
In some situations you may need the ability to hold a particular key down until you instruct the key to be released. Several keys, such as ALT, CTRL, SHIFT and the Windows key, already have options in the Text Type command to hold them down and release them.

For other keys or virtual keys, there is a special syntax that must be typed into the Keystrokes edit box. For example, enter <KEYD:VK_TAB> to hold down the "TAB" key. Enter <KEYU:VK_TAB> to release the key when needed. Or insert <KEYD:Q> to hold down the "Q" key and <KEYU:Q> to release it.

Click here to view a list of virtual keys available for use in Macro Express.

Symbols

To insert symbols or characters into the macro text click on the Symbols button to open the Symbols window as shown below. Locate the symbol or character, highlight it and click on the OK button to insert it into the script. Or double click on the symbol to insert into the script.



Please select a character from the chart below. These characters are specific to the font selected. If the program receiving the character is using a different font, the character may not be displayed correctly.

€	ˆ		˜		¨	°	¸	À	È	Ð	Ø	à	è	ð	ø
	‰	‘	™	¡	©	±	¹	Á	É	Ñ	Ù	á	é	ñ	ù
‚	Š	’	š	¢	ª	²	º	Â	Ê	Ò	Ú	â	ê	ò	ú
ƒ	‹	“	›	£	«	³	»	Ã	Ë	Ó	Û	ã	ë	ó	û
„	Œ	”	œ	¤	¬	´	¼	Ä	Ì	Ô	Ü	ä	ì	ô	ü
…		•		¥	-	µ	½	Å	Í	Õ	Ý	å	í	õ	ý
†	Ž	–	ž	¦	®	¶	¾	Æ	Î	Ö	Þ	æ	î	ö	þ
‡		—	Ÿ	§	¯	·	¿	Ç	Ï	×	ß	ç	ï	÷	ÿ

Arial

Select a font

Select the font from the drop down list. Symbols associated with the font selected are displayed. The program default is the Arial font.

Alt-Key Sequence

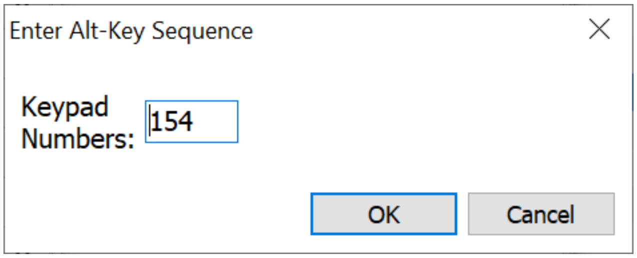
Use this option to embed an Alt-Key Sequence directly into a macro. The Alt-Key Sequence is a method of inserting symbols into most Windows applications and is done by holding the ALT key down and pressing digits on the numeric keypad. Clicking on the "Alt-Key Sequence" button presents a window requesting the numeric digits that would typically be typed via the numeric keypad.

Use this functionality to enter characters that may not be present on your keyboard, such as á, Ô, ©, ¾, ¿ or п, or characters not found in the symbols list represented above.

There are three ways to enter special characters using the Alt-Key sequence:

1. + 1 to 3 digits

Do not type a leading 0 for this method.


Enter Alt-Key Sequence

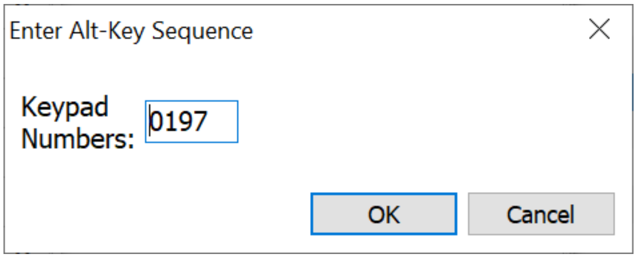
Keypad Numbers: 154

Text Type Command > Symbols Button > Alt-Key Sequence Button

- For example will insert the Ü character.

2. + 3 digits

This method requires that the first digit typed be 0.


Enter Alt-Key Sequence

Keypad Numbers: 0197

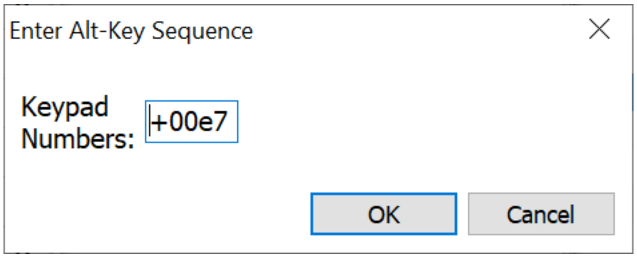
Text Type Command > Symbols Button > Alt-Key Sequence Button

- For example will insert the Å character.

✓ **Note:** Be mindful when to include the leading zero or not, because the outcome may result in different characters. For example,   will insert the ≈ character but   inserts the ÷ character.

3.   + hexadecimal value for the character

Use this method to have Macro Express type Unicode characters into your application. The first digit input in the edit field must be the + character.



Enter Alt-Key Sequence

Keypad Numbers: +00e7

Text Type Command > Symbols Button > Alt-Key Sequence Button

- For example   will insert the ç character.

Using Hexadecimal values with Microsoft Word and other Office products

Microsoft Word and several other Office products do not allow for the method described above. If it does not work in your Office or alternative Office product, use the following method.

Enter the hexadecimal value, without the + (plus) sign, directly into the Text Type edit field. Follow the value with an Alt Key Down, the letter x and an Alt Key Up.

Below are a couple of examples.

- Telephone image
  260e<ALTD>x<ALTU>

- British Pounds Sterling symbol
  20a4<ALTD>x<ALTU>

✓ **Note:** To use the hex input method you need to set or create the REG_SZ registry value HKEY_CURRENT_USER\Control Panel\Input Method\EnableHexNumpad and give it a value of 1. You will need to log off and back on or reboot the computer before this setting takes effect.

For those unfamiliar with this process, the steps to create the value are listed below.

1. Open the Windows Registry Editor - from the Windows Start menu search for regedit.exe.

2. Expand HKEY_CURRENT_USER and then Control Panel.
3. Right mouse click on Input Method and select New and then String Value from the menu.
4. Rename the NewValue#1 entry to EnableHexNumpad.
5. Right click on the EnableHexNumpad entry you just created and select Modify from the menu.
6. In the Value Data Field enter the number 1 and click OK.
7. The entry should show EnableHexNumpad as the Name, REG_SZ as the Type, and 1 in the Data column.
8. Log off and back on or reboot for the new setting to take effect.

Below is a sample of various characters generated via the three Alt Keypad methods and their associated codes. A quick Internet search will provide the codes and resulting character values for each of these methods.

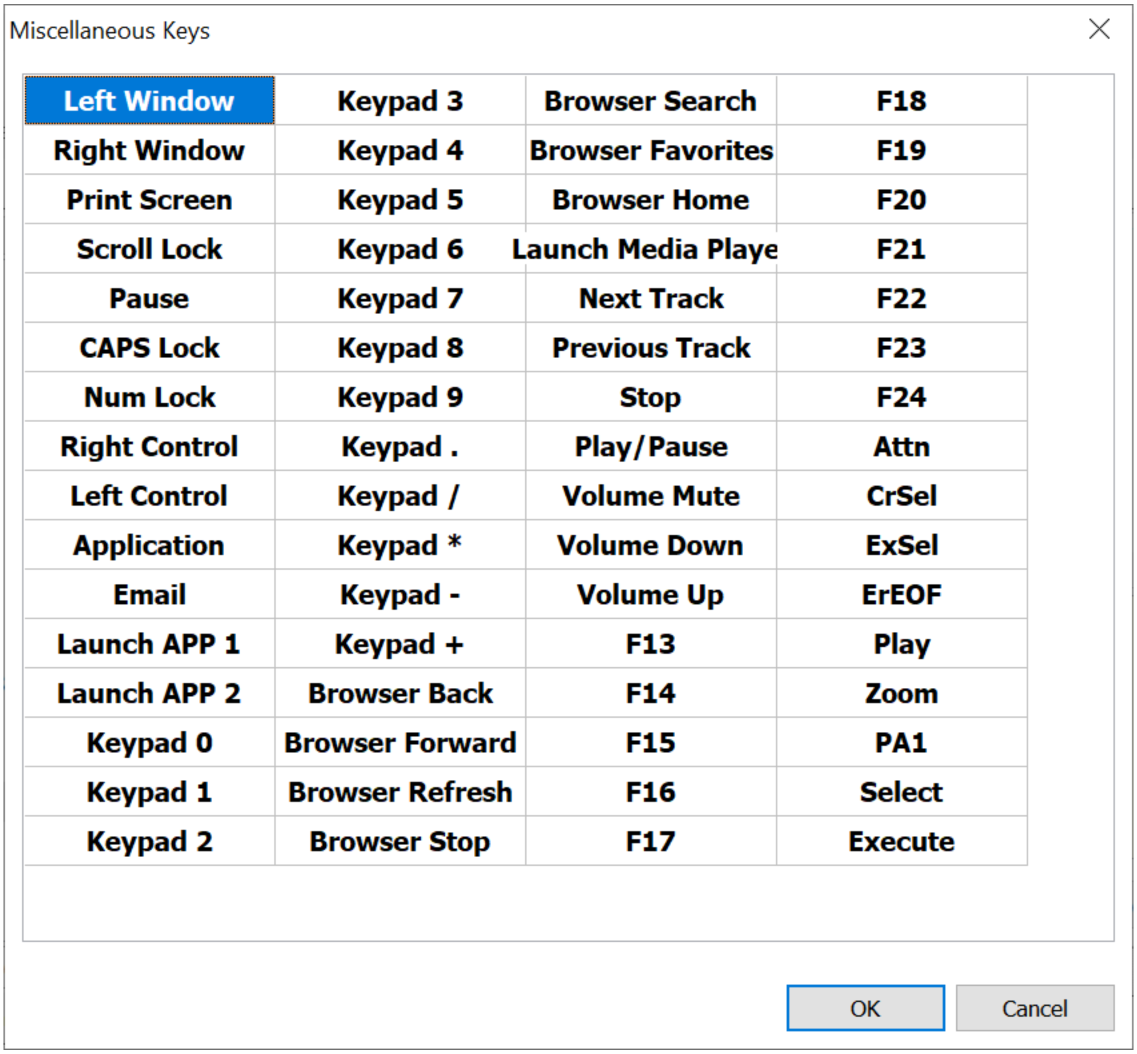
Character	Alt Code	Character	Alt Code + 0	Character	Alt + Hexadecim
Ç	Alt 128	€	Alt 0128	ç	Alt +00e7
Ñ	Alt 165	¥	Alt 0165	?	Alt +0255
+	Alt 195	Ã	Alt 0195	?	Alt +02ab
-	Alt 196	Ä	Alt 0196	?	Alt +01ff
+	Alt 201	É	Alt 0201	齾	Alt +9f7e
+	Alt 211	Ó	Alt 0211	☆	Alt +2606
¦	Alt 222	Þ	Alt 0222	£	Alt +20a4
Θ	Alt 233	é	Alt 0233	¥	Alt +00a5
≈	Alt 247	÷	Alt 0247	☎	Alt +260e

Variables

Insert variables into the macro text. The variable value is typed during the playback of the macro. Click on the Variables list box to select the desired variable and then click on the Insert button. If selecting T[1] as the variable, it displays as %T[1]% in the macro script.

Misc Keys

Click on the Misc Keys button to display a list of miscellaneous keys that may be inserted into the Text Type command. Highlight the key needed and click the OK button to insert the key into the edit field. The key is placed where the text cursor is positioned.



Left Window	Keypad 3	Browser Search	F18
Right Window	Keypad 4	Browser Favorites	F19
Print Screen	Keypad 5	Browser Home	F20
Scroll Lock	Keypad 6	Launch Media Playe	F21
Pause	Keypad 7	Next Track	F22
CAPS Lock	Keypad 8	Previous Track	F23
Num Lock	Keypad 9	Stop	F24
Right Control	Keypad .	Play/Pause	Attn
Left Control	Keypad /	Volume Mute	CrSel
Application	Keypad *	Volume Down	ExSel
Email	Keypad -	Volume Up	ErEOF
Launch APP 1	Keypad +	F13	Play
Launch APP 2	Browser Back	F14	Zoom
Keypad 0	Browser Forward	F15	PA1
Keypad 1	Browser Refresh	F16	Select
Keypad 2	Browser Stop	F17	Execute

✓ **Note:** Make sure that the cursor is positioned correctly before inserting symbols, variables, etc. Symbols, keystrokes, variables, etc. are placed in the script wherever the cursor is located.

Playback Options

Select from the drop down list one of four options for playing back the text. These are explained below.

Simulate Keystrokes

The simulate keystrokes option types the text keystroke by keystroke into the receiving application. This is the default selection.

❖ **Caution:** It is possible when typing back long strings of text that the Keyboard Buffer may overflow. When this happens some of the text may be missing or scrambled or the computer emits a strange high pitched sound. In these instances the text is playing back too fast. To correct this problem, either select the Use Clipboard to Paste Text option, set a default Text Type Delay from the Preferences window, or use the Keystroke Speed command to control the text playback speed.

Use Clipboard and Paste Text

This option pastes the text through the Windows clipboard into the receiving application. This option should only be used with text. Keys such as Alt, Ctrl, Shift, etc. paste back as <ALT>, <CONTROL>, <SHIFT>, etc. instead of simulating the actual keystroke.

Send Text Directly to Control

If a Control Variable is previously defined, the text entered in this command is typed into the selected Window Control. The control chosen must be able to accept text input. Otherwise this command will appear to do nothing.

Paste Rich Text

Select this option to display components that allow for rich text in the keystrokes edit box. Select various font types, sizes, colors for the text. Also bold, underline, italicize or justify the text.

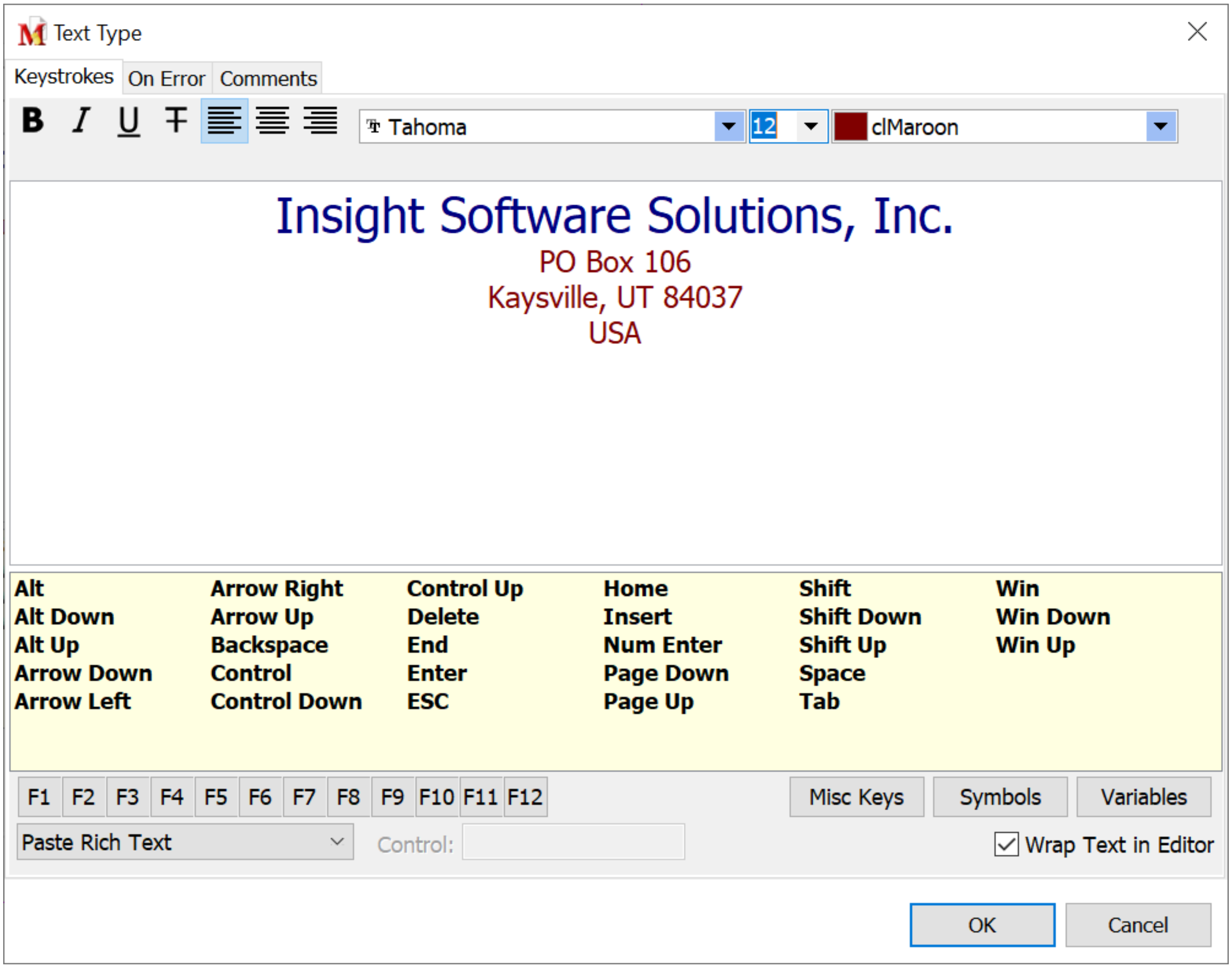
Use the available components to create the rich text or copy the rich text from another source and paste into the edit box. The rich text will be pasted into applications that accept rich text.

Wrap Text in Editor

Select this option to have the text in the Keystrokes edit field wrap so that all text is visible without scrolling to the right. This function performs similar to the Word Wrap feature in text editor programs, such as Notepad and others.

Rich Text

Macro Express Pro has the ability to play back text in a rich text format. Select the Paste Rich Text option from the drop down menu to display the rich text editor options. Create the rich text content directly in the Text Type command or copy the rich text from another source and paste it into the edit box.



Text Type

Font: Tahoma; Size: 12; Color: clMaroon

Insight Software Solutions, Inc.
PO Box 106
Kaysville, UT 84037
USA

Paste Rich Text

Control:

☑ Wrap Text in Editor

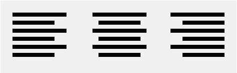
Script Editor > Expand Keyboard Category > Text Type > Select Paste Rich Text from drop down menu

Paste Rich Text

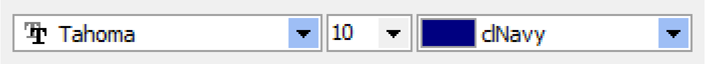
With this option selected, the edit box supports rich text. Create macros with varying fonts, sizes, colors, etc. Highlight the text in the edit box and select from the rich text options described below.

B *I* U Ŧ

Display text with bold, underline, italics and strike through features.



Left, center or right justify the text.



Choose from different font types, sizes and colors.

Colors such as clWindowText, clWindowFrame, clScrollBar are assigned to specific parts of the window. For example, on different computers clWindowText may be black, dark gray or white depending on the theme or custom color choices.

Things to consider when using Rich Text

- If the "Paste Rich Text" option is selected, the text will always be pasted from the clipboard. Anything previously saved to the clipboard will be lost.
- If the "Paste Rich Text" option is turned off after creating rich text and the Text Type command is saved, the rich text formatting is lost and will need to be reformatted, if desired.
- If the application that Macro Express is pasting into doesn't support rich text, such as Notepad, then the text is pasted in plain text format.
- Only the Tab, Enter and Num Pad Enter keys will play back correctly when being pasted through the clipboard. Other keys will just paste back their text equivalent, such as <Alt> or <CTRL>.
- Variable values play back correctly when pasted in rich text format.

Logic

Using If Statements

The If statements add a new level of sophistication and intelligence to macros. They allow macros to take action based on certain conditions. If the condition is met, the macro takes a specific course of action. If it is not met, the macro does something else or ignores the condition. For example, insert text into a window if the window is open. If it is not open, have the macro do something else.

Below are some examples and things to keep in mind when using If statements.

1. End If

Every If statement must have a closing End If statement. The End If command signifies the end of the If statement and conditional logic used in the macro.

This sample macro highlights a line of text in Notepad and copies it to the clipboard. If the clipboard contains the word "Computer", the line is pasted into Excel. If the line does not contain the word "Computer", then the conditional is ignored and the next line in Notepad is copied to the clipboard.

Sample Macro

```
Repeat Start (Repeat 150 times)
  Text Type (Simulate Keystrokes): <SHIFT><END>
  Clipboard Copy
  Delay: .25 seconds, without ability to halt
  If Clipboard Contains "Computer"
    Window Activate: Excel
    Wait for Window Title: Excel
    Clipboard Paste
    Delay: .25 seconds, without ability to halt
    Text Type (Simulate Keystrokes): <ENTER>
    Window Activate: notepad
  End If
  Text Type (Simulate Keystrokes): <ARROW
DOWN><HOME>
End Repeat
```

2. Else

The Else statement is the primary way to provide an either/or type option. If one condition isn't met, then the macro takes an alternative course of action. It does something else.

This sample macro is similar to the one above but includes an Else statement. The macro highlights a line of text in Microsoft Word and copies it to the clipboard. If the clipboard contains the word "Computer", the text is pasted into Notepad. Because of the Else statement, the text is pasted into WordPad if the clipboard does not contain the word "Computer".

Sample Macro

```
Repeat Start (Repeat 10 times)
  Text Type (Simulate Keystrokes): <SHIFT><END>
  Clipboard Copy
  Delay: .25 seconds, without ability to halt
  If Clipboard Contains "Computer"
    Window Activate: notepad
    Wait for Window Title: notepad
    Clipboard Paste
    Delay: .25 seconds, without ability to halt
    Text Type (Simulate Keystrokes): <ENTER>
    Window Activate: Microsoft Word
  Else
    Window Activate: Document - WordPad
    Clipboard Paste
    Delay: .25 seconds, without ability to halt
    Text Type (Simulate Keystrokes): <ENTER>
    Window Activate: Microsoft Word
  End If
  Text Type (Simulate Keystrokes): <ARROW DOWN><HOME>
End Repeat
```

3. AND/OR/XOR

The AND/OR/XOR options provide additional If Statement flexibility. The AND option requires that two If statements be true before action is taken. The OR option requires that either one or the other If statement be true before action is taken. With the XOR option, one If statement must be true and the other false in order for subsequent action to be taken.

Example macros using these options are included in the AND/OR/XOR help topic.

4. Nested If Statements

If statements may be nested inside other If statements. This example is a segment of a larger macro. It illustrates the nesting of one If statement within another. Each of the nested If statements requires a matching End If command.

Sample Macro

```
Variable Modify String %T[15]%: Copy Whole Text (%N[1]%)
  If Variable %N[1]% Is Greater Than 1
    Variable Modify String: Replace "(" in %T[16]% with ""
    Variable Modify String: Replace ")" in %T[16]% with ""
  Else
    If Variable %N[1]% Equals 1
      Variable Modify String: Replace "(s)" in %T[16]% with ""
    End If
  End If
```

AND/OR/XOR

These commands increase the power and flexibility of the If statements. The commands are inserted directly into the script as there is not an associated edit window. From the Script Editor click on the Logic category of commands to expand the category. Then select either the AND, OR or XOR command to insert into the script.

AND

Use this command if requiring two or more conditions to be valid. Multiple AND commands may be placed between the If statements. In this scenario only one End If command is required at the end of the string.

Sample Macro

Let's use an example of T[1] and T[2]. We can create a macro that states if T[1] = Test AND if T[2] = Example, run the macro "Macrotest.

If T[1] does not = Test or if T[2] does not = Example, then a text box will display instead of running the "Macrotest" macro. Both IF statements must be true in order for the "Macrotest" macro to continue.

```
If Variable %T[1]% = "Test"
  AND
If Variable %T[2]% = "Example"
  Macro Run: Macrotest
Else
  Text Box Display: Not Found
End If
```

OR

Use this command if requiring one of two or more conditions to be valid. Multiple OR commands may be placed between the If statements. In this scenario only one End If command is required at the end of the string.

Sample Macro

In this example only one of the two If statements must be valid. Either T[1] can equal Test or T[2] can equal Example. Both are not required in order for the "Macrotest" macro to run.

```
If Variable %T[1]% = "Test"
  OR
If Variable %T[2]% = "Example"
  Macro Run: Macrotest
Else
  Text Box Display: Not Found
End If
```

XOR

With this option, one of the variables must be false and the other valid. Only one End If is required when the XOR command is placed between two If statements.

Sample Macro

In this example T[1] would equal Test and T[2] would NOT equal Example, or vice versa in order for the "Macrotest" macro to run. If both If statements were invalid or both were true, the macro would display a text box and not run the "Macrotest" macro.

```
If Variable %T[1]% = "Test"
  XOR
If Variable %T[2]% = "Example"
  Macro Run: Macrotest
Else
  Text Box Display: Not Found
End If
```

Break

The Break command may be used with the Repeat commands, Process a Text File or Process ASCII Text File commands. Each of these commands processes data in a loop. Use the Break command to exit out of a loop before the entire cycle is completed.

There is not an edit window associated with this command. From the Logic category of commands click on the Break command to insert it directly into the script.

For example, create conditional statements within the Process File or Repeat loops using If commands. If the condition is met, the Break command exits out of the loop instead of continuing through the entire process. The macro skips all other steps in the loop. It resumes playback with the first command in the script found after the End Repeat or End Process command and finishes the rest of the macro.

Sample Macro

Here's an example of how this would work. The first few commands count the number of files in a folder. When the macro loops to the 30th file, it breaks out of the Repeat loop. It sends an email containing the data appended to the text file selected in line seven of the macro.

```
1. Variable Set Integer %N[1]% to 0
2. Repeat with Folder C:\Downloads\Macro Express
3.   Variable Modify Integer: %N[1]% = %N[1]% + 1
4.   If Variable %N[1]% Equals 30
5.     Break
6.   End If
7.   Variable Modify String: Append %T[1]% to
"C:\Macexp.txt"
8. End Repeat
9. Email Send: Know Recipients, Fixed Text, Attachments
```

- **Note:** The Repeat Exit command performs the same function as the Break command within repeat loops. Use the Continue command if needing to skip to the end of the Repeat or Process File loop and return to process the next record or continue repeating.

Case and Switch

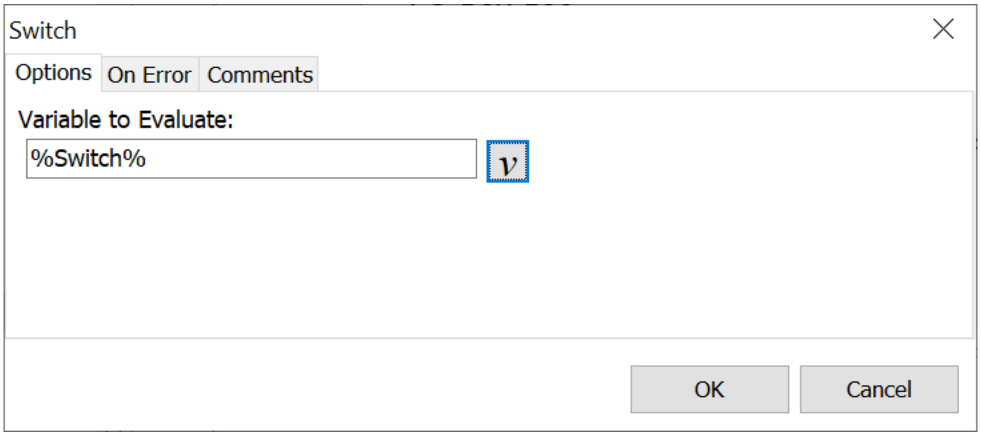
The Switch/Case group of commands allows for the writing of complicated conditional statements that are simpler and more readable than those using a number of nested If Variable statements. Pascal or C programmers will recognize that these commands are similar to the Switch/Case statements in C and the Case statement in Pascal.

The Switch/Case commands test the content of a single variable against a series of values and execute specific macro commands based on the value in the variable.

A Switch/Case block of macro commands always begins with the Switch command and ends with the End Switch command. Place one or more Case/End Case blocks inside the Switch/Case block. The Case/End blocks begin with a Case command and end with an End Case command. Optionally include one Default Case/End Case block.

Switch

The Switch command requires a variable. This may include any type of variable, such as Integer, Text or Decimal. But all values in the Case statements that follow must be consistent with the type of variable chosen for the Switch statement. If the values are not consistent, the results may be unpredictable.



Variable to Evaluate: %Switch%

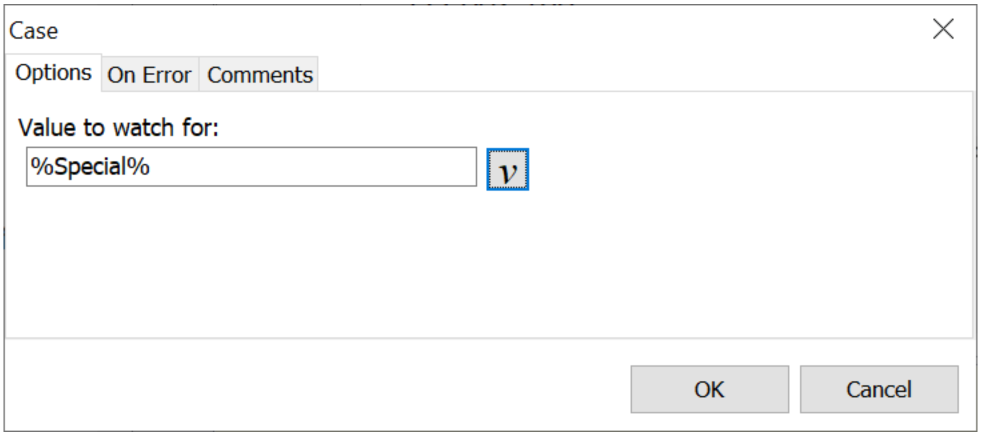
Script Editor > Expand Logic Category > Switch

End Switch

The End Switch command is required to end a Switch Block and is similar to the "End If" command. No edit window is available. The command is inserted directly into the script.

Case

The Case command starts a Case/End Case block of macro commands.



Value to watch for: %Special%

Script Editor > Expand Logic Category > Case

The Case command may contain a variable or it may contain a value that is entered directly in the Case command. For example, use a value of %T[2]% or enter a text string such as "Done".

A Case/End Case block may contain more than one Case command with a single End Case command at the end of the block.

Sample Macro

This example tests the value in the T[1] variable for the word Found or any of the words Done, Finished or Completed.

```
Switch (%T[1]%)
  Case: Found
    Text Box Display: T[1] equals Found
  End Case
  Case: Done
  Case: Finished
  Case: Completed
    Text Box Display: The comparison is done
  End Case
End Switch
```

Default Case

The Default Case/End Case block is optional. This command is similar to the standard case, but acts like an "Else" command. Basically, if none of the conditions are met in the previous case statements, then the default case is executed. No edit window is available. The command is inserted directly into the script.

If used, the Default Case/End Case block must be placed after all other Case/End Case blocks. Any Case commands that follow the Default Case command will not run.

Sample Macro

This example tests the variable T[1] for the word Found.

```
Switch (%T[1]%)
  Case: Found
    Text Box Display: T[1] equals Found
  End Case
  Default Case
    Text Box Display: T[1] does not equal Found
  End Case
End Switch
```

A Case command and a Default Case command may be combined in a single block.

Sample Macro

In this example, if the variable T[1] either contains the word Searching or if it does not contain the word Found, then the commands in the Default Case block will run.

```
Switch (%T[1]%)
  Case: Found
    Text Box Display: T[1] equals Found
  End Case
  Case: Searching
  Default Case
    Text Box Display: T[1] does not equal Found
  End Case
End Switch
```

End Case

The End Case command defines the end of a Case/End Case or Default Case/End Case block of macro commands. No edit window is available. The command is inserted directly into the script.

Else / End If

The Else and End If commands are inserted directly into the script as there is not an edit window associated with them. From the Script Editor click on the Logic category of commands to expand the category. Then insert the Else or End If command into the script.

Else

The ELSE command is used in conjunction with a conditional statement. Everything after an ELSE command until an ENDIF command is reached will be executed if the conditional is FALSE.

End If

The ENDIF command is used to mark the place where macro control should stop after a conditional statement is encountered.

Example

Below is a small example of how to use the If statements. For this example we selected the If Clipboard Text Equals command.

If the text saved to the clipboard is "Macro Express" then the macro will open the www.macros.com home page. The Else command separates the If statement. If something other than "Macro Express" is saved to the clipboard, the web site command is skipped. Instead, the macro activates the Notepad window and enters "This is a test".

Sample Macro

```
If Clipboard Equals "Macro Express"
  Web Site, "https://www.macros.com", using Default Web
    Browser
Else
  Window Activate: Notepad
  Wait for Window Title: Notepad
  Text Type (Simulate Keystrokes): This is a test.
End If
---- continue with remainder of macro commands ---
```

This is essentially the format used for any of the If statements when wanting to take two different actions based on the result of the If statement.

If two different actions are not needed, remove the Else statement as shown below. In this case, if the clipboard contents equal Macro Express, the macro opens the www.macros.com web page. If the clipboard contents are different, then the web site command is skipped. The End If command signifies that the If statement is finished and the macro continues with the rest of the commands.

Sample Macro

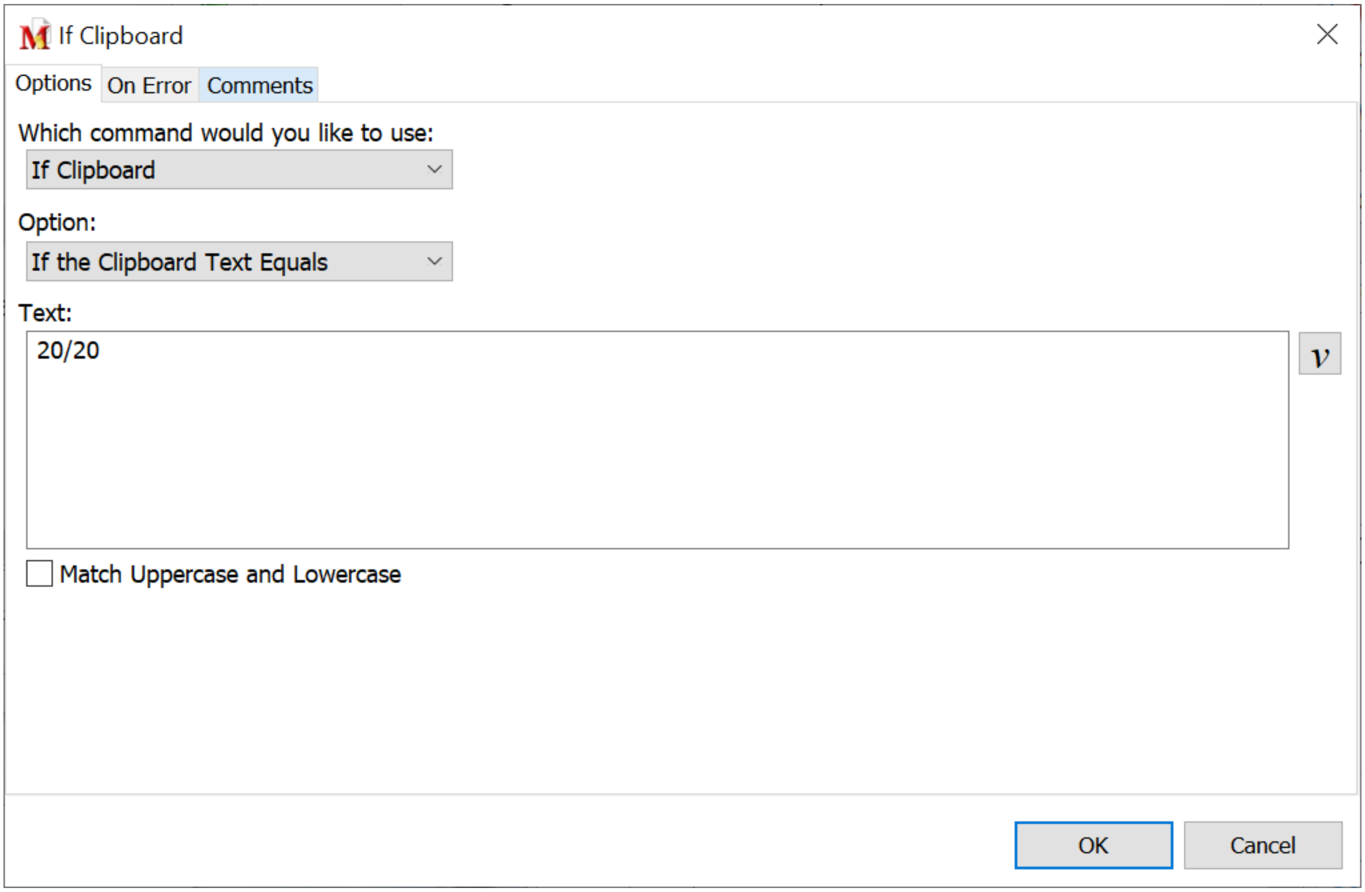
```
If Clipboard Equals "Macro Express"
  Web Site, "https://www.macros.com", using Default Web Browser
End If
---- continue with remainder of macro commands ---
```

Look at the AND/OR/XOR and Using IF Statements topics to see other examples using the If statements.

If Commands

If Clipboard

The If Clipboard command checks for a value saved to the clipboard.



Which command would you like to use: If Clipboard

Option: If the Clipboard Text Equals

Text: 20/20

☐ Match Uppercase and Lowercase

Script Editor > Expand Logic Category > If Clipboard

Options

Select from one of the four clipboard options available.

If the Clipboard Text Equals

Test if the text contents of the clipboard are equal to a specified value. The contents must match exactly.

If the Clipboard Contains

Test if the text contents of the clipboard contain a specific value. The value only needs to be found within the entire clipboard contents.

If the Clipboard does not Equal

Test if the text contents of the clipboard are not equal to a specified value. The clipboard contents do not match exactly.

If the Clipboard does not Contain

Test if the text contents of the clipboard do not contain a specific value.

Text

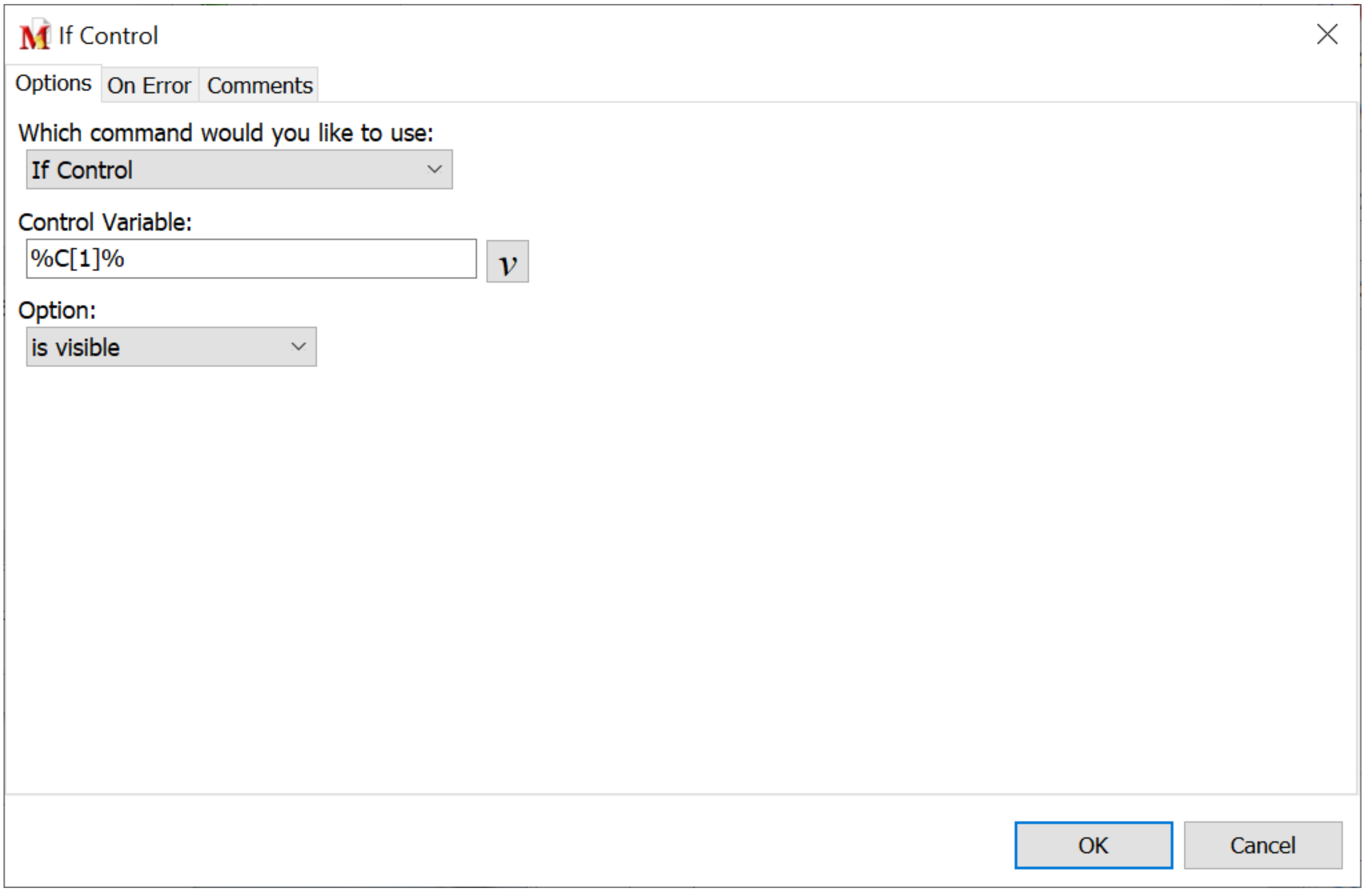
Insert the text to be compared against the clipboard contents.

Match Uppercase and Lowercase

Check this box if the matching text must be case sensitive - the uppercase and lowercase text must match.

If Control

The If Control command is used to determine the status of a Window Control saved to a variable.



Which command would you like to use: If Control

Control Variable: %C[1]%

Option: is visible

Script Editor > Expand Logic Category > If Control

Control Variable

Insert the name of the Control Variable in this field.

Option

There are eight options available for use. Select one from the drop down list.

Is Enabled

This option determines if the window with the control is the active window. In addition it checks to see if the control is enabled, or in other words, if the control is not grayed out.

Is Disabled

This option determines if the window with the control is not the active window. In addition it checks to see if the control is disabled, if the control is grayed out.

Exists

Use this option to determine if the control exists on any window currently open.

Does not Exist

This option determines if the control does not exist on any currently open window.

Is Visible

This option determines if the control is visible. The control does not need to have focus, it just needs to be visible. This control may be a Window Title, a button, a toolbar, an edit box or some other part of the window. A control on a window that is hidden or minimized may still be visible to this command.

Is not Visible

This option determines if the control is not visible.

Is Focused

This option determines if the control has focus.

✓ **Note:** Not all controls are written so that they can receive focus. If this is the case for the control being used, then the Visible or Enabled options would be a better choice. Otherwise the If Focused option will always come back false and the If Not Focused option will always test as true.

Is not Focused

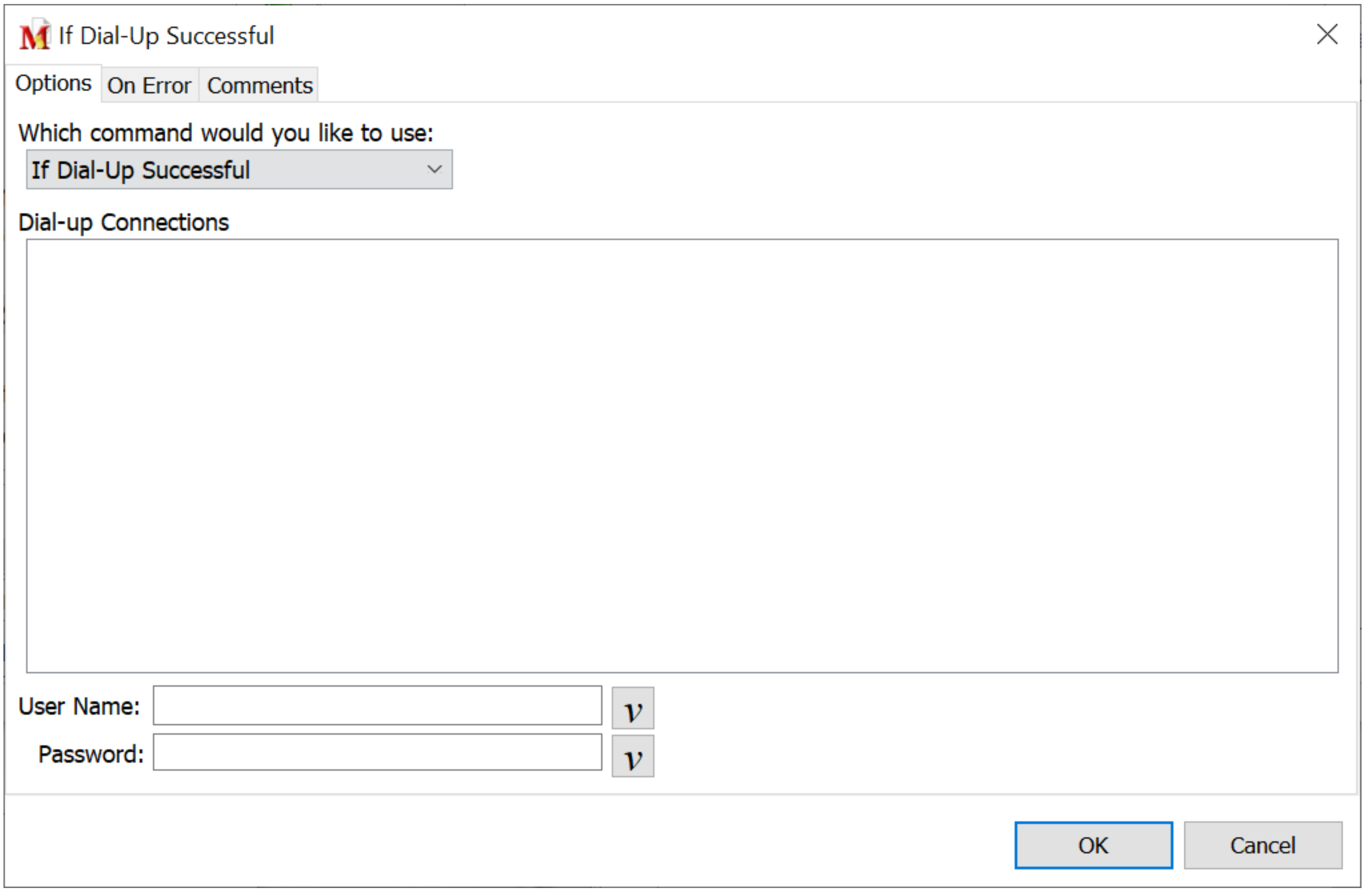
This option determines if the control does not have focus.

If Dial-Up Successful

*Command deprecated - See note below

The If Dial-up Successful feature tests if a dial-up networking attempt to connect to another computer or to the Internet via a modem was successful or not. All commands following this statement until an ENDIF or ELSE command is reached are executed if the dial-up connection is successful. If the dial-up fails and an ELSE command is used, then all commands following the ELSE until an ENDIF is reached will be executed. By placing this inside a "Repeat Until" loop, this command will repeatedly attempt to dial until a successful connection is made.

Dial-up networking must be set up and configured in Windows in order to use this function. Use the Create a Dial-up Connection or Change Dial-up Settings options in Windows to do this.



If Dial-Up Successful

Which command would you like to use: If Dial-Up Successful

Dial-up Connections

User Name:

Password:

Script Editor > Expand Logic Category > If Dial-Up Successful

Phone Book Entries

The Phone Book Entries field displays a list of all the phone book entries created in Windows. If none are displayed, a phone book entry will first need to be created in Windows. Choose the dial up entry.

User Name

Enter the user name for the account selected.

Password

Enter the password for the account selected.

- ✓ **Note:** When selecting a phone book entry, most likely the User Name will be displayed without requiring manual entry. Depending on the security settings the password may also be inserted. The actual password is not shown. A series of asterisks are displayed in its place.

Example

This set of commands instructs the macro to keep trying to dial up and connect until successful. After connecting it loads the www.macros.com site.

We have instructed the macro to repeat until variable T[1] equals 1. If the dial up connection is successful, the variable T[1] is set to 1. This action ends the Repeat loop and loads the www.macros.com site. If the connection was not successful, the variable T[1] is set to 0 and the macro continues repeating until T[1] equals 1.

Sample Macro

```
Repeat Until %T[1]% = "1"
   If Dial-Up Successful: my modem
      Variable Set String %T[1]%"1"
   Else
      Variable Set String %T[1]%"0"
   End If
Repeat End
WebSite: https://www.macros.com
```

*Deprecated Commands

Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

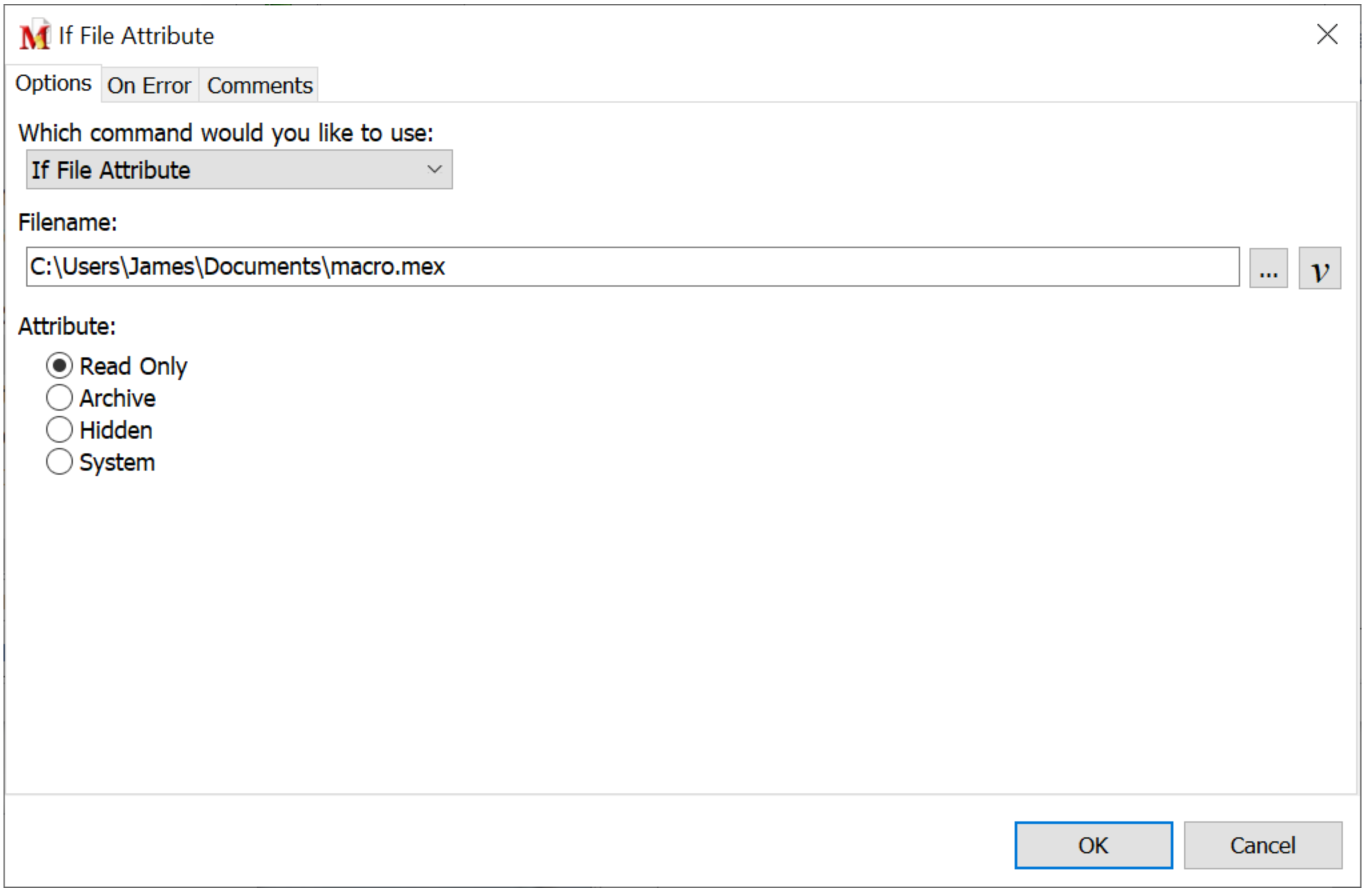
Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future.

Where possible we recommend that you avoid the use of deprecated macro commands.

By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option found in Options, Preferences, General - Misc tab.

If File Attribute

When using My Computer or Windows Explorer, right click on a file name to open the menu. Left mouse click on the Properties menu item to see more details about the file. The lower portion of the Properties window displays some or all of the following File Attributes - Archive, System, Hidden and Read Only.



Which command would you like to use: If File Attribute

Filename: C:\Users\James\Documents\macro.mex

Attribute:

- ● Read Only
- ○ Archive
- ○ Hidden
- ○ System

Script Editor > Expand Logic Category > If File Attribute

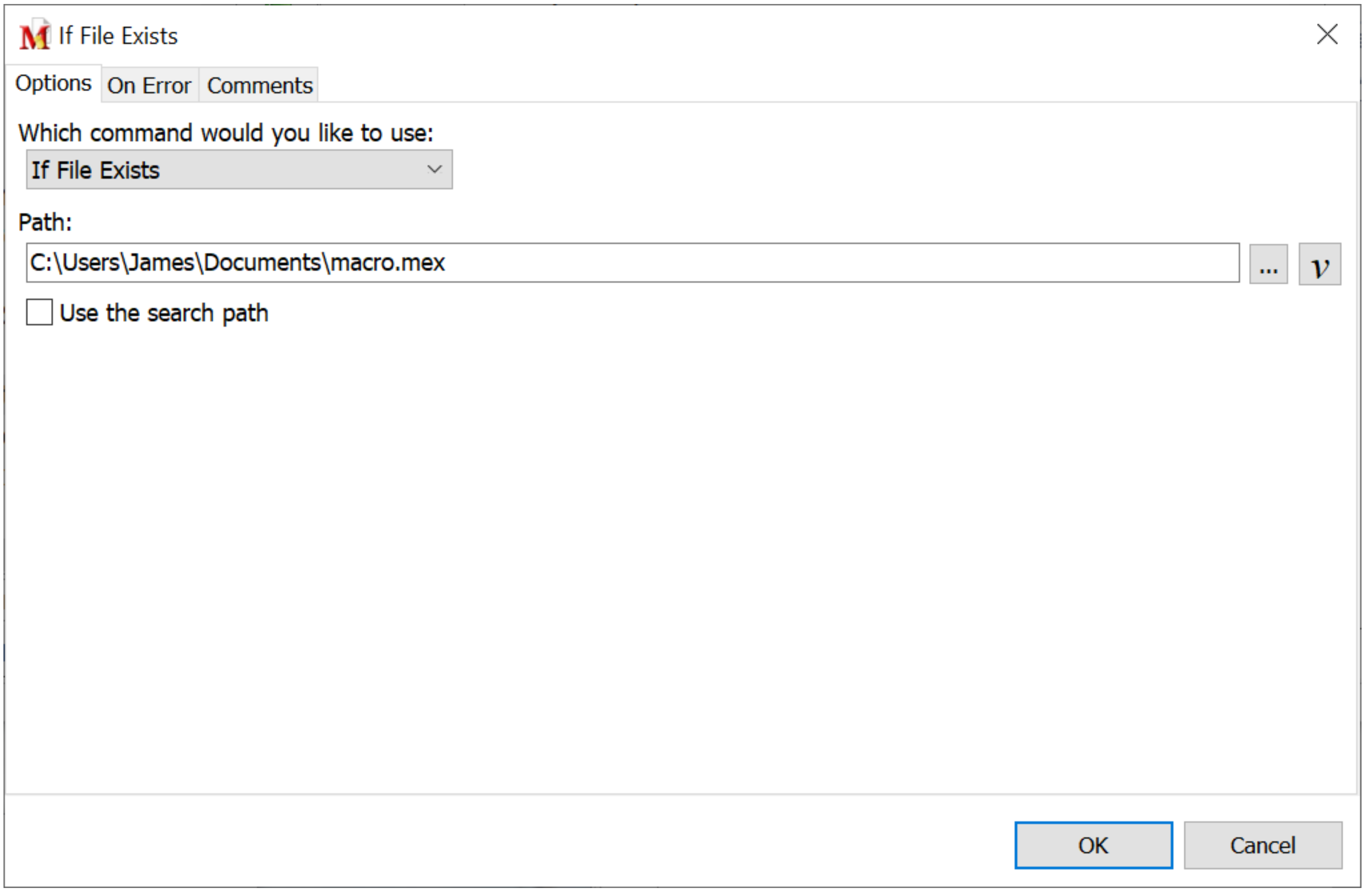
The If File Attribute command is used to test the status of a file attribute. First select or enter the file name and path in the Filename field. Then select the attribute to test.

For example, check to see if the file is marked as Read Only. If it is, remove the Read Only flag by using the Set File Attributes command.

If File Exists

This command tests if the selected file exists on the computer.

✓ **Note:** This command does not support wildcards.



Which command would you like to use: If File Exists

Path: C:\Users\James\Documents\macro.mex

□ Use the search path

Script Editor > Expand Logic Category > If File Exists

Path

Insert the file path in this field or click on the Browse button to locate the file to test.

Use the Search Path

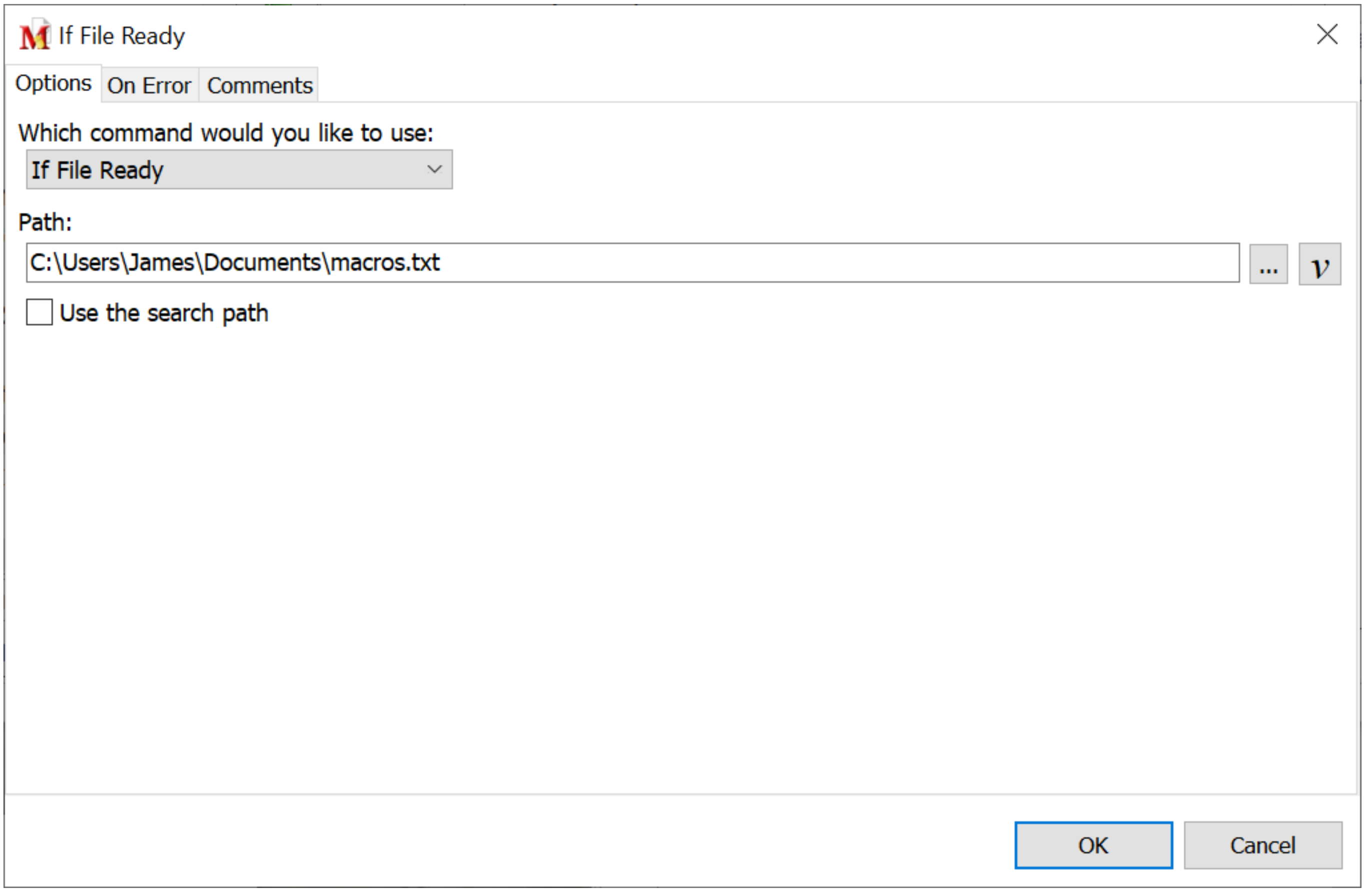
Select this option when only inserting a partial file path or only a file name such as test.txt. In this case the macro attempts to locate the file using the Windows search path stored in the Environment variables.

If the Use the Search Path option is not selected and only the file name is inserted, then the macro will look in the Macro Express default folder. Use the Change Default Folder command to set the default folder.

If File Ready

This command tests if the file selected is no longer busy and ready to accept input. During macro playback it's possible that another application is still writing to it. The If File Ready command checks to ensure the file is no longer being written to.

- ✓ **Note:** If the file does not have write permissions then the macro will always flag the file as not being ready. In Windows Vista and later, any files located in the Program Files folder do not have write permission.



Which command would you like to use: If File Ready

Path: C:\Users\James\Documents\macros.txt

□ Use the search path

Script Editor > Expand Logic Category > If File Ready

Path

Insert the file path in this field or click on the Browse button to locate the file to test.

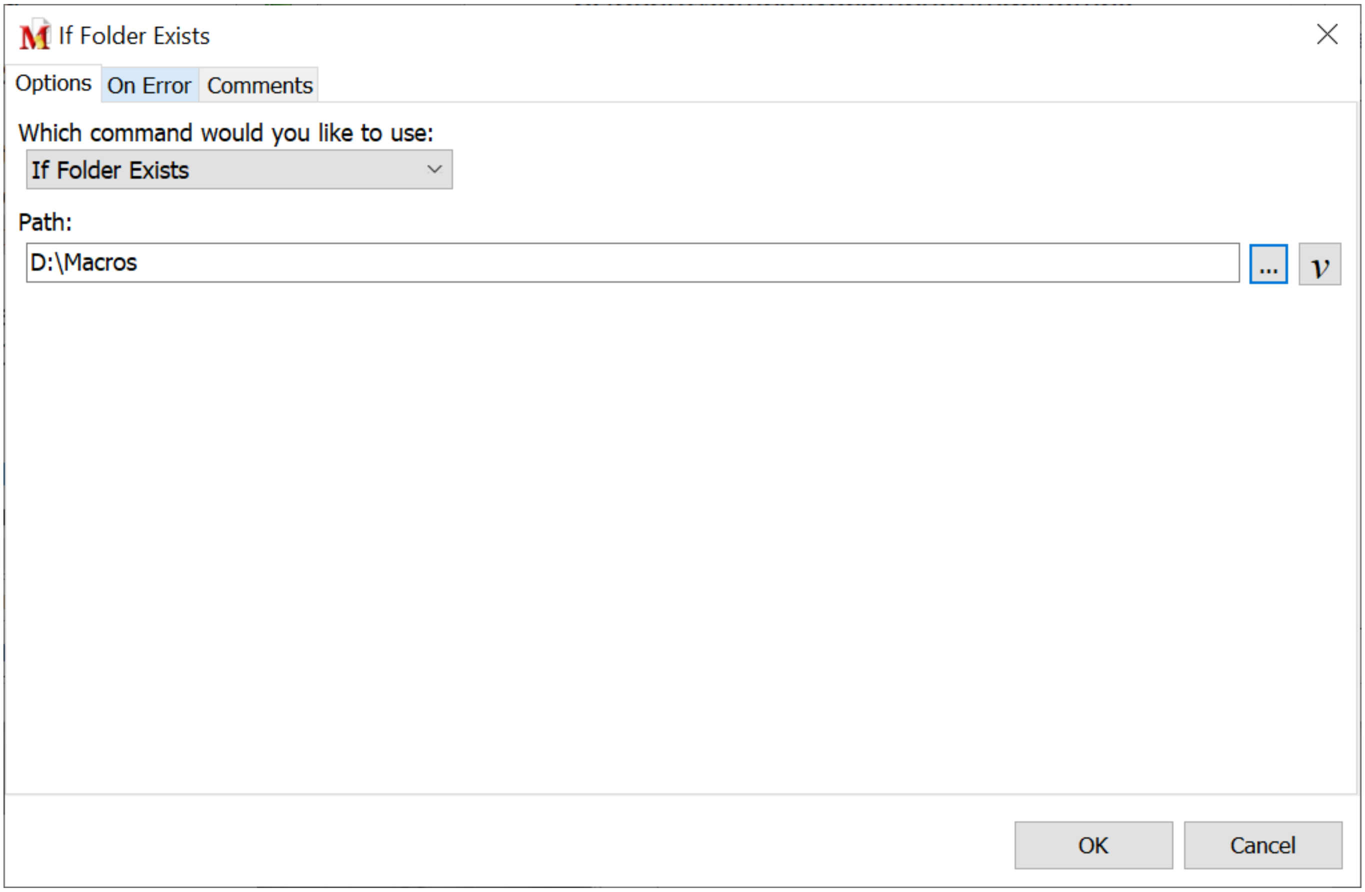
Use the Search Path

Select this option when only inserting a partial file path or only a file name such as test.txt. In this case the macro attempts to locate the file using the Windows search path stored in the Environment variables.

If the Use the Search Path option is not selected and only the file name is inserted, then the macro will look in the Macro Express default folder. Use the Change Default Folder command to set the default folder.

If Folder Exists

This command tests if the folder selected exists on the computer.



Which command would you like to use: If Folder Exists

Path: D:\Macros

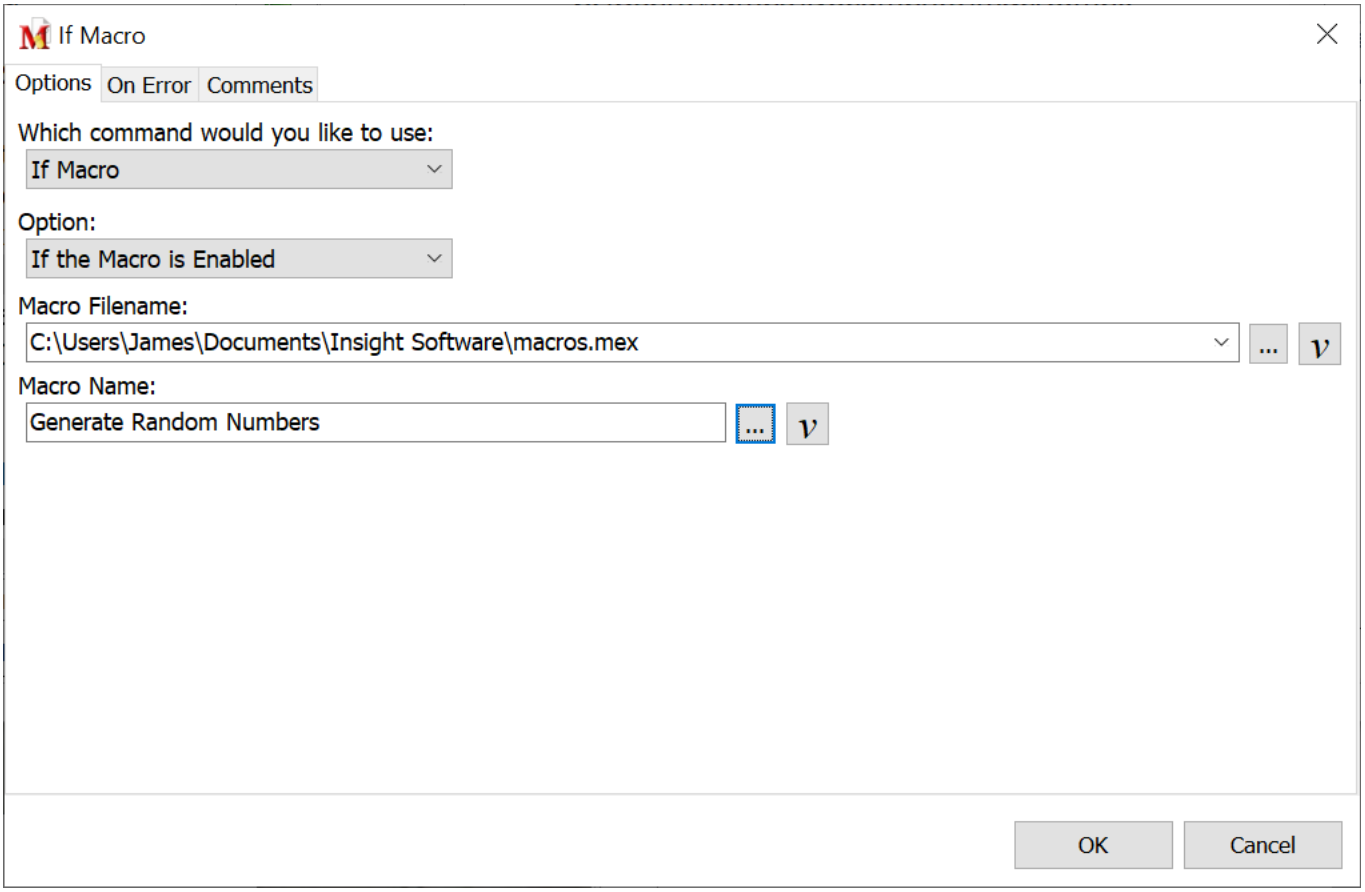
Script Editor > Expand Logic Category > If Folder Exists

Path

Insert the path to the folder in this field or click on the Browse button to locate the folder to test.

If Macro

The If Macro command tests the status of a macro. It tests if the macro is enabled or disabled, or if the requested macro even exists.



Which command would you like to use: If Macro

Option: If the Macro is Enabled

Macro Filename: C:\Users\James\Documents\Insight Software\macros.mex

Macro Name: Generate Random Numbers

Script Editor > Expand Logic Category > If Macro

Option

There are four options available.

If the Macro is Enabled

This option checks if the macro state is set to Enabled.

If the Macro is Disabled

This option checks if the macro state is set to Disabled.

If the Macro Exists

This option checks if the selected macro is located on the computer.

If the Macro Does Not Exist

This option determines if the macro is not found on the computer.

Macro Filename

Insert the name of the macro file (.mex file) that contains the macro to test. Or click on the Browse button to locate the macro file.

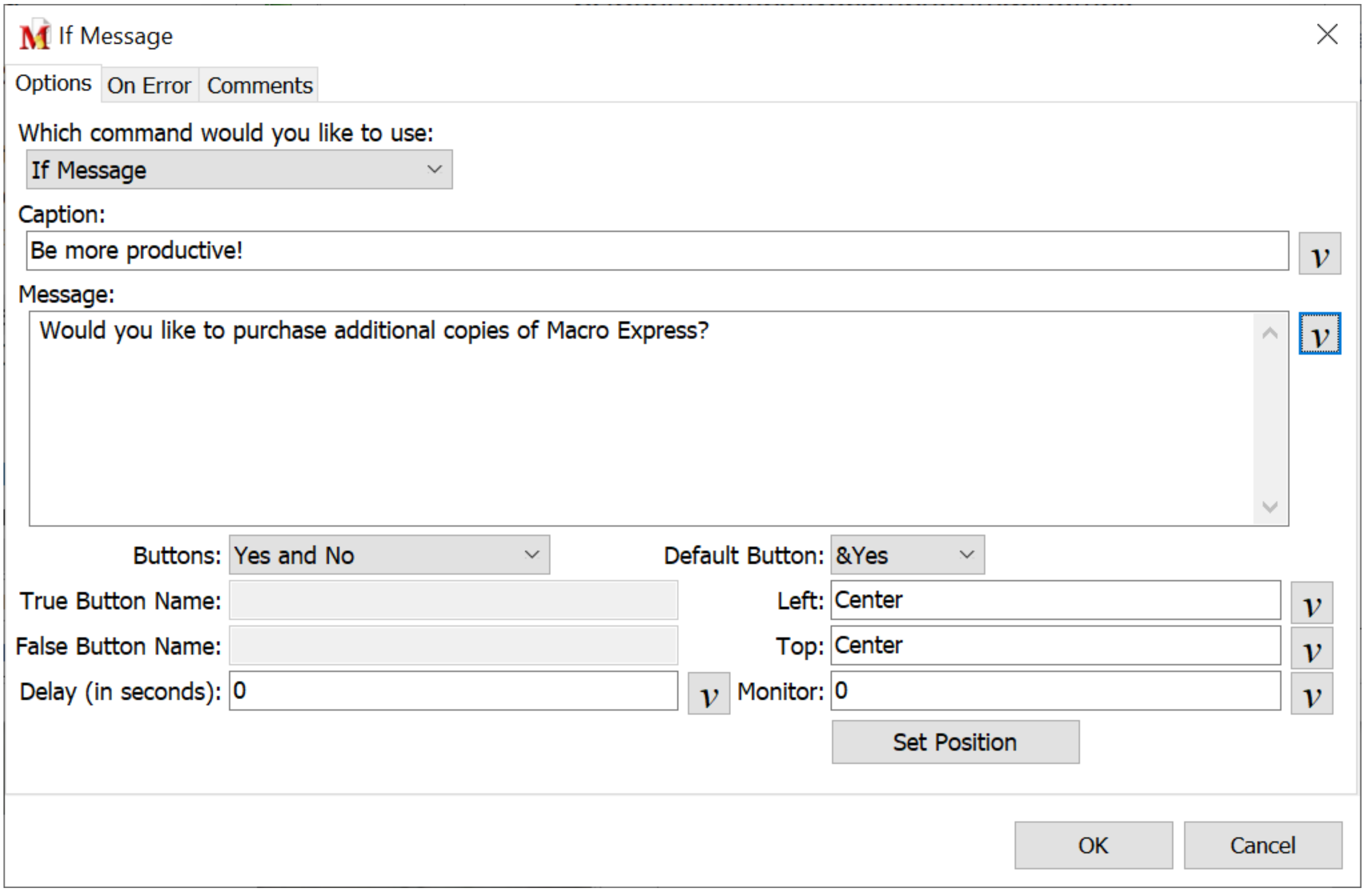
Macro Name

Insert the macro name or click on the Browse button to locate the name of the macro to test.

If Message

The If Message Command prompts users with a question during macro playback. The question that appears in the dialog box also displays two buttons below the question. The buttons can be set to Yes/No, OK/Cancel, True/False or User defined. This statement requires an ENDIF statement and can optionally use the ELSE statement.

Text Variables may be placed in the input fields of this screen. For example, place the variable T[1] in the message. To do this enter %T[1]% into the Message field.



Which command would you like to use: If Message

Caption: Be more productive!

Message: Would you like to purchase additional copies of Macro Express?

Buttons: Yes and No

Default Button: &Yes

True Button Name:

False Button Name:

Delay (in seconds): 0

Left: Center

Top: Center

Monitor: 0

Script Editor > Expand Logic Category > If Message

Caption

The caption is the text that appears in the Window Title bar of the message window created.

Message

Insert the message or question to be posed into this field.

Buttons

Select from four different button types. These are the labels for the buttons that appear with the message during macro playback. The button names selected should correspond with the type of question asked.

Yes/No
OK/Cancel
True/False
User Defined

The User Defined option creates custom labels, for use languages other than English, etc.

Default Button

Select which button is the default button when the message is displayed. The default button is highlighted so that pressing the Enter key will make the selection, rather than clicking with the mouse.

Delay in Seconds

If a choice is not made within the number of seconds entered in this field, the macro continues by selecting the Default Button mentioned above.

Positioning the Message Window

Position the message box on the screen. Insert exact coordinates or use the Set Position Manually button to set where the window should appear during macro playback. See the Window Positioning topic for additional information.

Sample Macro

We'll create a simple example of how this command works. The If Message was created using the image shown above in this help topic.

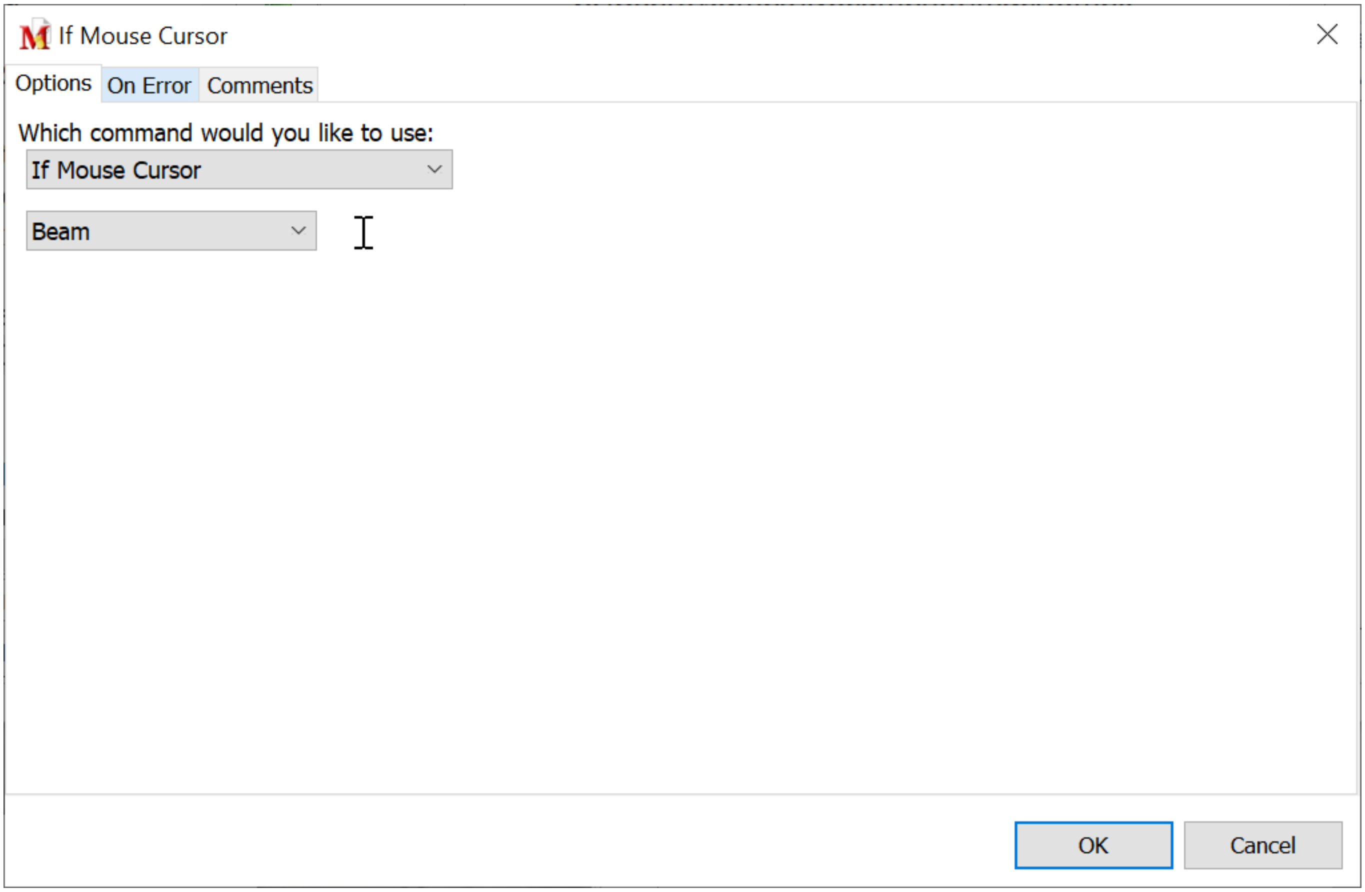
When running the macro and you answer yes to the question, "Would you like to purchase additional copies of Macro Express?", you are taken to the order page on the Macro Express web site. If you answer no, then a text box displays with the message "Maybe I should rethink my answer".

Macro Script

```
If Message: "Become more Productive!"
  Web Site: https://www.macros.com/order.htm
Else
  Text Box Display: Macro Express
End If
```

If Mouse Cursor

This command checks the status of the mouse cursor. The macro takes action based on whether or not the status of the mouse cursor matches the cursor type selected. Choose from the available mouse cursor options.



Which command would you like to use:
If Mouse Cursor

Beam

Script Editor > Expand Logic Category > If Mouse Cursor

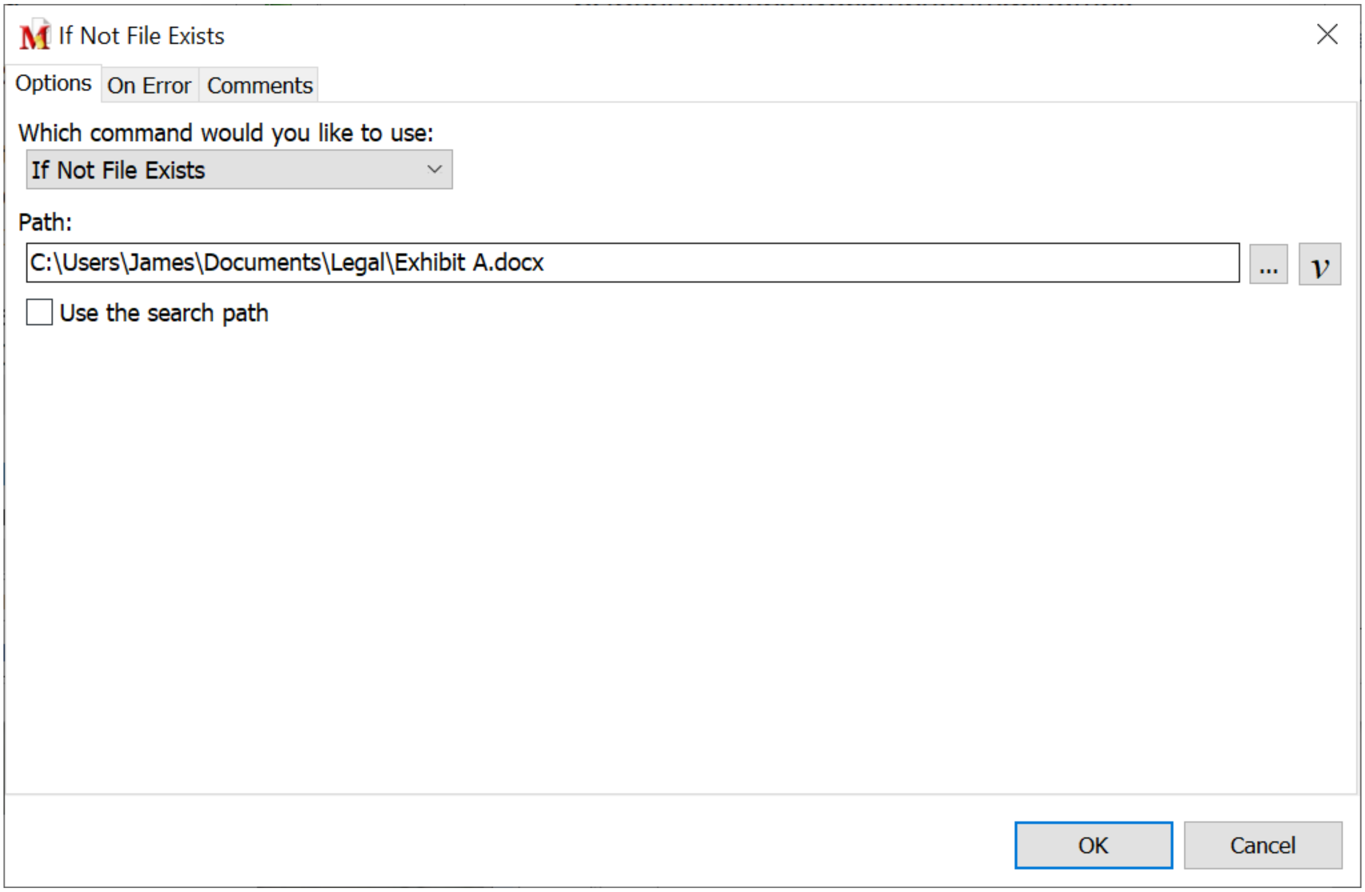
Type of Cursor

Select the mouse cursor from the drop down list. The name and a graphical representation of the cursor are displayed to ensure selection of the correct cursor type.

If Not File Exists

This command determines if the file selected does not exist on the computer.

- ✓ **Note:** This command does not support wildcards.



Which command would you like to use: If Not File Exists

Path: C:\Users\James\Documents\Legal\Exhibit A.docx

□ Use the search path

Script Editor > Expand Logic Category > If Not File Exists

Path

Insert the file path in this field or click on the Browse button to locate the file to test.

Use the Search Path

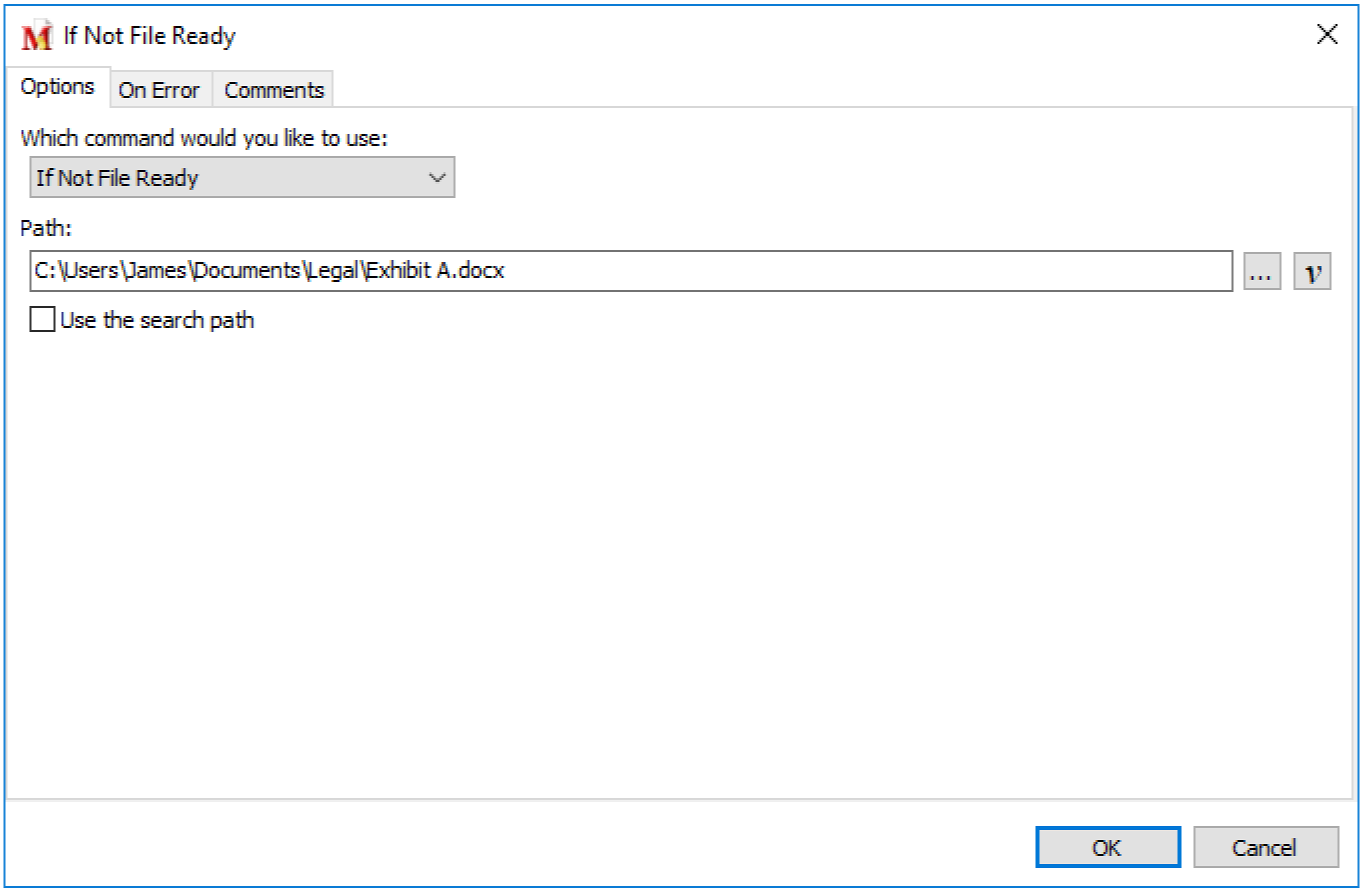
Select this option when only inserting a partial file path or only a file name such as test.txt. In this case the macro attempts to locate the file using the Windows search path stored in the Environment variables.

If the Use the Search Path option is not selected and only the file name is inserted, then the macro will look in the Macro Express default folder. Use the Change Default Folder command to set the default folder.

If Not File Ready

This command determines if the file selected is not ready to accept input. It checks if the file is still busy. During macro playback it's possible that another application is still writing to it. The If Not File Ready command checks if the file is still being written to.

- ✓ **Note:** If the file does not have write permissions then the macro will always flag the file as not being ready. In Windows Vista and later, any files located in the Program Files folder do not have write permission.



Which command would you like to use: If Not File Ready

Path: C:\Users\James\Documents\Legal\Exhibit A.docx

□ Use the search path

Script Editor > Expand Logic Category > If Not File Ready

Path

Insert the file path in this field or click on the Browse button to locate the file to test.

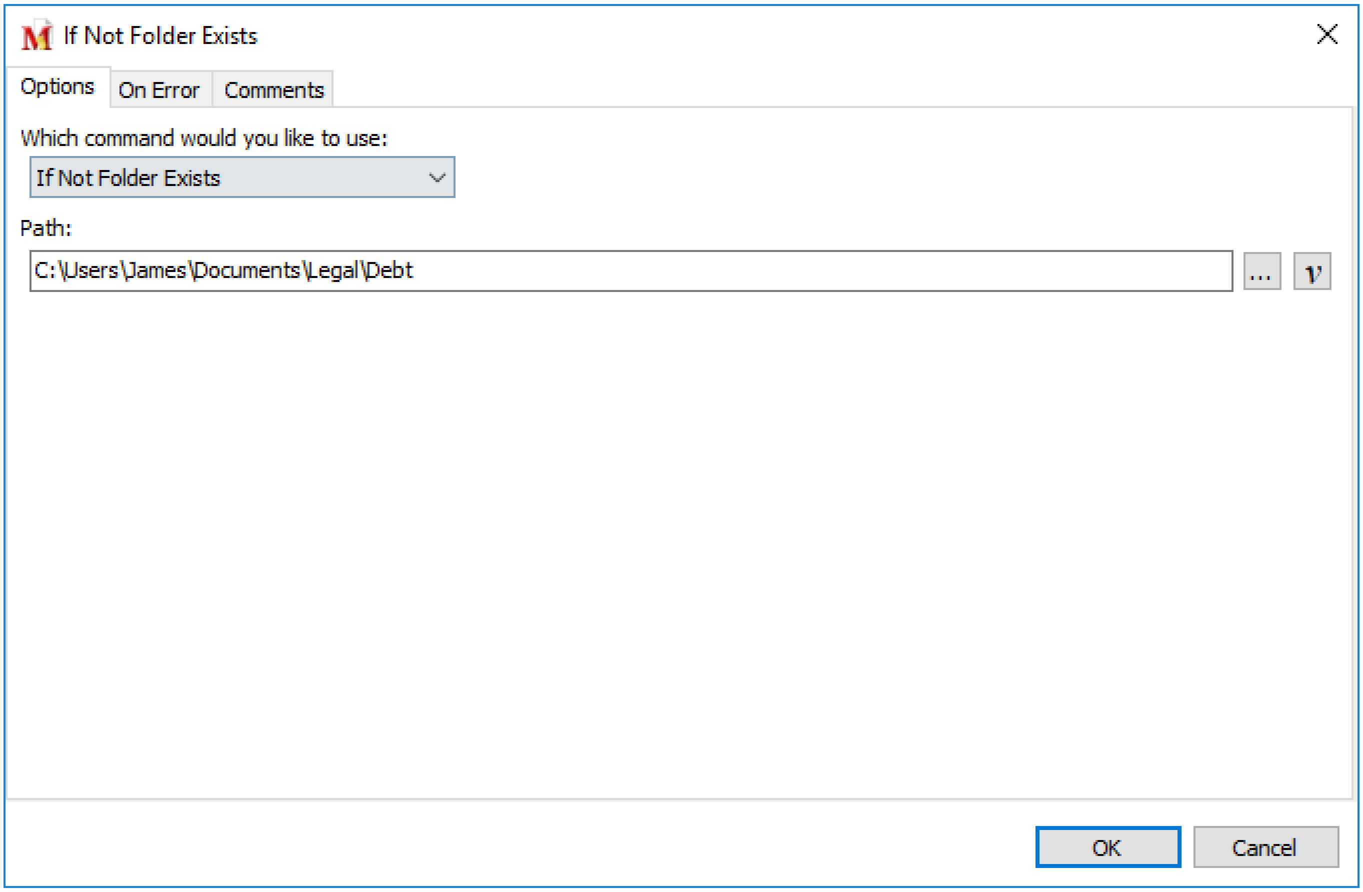
Use the Search Path

This option can be used when not inserting a full file path, only a file name such as test.txt. With this option selected, the macro attempts to locate the file using the Windows search path stored in the Environment variables.

If this option is not selected and only the file name is inserted, then the macro will look in the Macro Express default folder. Use the Change Default Folder command to set the default folder.

If Not Folder Exists

This command determines if the folder selected does not exist on the computer.



Which command would you like to use: If Not Folder Exists

Path: C:\Users\James\Documents\Legal\Debt

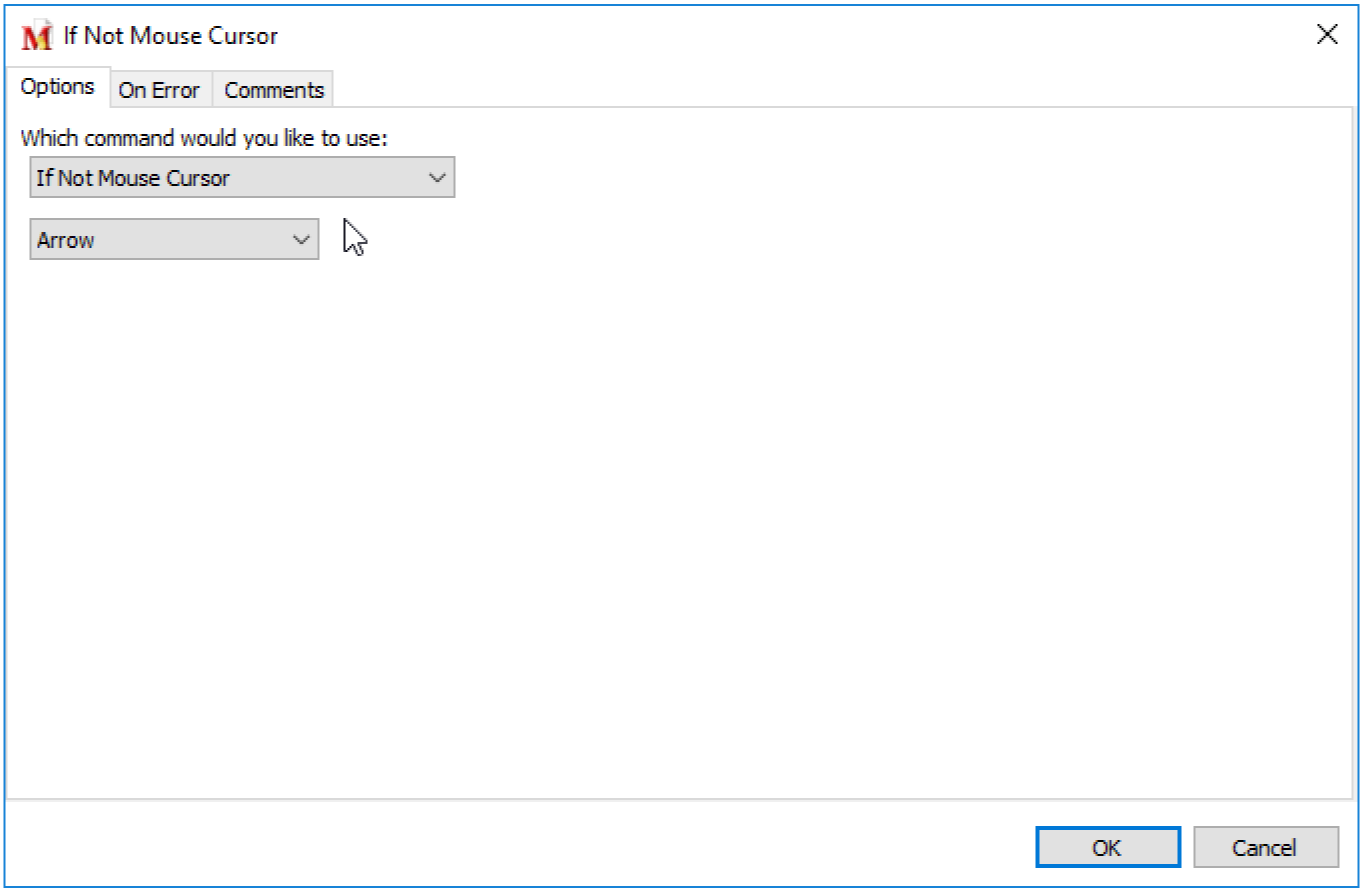
Script Editor > Expand Logic Category > If Not Folder Exists

Path

Insert the path to the folder in this field or click on the Browse button to locate the folder to test.

If Not Mouse Cursor

This command is essentially the opposite of the If Mouse Cursor command. It checks the status of the mouse cursor, but takes action based on whether the status of the mouse cursor does not match the selection chosen. Choose from twenty one different mouse cursor options.



Which command would you like to use: If Not Mouse Cursor

Arrow

Script Editor > Expand Logic Category > If Not Mouse Cursor

Type of Cursor

Select the mouse cursor from the drop down list. The name as well as a graphical representation of the cursor are displayed to ensure selection of the correct cursor type.

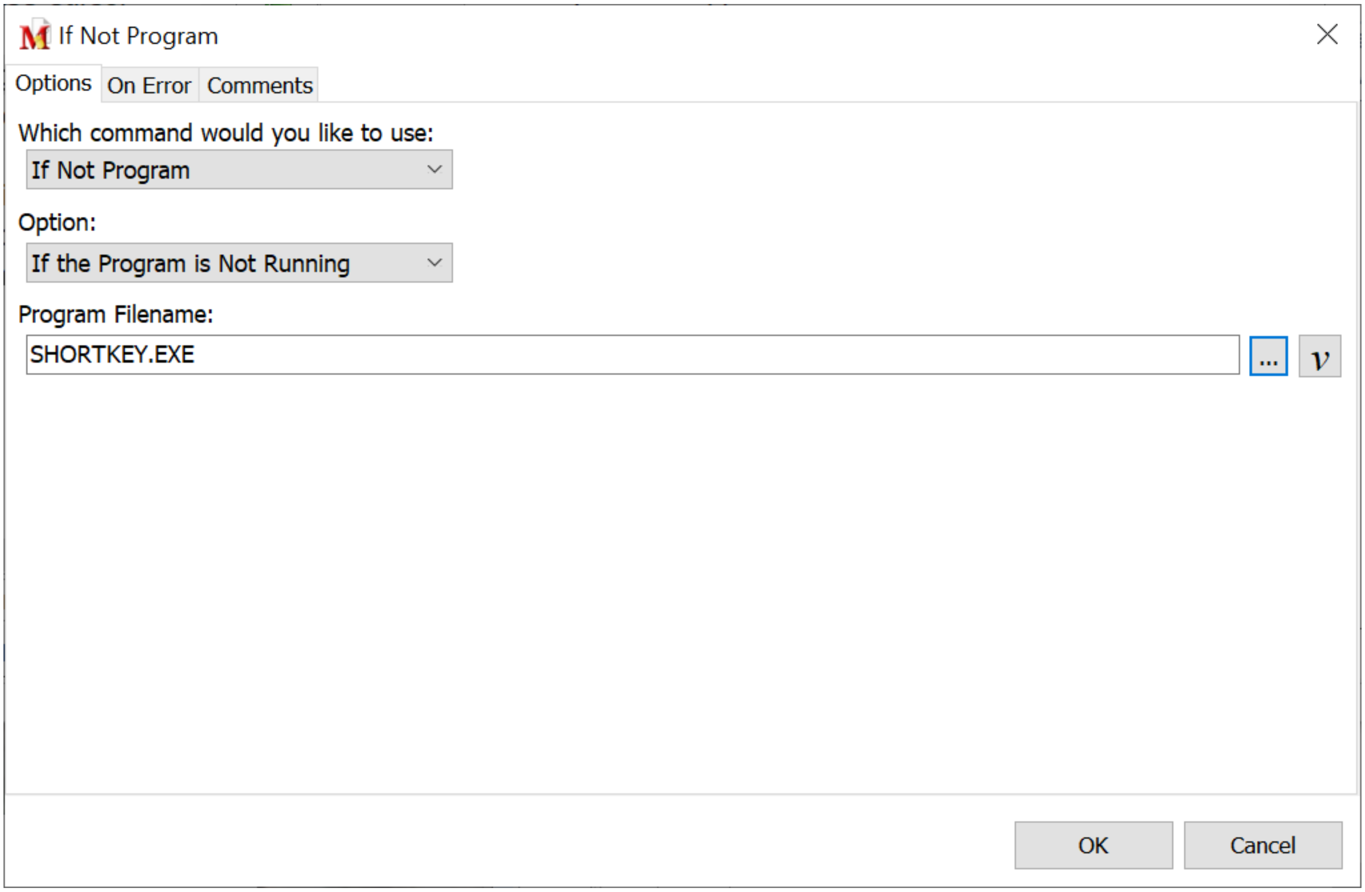
If Not Online

The If Not Online command tests if there is currently not a connection to the Internet.

There is not a program window associated with this command. The If Not Online command is inserted directly into the script. In the Script Editor, click on the Logic command category to expand the list and locate the If Not Online command.

If Not Program

This command is similar to the If Not Window command. It checks if the specified program is not on top (does not have focus) or if the program is not running.



Which command would you like to use: If Not Program

Option: If the Program is Not Running

Program Filename: SHORTKEY.EXE

Script Editor > Expand Logic Category > If Not Program

Option

Select the option from the drop down list.

If the Program Is Not Focused

Select this option to test if the selected program window does not have focus.

If the Program Is Not Running

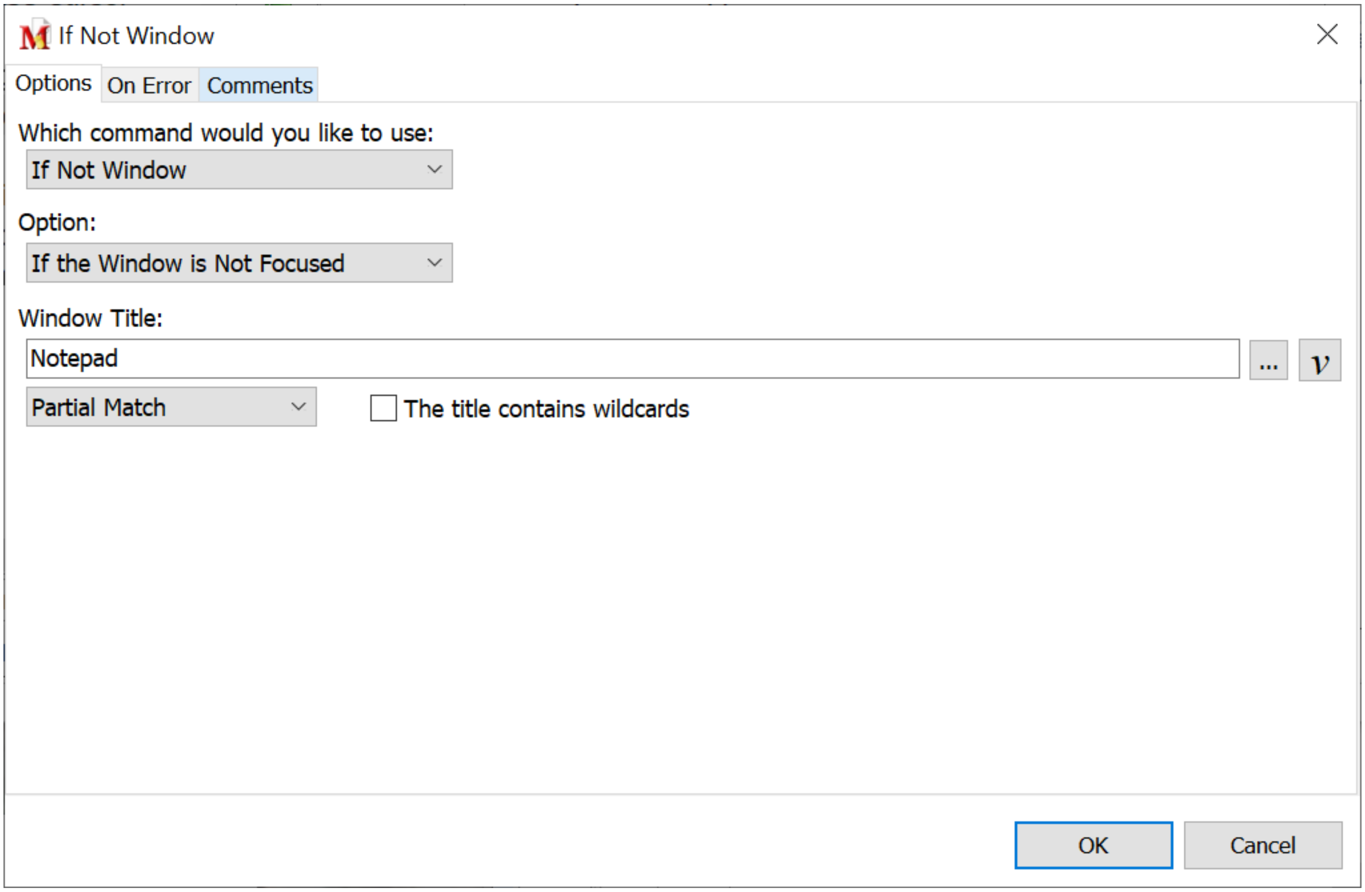
Select this option to test if the selected program is currently not running.

Program Filename

Insert the name of the program and path to the executable in this field or click on the Browse button to navigate to the file.

If Not Window

There are two If Not Window options available. One checks to see if the window is not on top (does not have focus) or if the window is currently not open.



Which command would you like to use: If Not Window

Option: If the Window is Not Focused

Window Title: Notepad

Partial Match

□ The title contains wildcards

Script Editor > Expand Logic Category > If Not Window

Option

Select the option from the drop down list.

If the Window Is Not Focused

Select this option to test if the selected window does not have focus.

If Window Does Not Exist

Select this option if to test if the selected window is not found (not open on the computer).

Window Title

Insert the name of the window to test or click on the Browse button to select from the Hidden or Visible windows currently running on your computer.

Partial Match/Exact Match

Select whether the window title must match exactly or if a partial match is sufficient.

Exact Match

The Exact Match requires that the window name entered matches exactly with the name of the window used in the If statement. For example, if using Notepad as the window name, but the actual window open when the macro runs is Notepad - Untitled, then the macro would ignore the If statement.

Partial Match

An exact window title match is not required. For example, insert "notepad" or "notep" as the window title and either would be recognized by the If statement. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

Select this option if using a wildcard in the window title name. A wildcard includes the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

If Online

The If Online command tests if there is a connection to the Internet.

There is not a program window associated with this command. The If Online command is inserted directly into the script. In the Script Editor, click on the Logic command category to expand the list and locate the If Online command.

Example

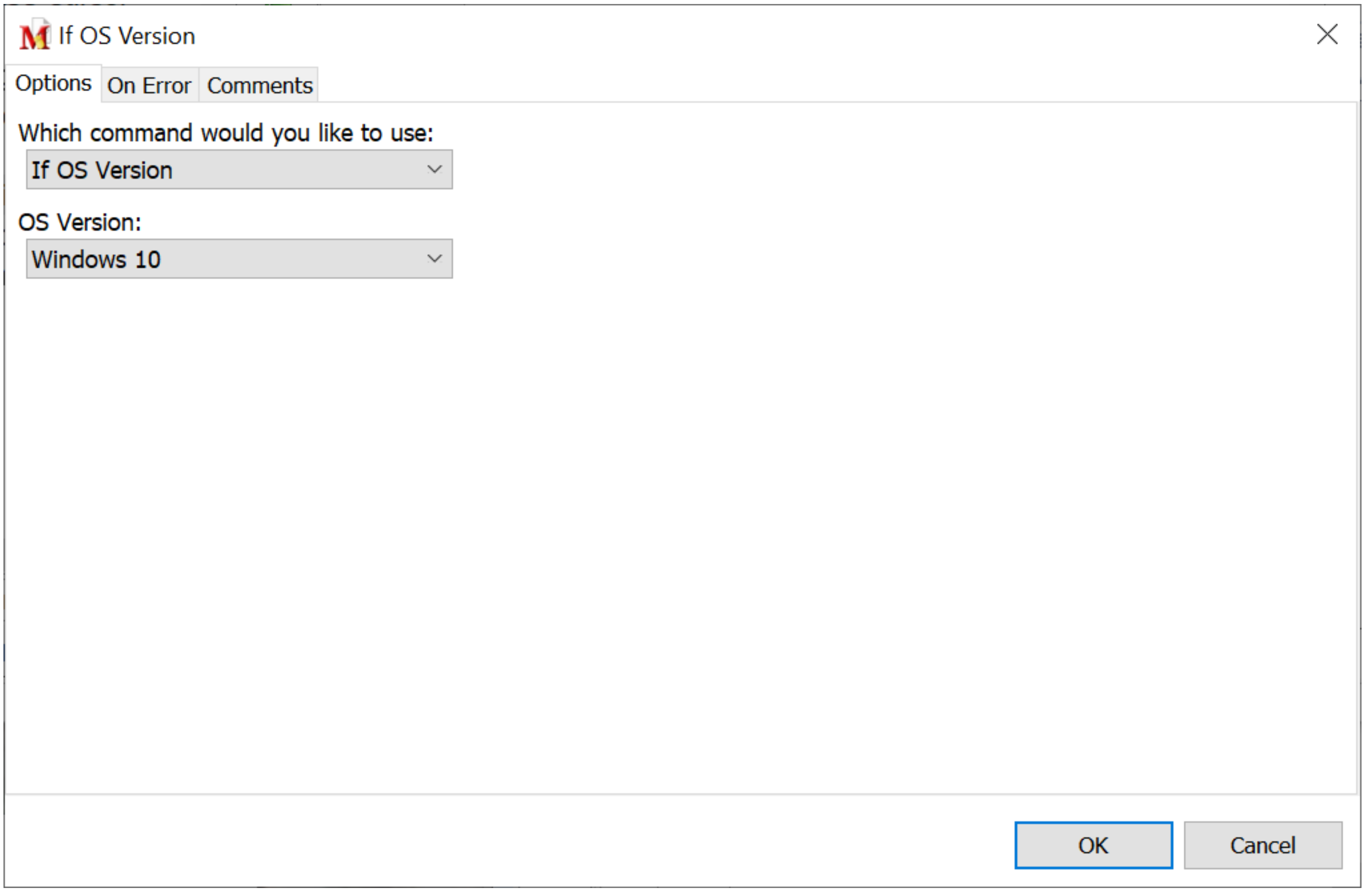
This macro simply loads the web site www.macros.com if connected to the Internet. If not, the macro will dial, connect to the Internet and open the web page.

Sample Macro

```
If Online
  Web Site, "https://www.macros.com", using Default Web
Browser
Else
  Dial-Up Networking: My Modem
  Web Site, "https://www.macros.com", using Default Web
Browser
End If
```

If OS Version

Not all Microsoft operating systems work the same. Some things function differently in each OS. The "If OS Version" command can help with this problem. A macro can be tailored to the operating system of the user.



Which command would you like to use: If OS Version

OS Version: Windows 10

Script Editor > Expand Logic Category > If OS Version

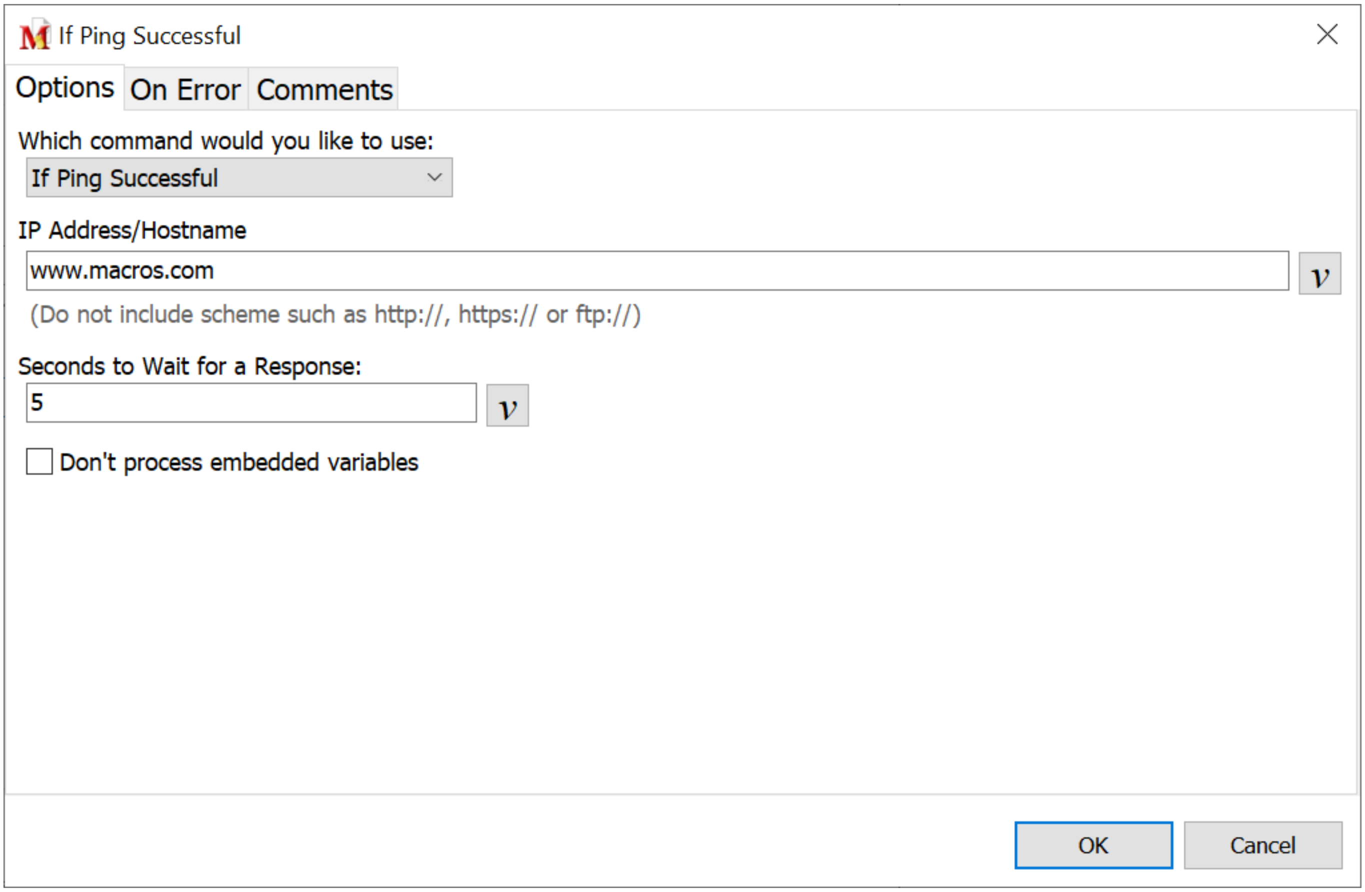
In the sample below we created a simple macro that does something different based on the operating system in use. This is how it would be displayed in the Scripting Editor. This macro basically states that if running Windows Server 2012 or 2016, then run the macro named Server. The OR command allows for selection between two or more If commands. If running Windows 7, 8.1 or 10, run the macro named Win10.

Sample Macro

```
If OS Version is Windows Server 2012
  Or
If OS Version is Windows Server 2016
  Macro Run: Server
Else
  If OS Version is Windows 7
    Or
  If OS Version is Windows 8.1
    Or
  If OS Version is Windows 10
    Macro Run: Win10
  End If
End If
```

If Ping Successful

This option first attempts to ping the server designated. If the server responds within the specified time frame, then the result will be true.



Which command would you like to use: If Ping Successful

IP Address/Hostname: www.macros.com

(Do not include scheme such as http://, https:// or ftp://)

Seconds to Wait for a Response: 5

□ Don't process embedded variables

Script Editor > Expand Logic Category > If Ping Successful

IP Address/Hostname

Insert the IP Address or website to ping. Do not include modifiers such as http:// or ftp://. If pinging our Macro Express website, insert www.macros.com.

Seconds to Wait for a Response

Insert the number of seconds to wait before determining if the ping is successful.

Don't process embedded variables

When using a variable in the address field, this option ignores any % signs found within the variable value and will not attempt to treat the % signs as an embedded variable. It will just expand the % sign as a standard character during macro runtime.

Example

Assume for example a variable of %T[1]% with a value of www.%DomainName%.com

1. With this option selected, the macro will ping the site found at www.%DomainName%.com

2. If the option is not selected, then the macro will assume that there is a variable of %DomainName% and look to expand its value.

   a. If the variable %DomainName% does not exist, then the macro will generate an error because it could not find the value for this variable.

   b. Otherwise it will expand the value of %DomainName% within the main variable.

Example

Here is a simple example of how to use this command. The macro is pinging the www.macros.com site to make sure that the server is running.

- If the ping is successful a message is logged to the ping.log file with a date and time stamp.
- If the server is down and the ping was not successful, an email is sent for someone to look into the problem.

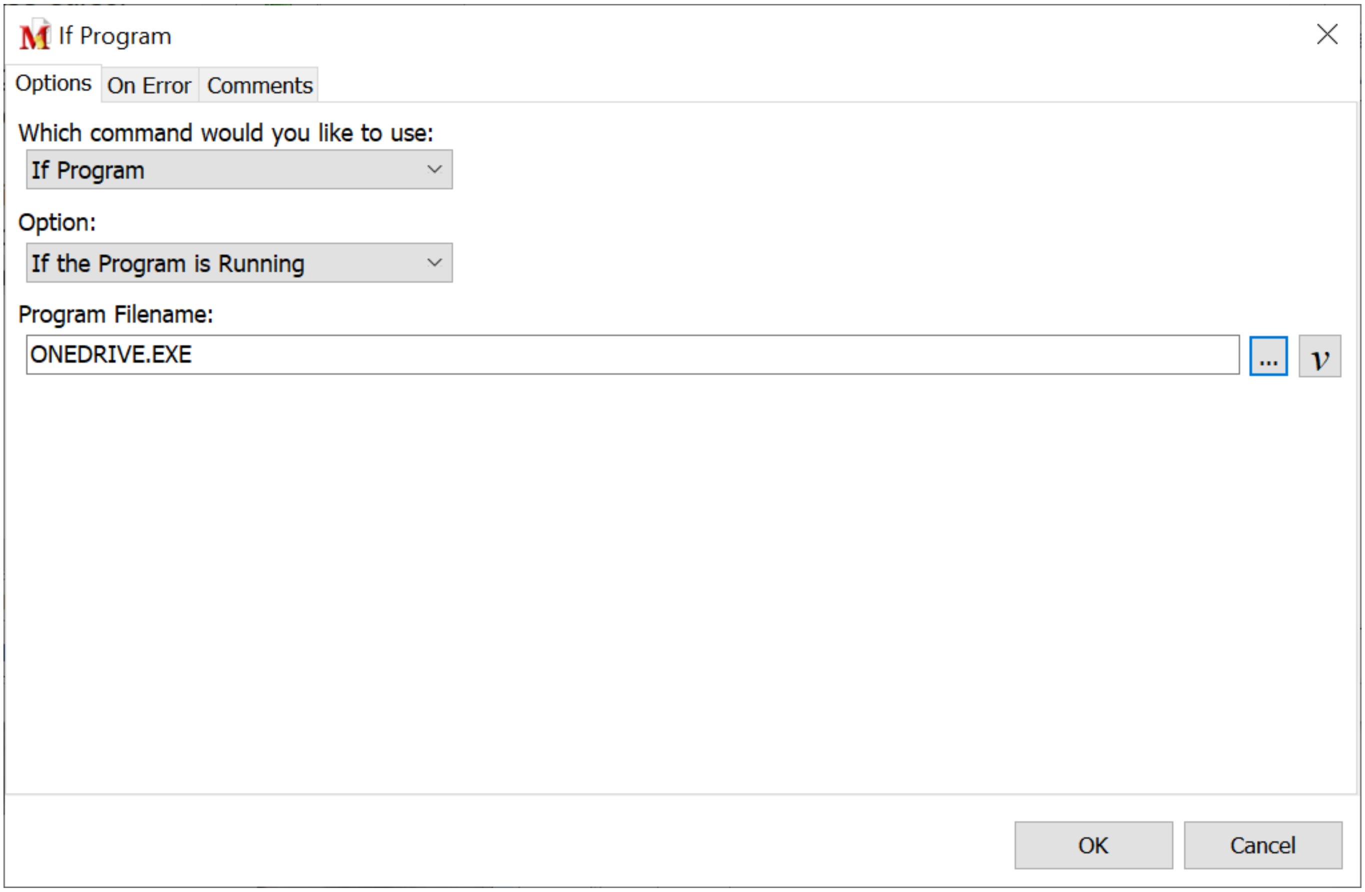
This could be set up as a scheduled macro to run every so often.

Sample Macro

```
If Ping Successful: "www.macros.com"
  Log Message to "ping.log"
Else
  E-Mail Send: Know Recipients, Fixed Text, No Attachments
End If
```

If Program

This command is similar to the If Window command. It checks if the specified program is on top (has focus) or if the program is running.



Which command would you like to use: If Program

Option: If the Program is Running

Program Filename: ONEDRIVE.EXE

Script Editor > Expand Logic Category > If Program

Option

Select the option from the drop down list.

If Program On Top

Select this option to test if the selected program has focus.

If Program Running

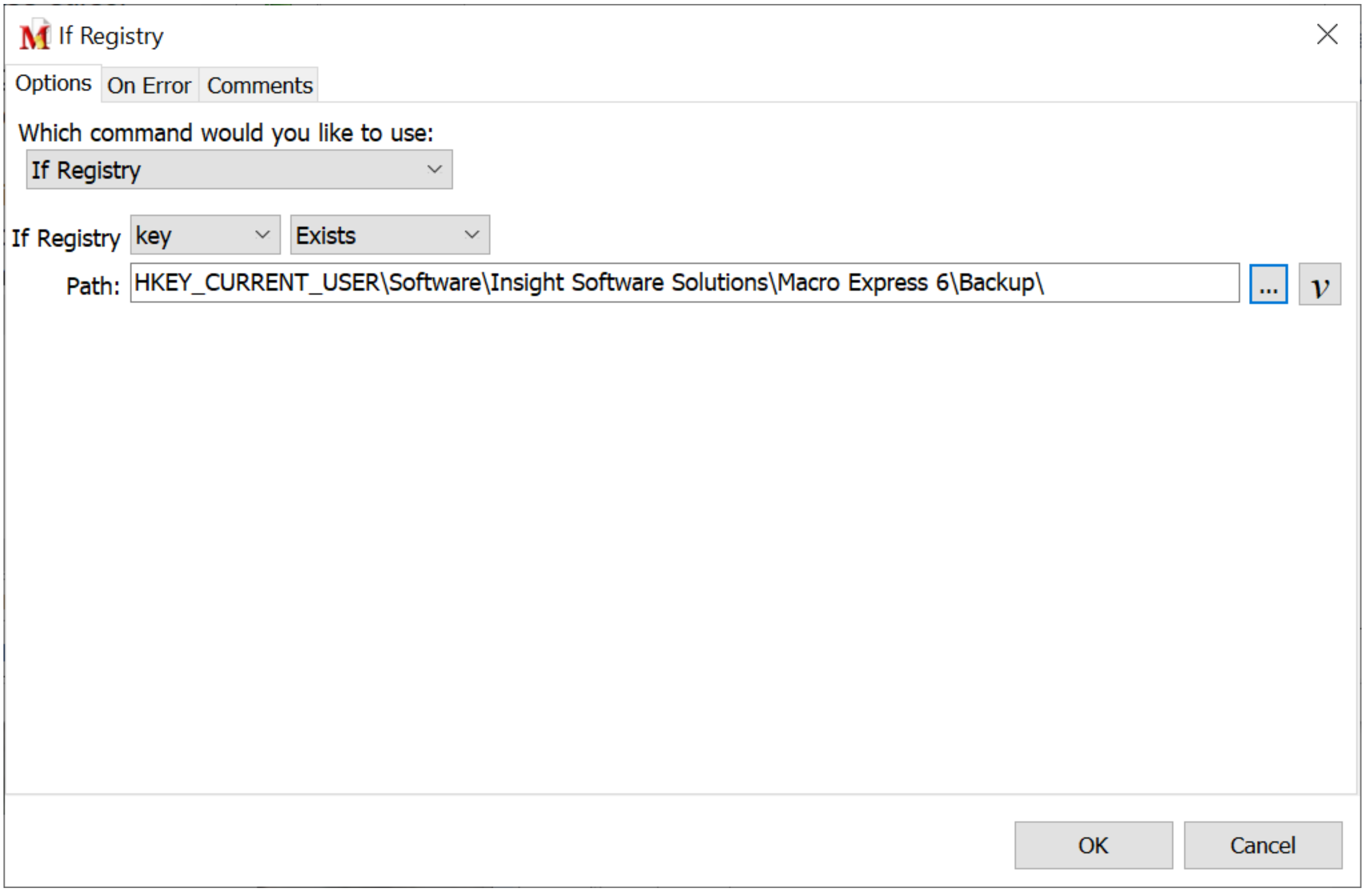
Select this option to test if the selected program window is currently running. It does not have to be the program that is on top. The application may be running in the background.

Program Filename

Insert the name of the program and path to the executable in this field or click on the Browse button to navigate to the file.

If Registry

This command determines whether a Windows Registry Key or Value exists or does not exist.



If Registry
Options
On Error
Comments
Which command would you like to use:
If Registry
If Registry
key
Exists
Path:
HKEY_CURRENT_USER\Software\Insight Software Solutions\Macro Express 6\Backup\
...
v
OK
Cancel

Script Editor > Expand Logic Category > If Registry

If Registry

This section consists of two parts. Select either registry value or registry key. Then select the If Exists or Does not Exist option. The macro determines if the registry key or value exists or does not exist on the computer.

Path

Select the registry path. Either type in the full path or click on the Browse button to navigate to the key or value needed.

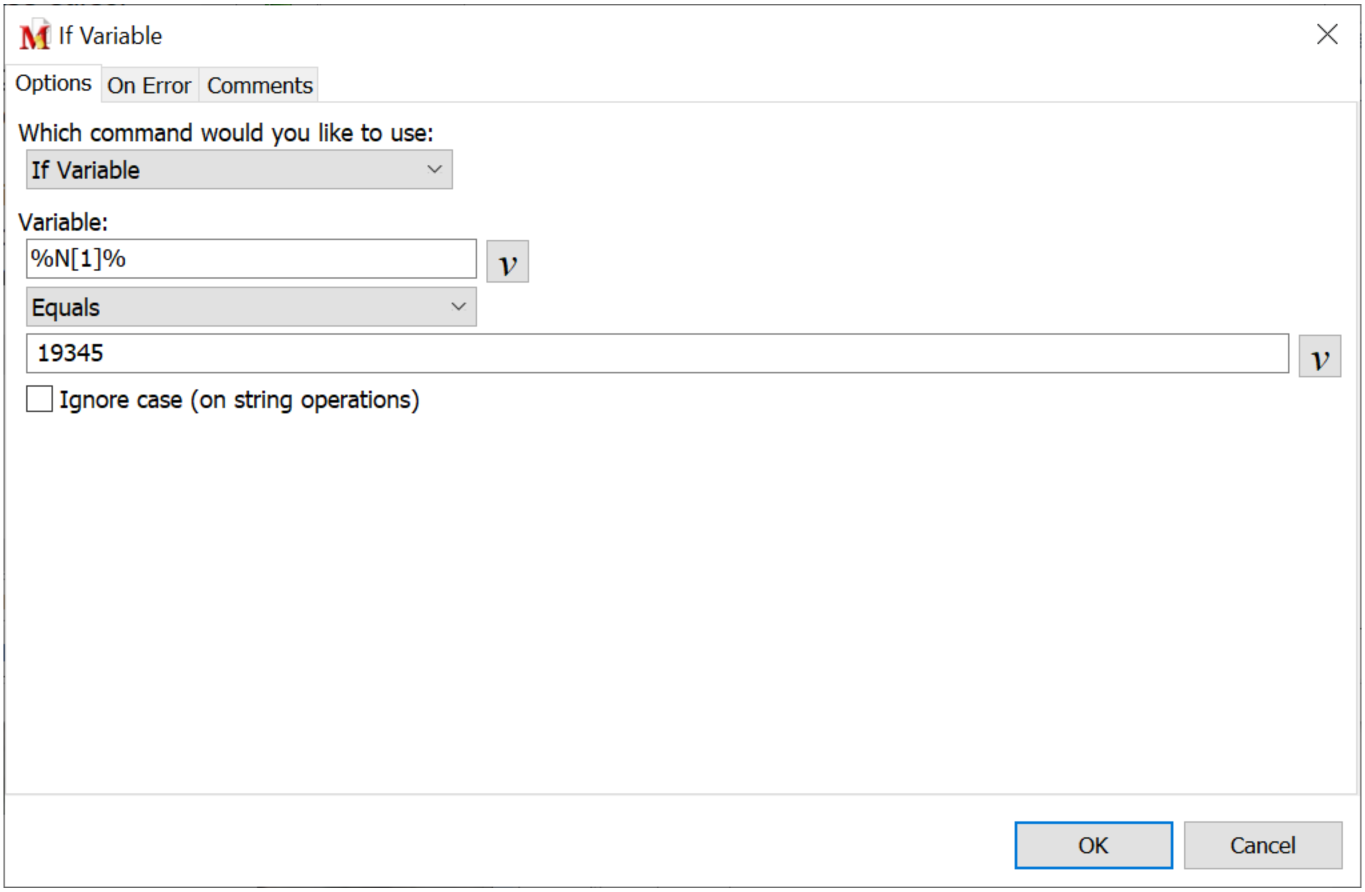
- ✓ **Note:** ALL registry commands (except the registry read and If Registry commands) REQUIRE full Read/Write access to the registry in those parts that the user is wishing to modify. This is only an issue with versions of Windows where security permissions can be set. The system administrator is responsible for these settings as we do not have the ability to modify them.

If Variable

If commands insert logic statements into macros. The If Variable command is one of the most powerful If commands available. Use it to compare a previously saved variable value such as a text string, window title, a number, or anything else for that matter with another value. If the values match (or don't match), the macro can take a specific course of action based on the results.

The If Variable command compares either a text, integer or decimal variable to a fixed value or to another variable. Comparisons may be equal, not equal, less than, greater than, less than or equal to or greater than or equal to.

The "Contains" and "Does not Contain" conditions are available for use with Text variables to test if specific characters are found or not found in the string variable.



Which command would you like to use:
If Variable

Variable:
%N[1]%

Equals

19345

□ Ignore case (on string operations)

Script Editor > Expand Logic Category > If Variable

Variable

Select the variable to compare and insert in this field. The example above shows variable N[1] being compared.

Action

Select the action needed for comparing the variable with another value or variable. Below is a list of the actions (comparisons) available.

- Equals
- Does not Equal
- Is Less Than
- Is Greater Than
- Is Greater Than or Equal To
- Is Less Than or Equal To
- Contains
- Does Not Contain
- Exists
- Does Not Exist

Value to Compare with Variable

Insert the value to compare with the variable selected. Or insert a variable in this field to compare the value of one variable against another.

This field is not accessible when using the Exists or Does Not Exist options as no comparison is made. These two functions verify if the variable selected exists or not.

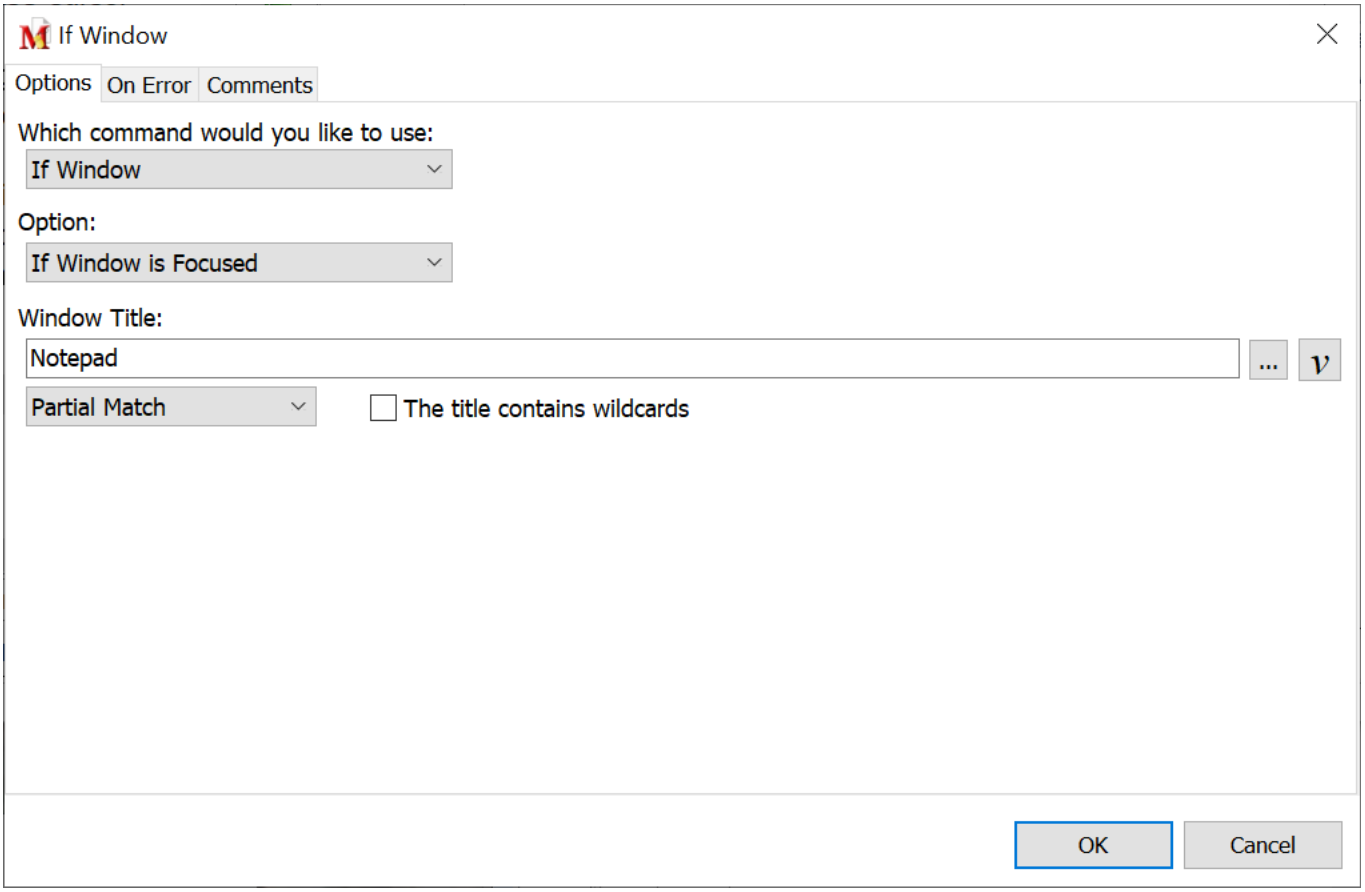
Ignore Case (on String Operations)

When comparing Text String variables, select this option if the comparison does not need to be case sensitive. If selected, the upper or lower case status of the text is ignored.

- ✓ **Note:** If using a Boolean variable in this command, the value to look for should be either True or False. True and False are the only values saved to a Boolean variable.

If Window

There are two If Window options available. One determines if the window is on top (has focus) and the other determines if the window is currently running.



If Window

Which command would you like to use: If Window

Option: If Window is Focused

Window Title: Notepad

Partial Match

□ The title contains wildcards

Script Editor > Expand Logic Category > If Window

Option

Select from the two If Window options described below.

If Window is Focused

Select this option to test if the selected window has focus.

If Window Running (Visible or Hidden)

Select this option to test if the selected window is currently running. The selected window may be a visible or hidden window.

Window Title

Insert the name of the window to test or click on the Browse button to select from the Hidden or Visible windows currently running on the computer.

Partial Match/Exact Match

Select whether the window title must match exactly or if a partial match is sufficient.

Exact Match

The Exact Match requires that the window name entered matches exactly with the name of the window used in the If statement. For example, if inserting Notepad as the window name, but the window currently open when the macro runs is Notepad - Untitled, then the macro would ignore the If statement.

Partial Match

An exact window title match is not required. For example, insert "notepad" or "notep" as the window title and either would be recognized by the If statement. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

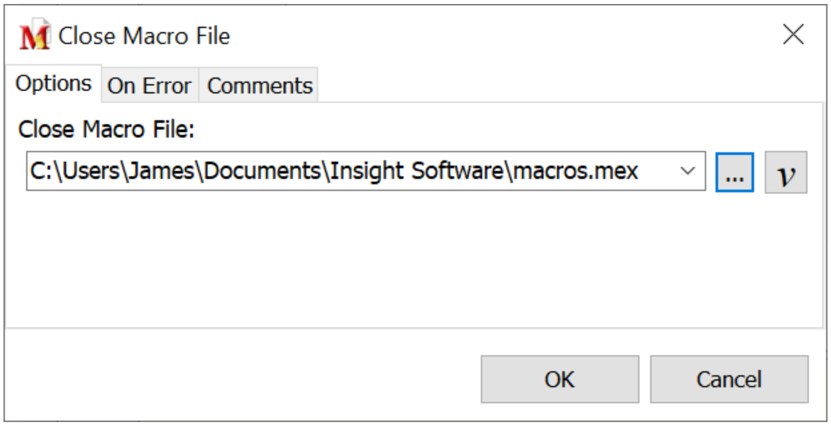
Select this option if including a wildcard in the window title name. A wildcard includes the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Macro Control

Close Macro File

Macro Express Pro allows for more than one macro file to be open at any given time. The Close Macro File command provides a means to close one of the files currently open.

- ✓ **Note:** If the macro file selected is the only file open when the macro runs, an error occurs, which may be handled by the On Error options. One macro file must be open at all times. If the file selected is not currently open when the macro runs, the Close Macro File command is ignored and the macro continues.



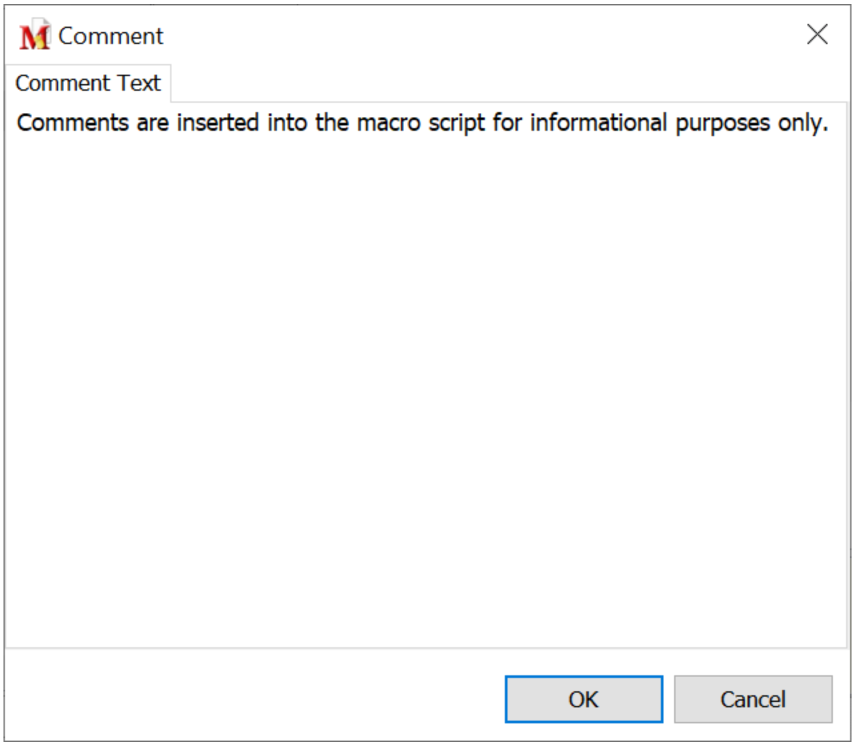
Close Macro File: C:\Users\James\Documents\Insight Software\macros.mex

Script Editor > Expand Macro Control Category > Close Macro File

Select from the drop down list to display all macro files currently loaded into Macro Express. Or click on the Browse button to select another macro file that is not in use.

Comment

When creating a large or complex macro, consider placing comments within the macro script itself. The comments are placed within the macro script in bold blue text. This helps to easily identify what each piece of a macro does, in order to more readily find, edit or change any of the macro commands or components. The comments are reminders only and are not played back as part of the macro.



Comments are inserted into the macro script for informational purposes only.

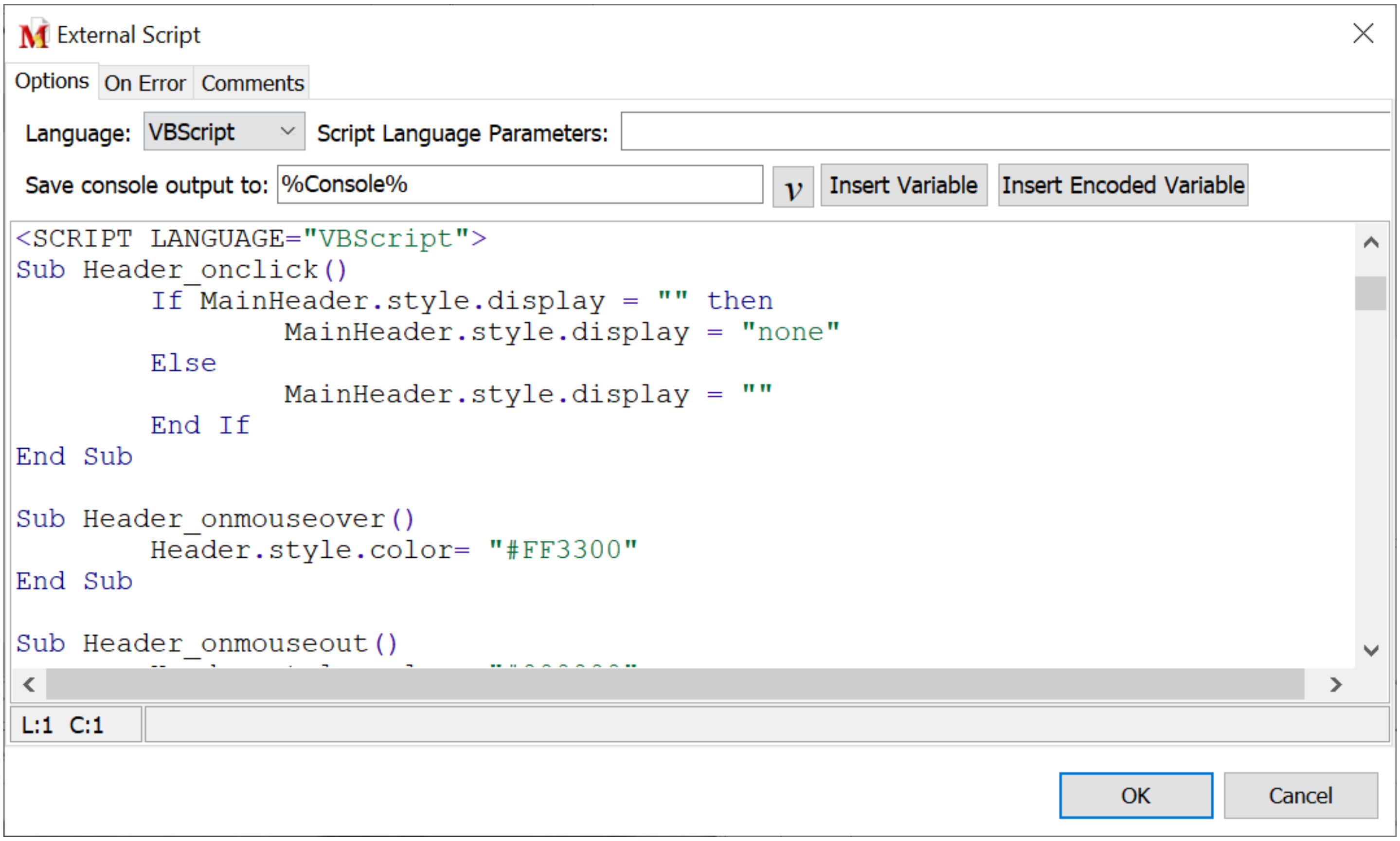
Script Editor > Expand Macro Control Category > Comment

In the edit box, insert the comments and click OK to save. The comments are inserted into the macro script where positioned. The comments are displayed in blue text in the Script Editor to easily distinguish comments from other commands.

Most commands allow for comments to be displayed on the same line as the command text in the Script Editor. Use the Comments Tab in most command windows to access this feature.

External Script

The External Script command provides a powerful addition to Macro Express, providing a new layer of customization to the program. This command allows a macro to run other script languages as a part of the overall macro. Select from JScript, VBScript or HTA/HTML script languages.



Language: VBScript

Script Language Parameters:

Save console output to: %Console%

```
<SCRIPT LANGUAGE="VBScript">
Sub Header_onclick()
        If MainHeader.style.display = "" then
                MainHeader.style.display = "none"
        Else
                MainHeader.style.display = ""
        End If
End Sub

Sub Header_onmouseover()
        Header.style.color= "#FF3300"
End Sub

Sub Header_onmouseout()
```

Script Editor > Expand Macro Control Category > External Script

Language

Select which scripting language to use from the drop down list.

Save console output to:

This function saves the output from the script to a text string variable. After the script runs it will contain output or error messages sent to the console by the script.

Enter the text string variable in the edit field or click on the Variables button to select the variable that will store the output information. This field may not be left empty.

Parameters

Include any parameters needed by the script language. The parameters are based on the script language chosen.

Script

Enter the script for the language selected. Either type the script in the edit box or copy from another source and paste.

Insert Variable

This option utilizes the contents of the variable when the script runs. For example, when the variable %Title% is inserted in the script, the contents of the variable are used when the script runs.

Insert Encoded Variable

There may be times when the name of a variable, such as %FirstName%, is needed by the script instead of the content of that variable. This option places some distinguishing characters in the variable name to instruct the script to not expand the variable content

For example the variable %FirstName% would be displayed in the script as {%}FirstName{%}. When using {%}FirstName{%} it will not replace the content of the variable before running and it will change {%}FirstName{%} to %FirstName%.

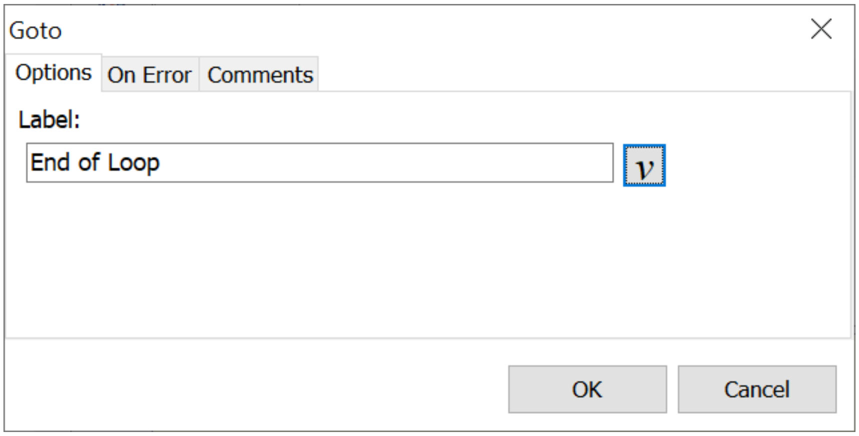
Search Functions

The following search features are available for the External Script edit box:

1. Ctrl + F - Search
2. F3 - Search Next
3. Shift + F3 - Search Previous
4. Ctrl + R - Search and Replace

Go To

The Go To command opens up numerous possibilities when using the conditional If statements. Use this feature to instruct macros to move to specific lines in the macro script without running the commands in between. For instance, if a variable doesn't equal a certain value, then skip the next 15 lines of the script and continue from that point. The macro can either jump ahead in the script or return to a previous line in the script.



Label:
End of Loop

Script Editor > Expand Macro Control Category > Go To

Label

Insert the name of the label previously created using the Label command. The macro goes to the line in the script where the label is located and starts playing back the next command in the script.

- ✓ **Note:** The label is limited to 32 characters in length and does not accept a variable value.

- ❖ **Warning:** It is not advisable to use the Go To command to exit out of a loop or an If statement. Do not exit completely out of a Repeat loop, the ASCII Text File Process command, Text File Process command or an If /End If block using the Go To command. Use the Break command instead.

Below are two macro syntax examples using the Go To and Label commands:

Acceptable

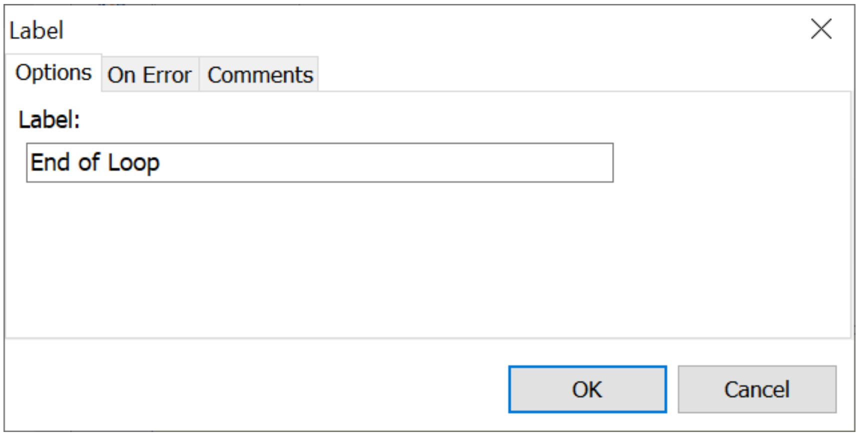
```
  If Variable %Test% Equals "1"
    Goto:ConditionMet
     ...Other macro steps
    :ConditionMet
  End If
```

Not Advisable

```
 If Variable %Test% Equals "1"
    Goto:ConditionMet
    ...Other macro steps
 End If
:ConditionMet
```

Label

The Label command is used in conjunction with the Go to command. It is used as a marker to be placed in the script. The Go to command moves to the line in the script where the Label is located and continues playing the macro from that point forward. The macro can either jump ahead in the script or return to a previous line in the script.



Label:

End of Loop

Script Editor > Expand Macro Control Category > Label

Label

Create a name for the label to be inserted into the script.

- ✓ **Note:** The label is limited to 32 characters in length and does not accept a variable value.

- ❖ **Warning:** It is not advisable to use the Go To and Label commands to exit out of a loop or an If statement. Do not exit completely out of a Repeat loop, the ASCII Text File Process command, Text File Process command or an If /End If block using the Go To command. Use the Break command instead.

Below are two macro syntax examples using the Go To and Label commands:

Acceptable

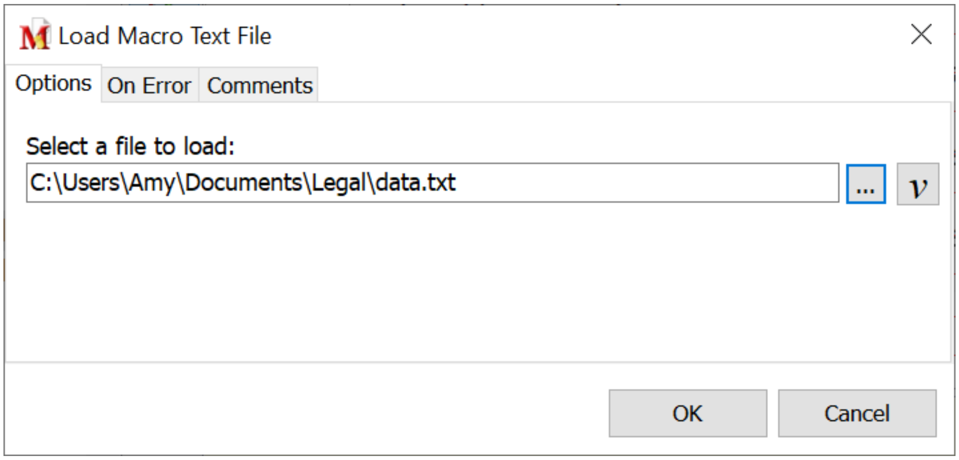
```
  If Variable %Test% Equals "1"
    Goto:ConditionMet
     ...Other macro steps
    :ConditionMet
  End If
```

Not Advisable

```
 If Variable %Test% Equals "1"
    Goto:ConditionMet
    ...Other macro steps
 End If
:ConditionMet
```

Load Macro Text File

Macro Express allows a text file to be loaded during the playback of a macro and execute the text as part of the macro. This is particularly helpful for creating dynamic macros. Change the contents of a text file to fit a specific need before playing the macro.



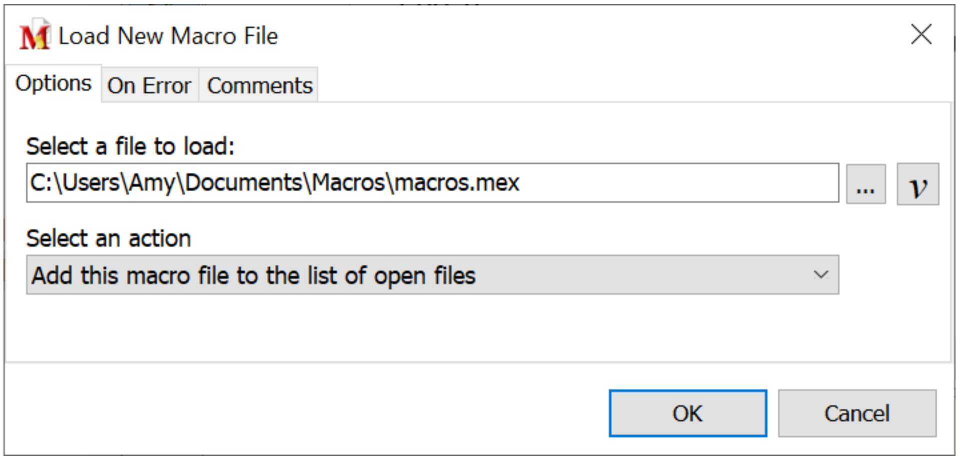
Select a file to load: C:\Users\Amy\Documents\Legal\data.txt

Script Editor > Expand Macro Control Category > Load Macro Text File

Enter a file name and path or click on the Browse button to search for the desired text file. Click the OK button when finished.

Load New Macro File

The Load New Macro File command loads a new macro file. You may choose to add the macro file to the list of open macro files or to close all other macro files before opening the new one.



Select a file to load:
C:\Users\Amy\Documents\Macros\macros.mex

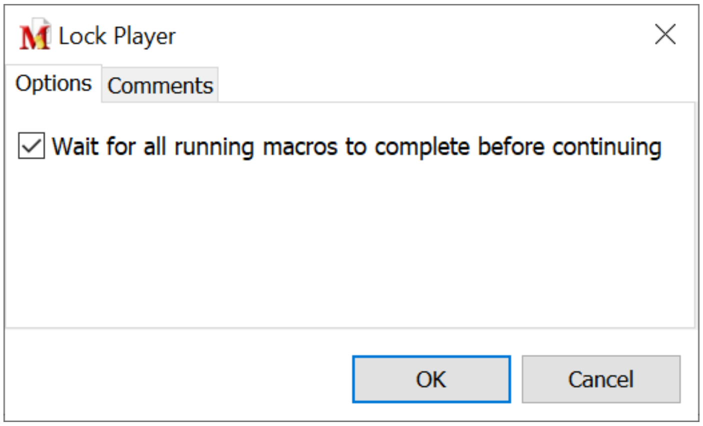
Select an action
Add this macro file to the list of open files

Script Editor > Expand Macro Control Category > Load New Macro File

- ✓ **Note:** When loading a Read Only file, the Default Action selected in the Read Only Prompt preference panel will be used. The Macro Express Player will not notify you during macro playback that you are loading a Read Only file. If the Explorer window is open while the macro runs, you may receive a Read Only prompt, depending on the settings selected in the Preference panel.

Lock / Unlock Player

Since more than one macro may be run simultaneously, this command prevents another macro from starting while the macro containing the Lock Player command runs.



☑ Wait for all running macros to complete before continuing

Script Editor > Expand Macro Control Category > Lock Player

Lock Player

The Lock Player command is typically placed as the first command in the macro script. This instructs the Macro Express Player to not allow another macro to start while the macro is playing. If a macro is already running when a macro containing the Lock Player command is started, it will continue until finished. But no other macros are allowed to start until the macro has finished.

Wait for All Running Macros to Complete before Continuing

With this option selected, the macro does not start until all other running macros have completed.

Unlock Player

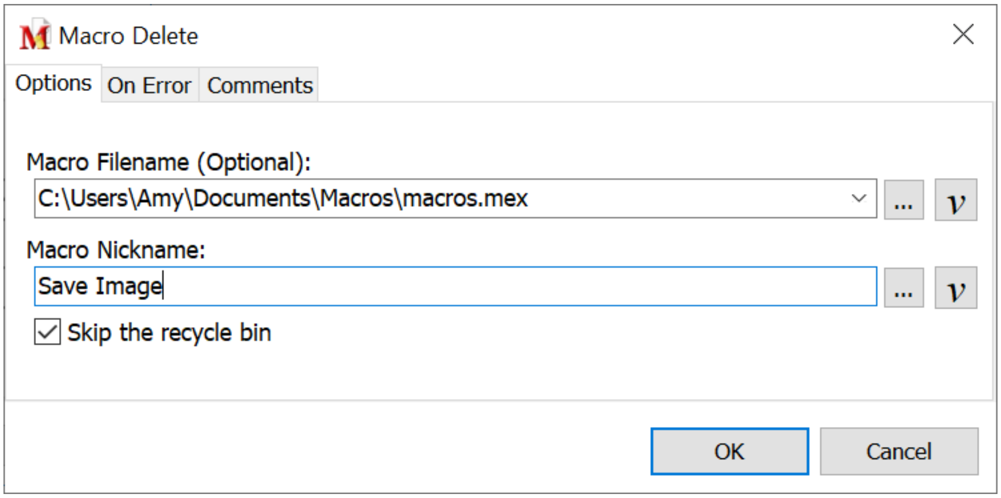
This command is typically placed at the end of the macro script. Unlock Player instructs the Macro Express Player to now allow other macros to run.

When creating a macro, the Unlock Player command is inserted directly to the macro script without an intermediate window opening.

- ✓ **Note:** If a macro is started by selecting the Run Now option, the Lock Player and Unlock Player commands are ignored in the running macro. The macro manually started by Run Now will also run, without waiting for the Unlock Player command.

Macro Delete

The Macro Delete command deletes a macro during playback of an existing macro. A typical use for this function would be to create reminder messages. For example, when creating a one time reminder message - to run an errand, make a call at a specific time, etc. - place the Macro Delete command as the last line of the script. After the message appears on screen during scheduled playback, the macro is deleted - saving the trouble of remembering to delete it later.



Macro Filename (Optional): C:\Users\Amy\Documents\Macros\macros.mex

Macro Nickname: Save Image

☑ Skip the recycle bin

Script Editor > Expand Macro Control Category > Macro Delete

Macro File Name

Since Macro Express allows more than a single macro file to be open at one time, click on the Browse button and locate the file that contains the macro to delete. Or manually type in the file and path.

Macro Nickname

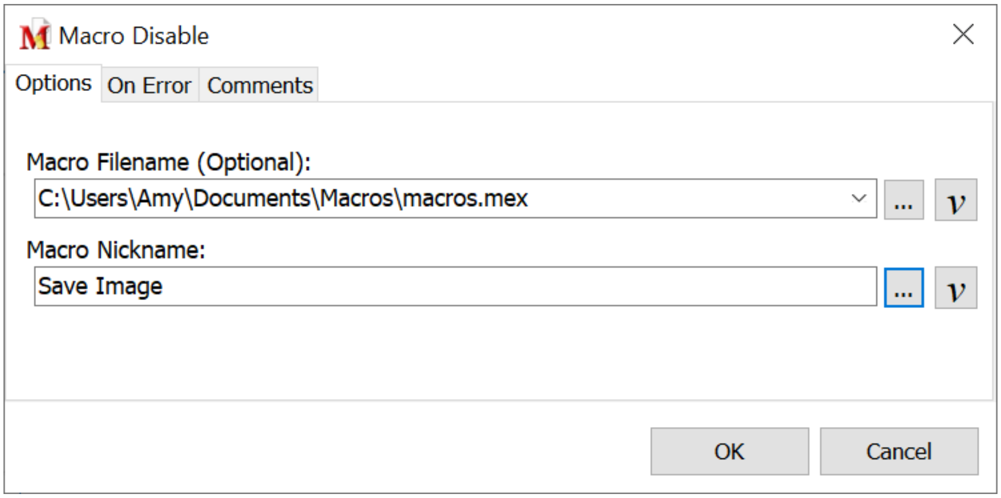
Click on the Browse button to select the macro to delete. Macro nicknames are displayed for all macro files that are currently open. If the macro file that includes the macro to be deleted is not currently open, then type the macro name in the edit box.

Skip Recycle Bin

Select this option to have the deleted macro bypass the Macro Express Recycle Bin.

Macro Disable

Use this function to disable another macro during macro playback.



Macro Filename (Optional): C:\Users\Amy\Documents\Macros\macros.mex

Macro Nickname: Save Image

Script Editor > Expand Macro Control Category > Macro Disable

Macro File Name

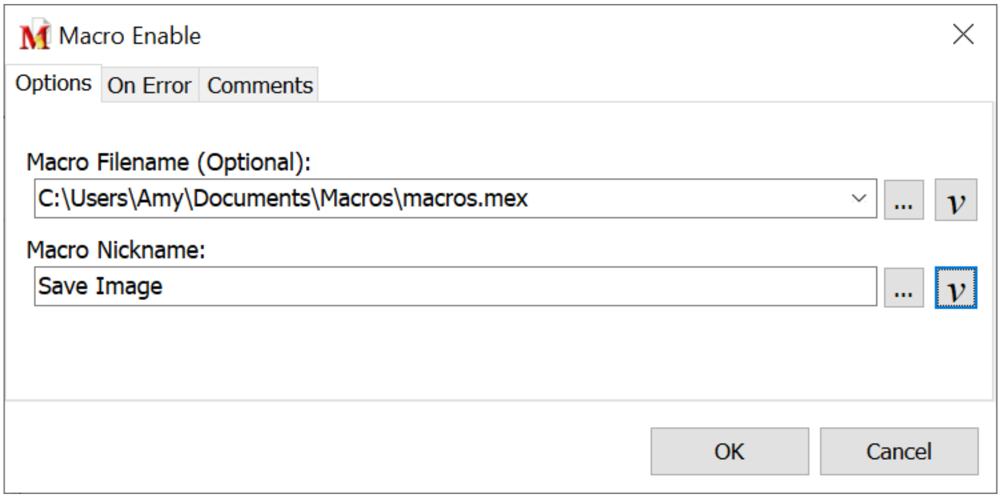
Since Macro Express allows more than a single macro file to be open at one time, click on the Browse button and locate the file that contains the macro to be disabled. Or manually type in the file and path.

Macro Nickname

Click on the Browse button to select the macro to be disabled. Macro nicknames are displayed for any macro files that are currently open. If the macro file containing the macro to be disabled is not currently open, then type the macro name in the edit box.

Macro Enable

This function instructs the macro to enable another macro that has previously been disabled (not currently available for use).



Macro Filename (Optional): C:\Users\Amy\Documents\Macros\macros.mex

Macro Nickname: Save Image

Script Editor > Expand Macro Control Category > Macro Enable

Macro File Name

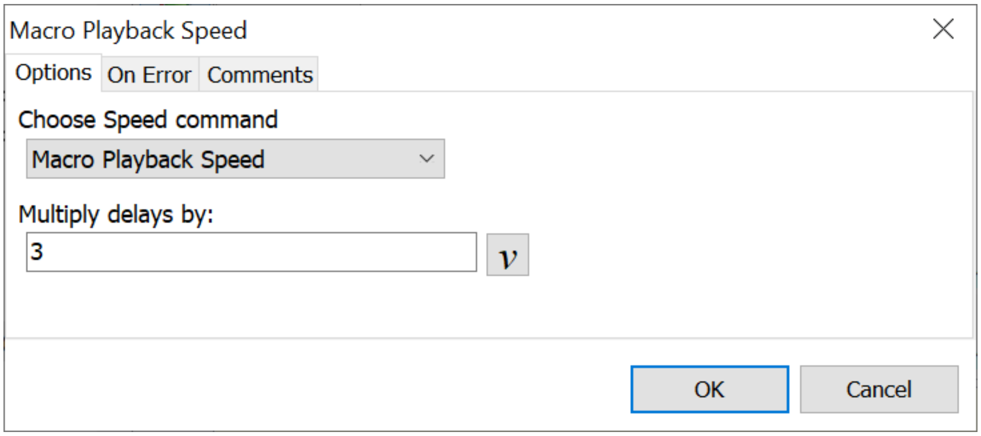
Since Macro Express allows more than a single macro file to be open at one time, click on the Browse button and locate the file that contains the macro to be enabled. Or manually type in the file and path.

Macro Nickname

Click on the Browse button to select the macro to be enabled. Macro nicknames are displayed for any macro files that are currently open. If the macro file containing the macro to be enabled is not currently open, then type the macro name in the edit box.

Macro Playback Speed

Macro Playback Speed adjusts how fast the macro runs. It is typically used to adjust the speed of a captured (recorded) macro. With timing delays recorded between each keystroke and mouse movement, use this option to speed up or slow down the macro.



Choose Speed command: Macro Playback Speed

Multiply delays by: 3

Script Editor > Expand Macro Control Category > Macro Playback Speed

Multiply Delays By

To slow down the macro playback speed, place a number greater than one in the "Multiply Delays By" field. For example, placing a "4" in this field multiplies the delays by four, causing the macro to play back 4 times slower.

To speed up the macro, place a value less than one in the "Multiply Delays By" field. For example, placing 0.5 in this field cuts the delay in half, speeding up the macro by a factor of 2.

Macro Return/Stop

Neither the Macro Return or Macro Stop command opens a dialog window for editing when used in the Script or Direct Editor. Open the Script Editor and expand the Macro Control category of commands. Double click on the Macro Return or Macro Stop command to place in the script.

Macro Return

This command is intended for use with the Macro Run command to return to the top level macro. Macro Return stops the execution of the called macro. Use Macro Return in the middle of a logic statement to return to the main macro before the end of the called macro. It is not necessary to put a Macro Return at the end of the macro.

Here is an example of how this works. Macro A is the top level macro and uses the Macro Run command to execute Macro B. Macro B includes a conditional statement looking for a particular value. If the value is not found, the Macro Command stops Macro B but continues with Macro A.

In other words, inserting a Macro Return command in Macro B stops the playback of Macro B and returns to Macro A where it continues running the remainder of Macro A's script.

Macro Stop

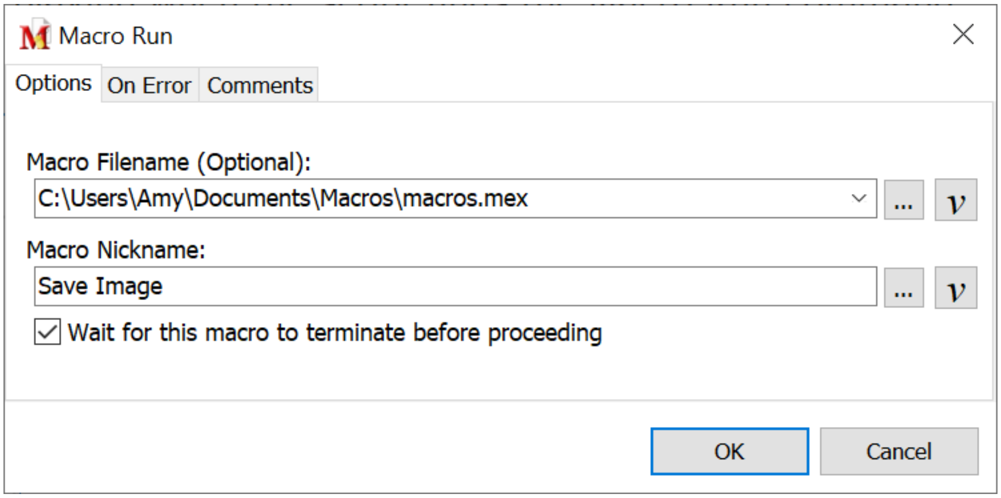
The Macro Stop command stops the execution of a macro. Use this primarily in conjunction with a conditional statement. If a desired condition is not met, terminate the macro.

In the Macro Return example above, if Macro B executes a Macro Stop, both Macro A and Macro B are terminated immediately.

Note: It is not necessary to place a Macro Stop command as the last line of a macro. The macro stops automatically after the last line of the macro is executed.

Macro Run

This function is the key to stringing macros together. Incorporate one or several macros into another macro by inserting this command into the macro script. The called macro starts playing when the script finds the Macro Run command.



Macro Filename (Optional): C:\Users\Amy\Documents\Macros\macros.mex

Macro Nickname: Save Image

☑ Wait for this macro to terminate before proceeding

Script Editor > Expand Macro Control Category > Macro Run

Macro File Name

Since Macro Express allows more than a single macro file to be open at one time, click on the Browse button and locate the file that contains the macro to be run. Or manually type in the file and path.

- ✓ **Notes:**
  1: If the Macro Filename field is left blank, Macro Express looks for the macro in the same file as the macro currently being written.
  2: The Macro Run command runs a macro regardless of whether or not it is disabled.
  3: If two macros have the same name the one with the earliest index number will run.

Macro Nickname

Click on the Browse button to select the macro to run. Macro nicknames are displayed for any macro files that are currently open. If the macro file containing the macro to be run is not currently open, then type the macro name in the edit box.

Wait for this Macro to Terminate before Proceeding

Check this box to make sure that the first macro does not continue running until the macro in the Macro Run command has finished playing back. If not selected, the original macro and the called macro will both play back simultaneously.

- ✓ **Note:** If the called macro is a Popup Menu macro, the wait option is ignored. The Popup menu will open and the original macro will continue running.

Hint

After inserting the Macro Run command into the script, right mouse click on the command. From the context menu, left click on Edit Macro to open a new tab in the Script Editor. The new tab displays the contents of the macro called in the Macro Run command. This provides a quick way to view the contents of the called macro.

Additional Information About Variables

Global variable values are passed from one macro to another when the macros are strung together using the Macro Run command.

For example if variable N[1] was assigned a value of "123" in the main macro and the called macro types out the value of N[1] to another application using the Text Type command, "123" is typed. There is no need to set N[1] to a value of "123" again in the called macro.

The variable values are cleared after the entire macro is run. To save the variable values for use with another macro, which is not part of the original macro or called macros, use the Variable Save and Variable Restore commands.

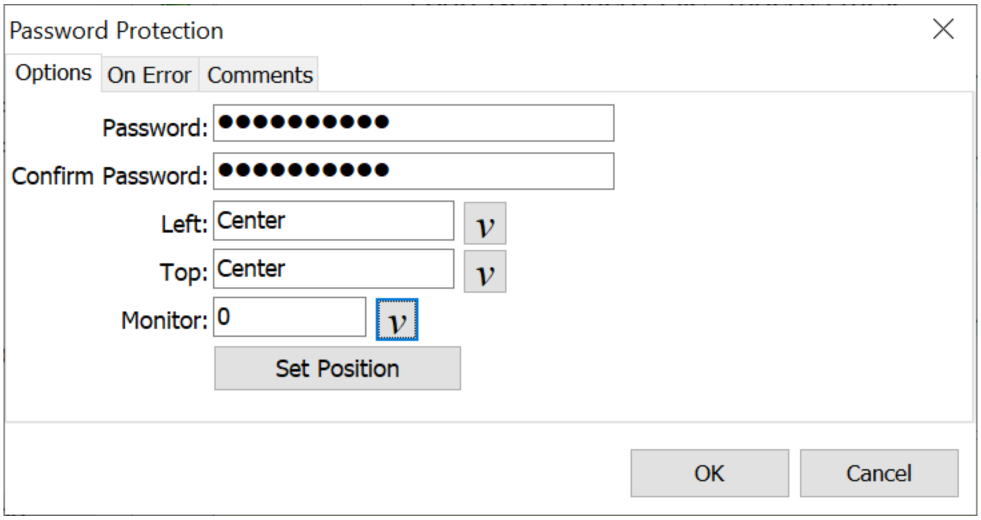
Similarly, other functions such as the Macro Playback Speed, Keystroke Speed, etc. are passed from the original macro to the called macro. If the variable value is reset in the called macro or the speed is changed in the called macro, those values are passed back to the original macro when it resumes.

- ✓ **Note:** The above statements are valid **IF** the Wait for this Macro to Terminate before Proceeding option, described above, **is** selected. If the option is **NOT** selected, then the values are not passed from one macro to the other.

Password Protection

The password protection feature requires that a password be entered before macro playback continues. The Encrypted Text command provides additional security for sensitive information.

- ✓ **Note:** The Password Protection command is somewhat redundant with the password options found in the Security tab of the Script Editor. Use the Security tab to require a user to enter a password before running the macro, before viewing the macro for editing or both. The difference being that the Password Protection command may be inserted anywhere in the macro script if requiring a user to insert a password at a particular point during macro playback.



Password: ●●●●●●●●●●

Confirm Password: ●●●●●●●●●●

Left: Center

Top: Center

Monitor: 0

Script Editor > Expand Macro Control Category > Password Protection

Insert the password protection command as the first item of the macro script to password protect the entire macro. If the password protection command is the first item of the macro, then entering the password is required before the macro starts. Or use the password option found in the Security tab of the Script Editor to accomplish the same thing.

The password protect command may also be inserted anywhere in the macro script. In this instance, the first part of the macro plays until it reaches the password protection command. A dialog opens prompting for the password. Insert the password, click OK and the remainder of the macro script plays.

Entering the Password

Input the password in the Password edit box. This field accepts a combination of letters, numbers and symbols found on the keyboard. Type the password again in the Confirm Password field to verify that the input was correct. If typed differently, a message displays indicating that the passwords do not match and to try again.

Positioning the Password Window

Define the position on the screen where the input box should appear during the playback of the macro. Insert exact coordinates or use the Set Position Manually button to set where the window should appear. See the Window Positioning topic for additional information.

Macro Express Commands

These commands all relate to Macro Express functionality. Only the Restart Macro Express command requires an edit window. For the other commands listed, open the Macro Express category of commands and select the command to insert into the script.

Empty Macro Recycle Bin

Macro Express has its own Recycle Bin. Any macros deleted from the Macro Explorer are sent to the Macro Express Recycle Bin. This macro command empties the Macro Express Recycle Bin during the playback of a macro. At this point the macros are permanently deleted from the program and cannot be recovered.

Reload Macro Express Preferences

This command reloads the Macro Express Preference settings from the Windows Registry.

Reset Hooks

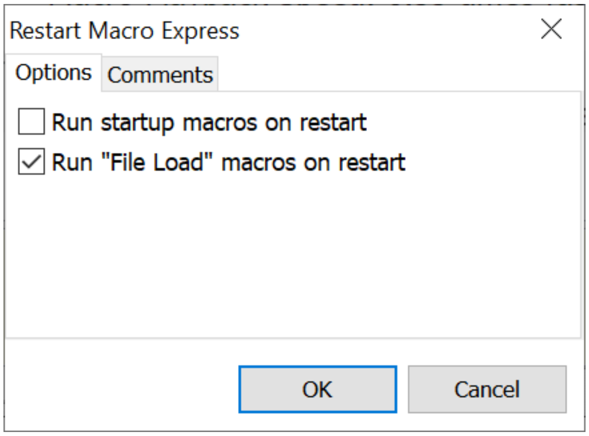
It is possible for other programs started after Macro Express to improperly install themselves into the system wide Windows keyboard and mouse hooks. This causes Macro Express to no longer be able to monitor keystrokes or mouse movements and clicks.

This command reinstalls the Windows keyboard and mouse hooks used by Macro Express while the macro is playing so that Macro Express will have the first look at the keystrokes and mouse movements/clicks and be able to monitor them again.

Restart Macro Express

This command closes Macro Express and restarts the program.

- ✓ **Note:** The Restart Macro Express command should be the last command in the script. Any commands in the script after the Restart Macro Express command are ignored and will not play back.



☐ Run startup macros on restart

☑ Run "File Load" macros on restart

Script Editor > Expand Macro Express Category > Restart Macro Express

Run startup macros on restart

If any Scheduled Macros are scheduled to play back on Macro Express startup, choose whether or not to have those macros run again after restarting Macro Express. Select this option to run the scheduled macros again after a program restart.

Run "File Load" macros on restart

If any macros are scheduled to start when a macro file loads, a restart of Macro Express causes these macros to run again. Choose whether or not to have the macros run again on startup of Macro Express. Select this option to run the File Load scheduled macro to run after a program restart.

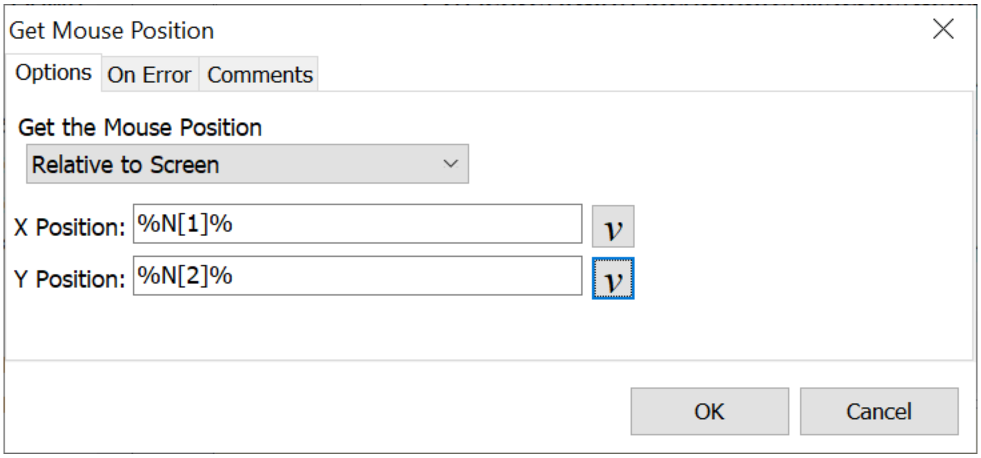
Terminate Macro Express

The Terminate Macro Express command shuts down the Macro Express Player (which also shuts down the editor if it is running) from within a macro. This is obviously only to be used as the last command in a macro as no more commands will execute after this one.

Mouse

Get Mouse Position

The Get Mouse Position command locates the current position of the mouse cursor on the screen or window and saves the X and Y coordinates to Integer variables. This is an easy way to return the mouse cursor to its original position after a macro has completed.



Get the Mouse Position: Relative to Screen

X Position: %N[1]%

Y Position: %N[2]%

Script Editor > Expand Mouse Category > Get Mouse Position

Select whether to save the mouse position relative to the screen or to the window that has focus. Then insert the variable names for saving the X and Y coordinates.

Sample Macro

This macro captures the initial position of the mouse cursor and saves the coordinates to variables N[1] and N[2]. The macro then moves the mouse cursor to the screen coordinates of 300 and 300. The last command instructs the mouse to move to coordinates N[1] and N[2], returning the cursor to its original position.

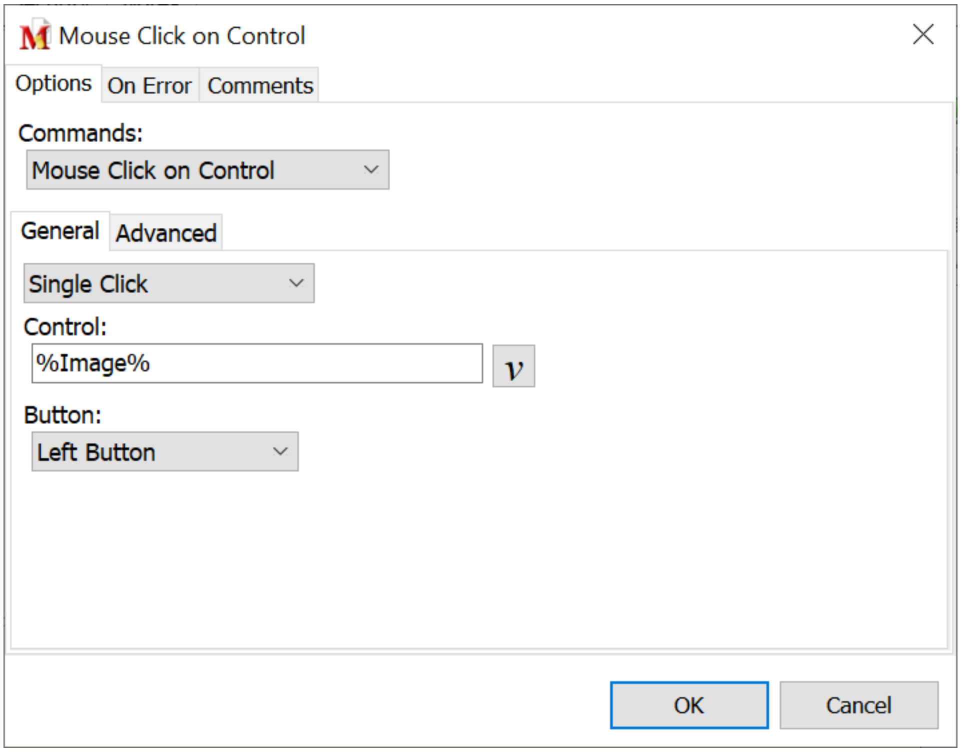
The delays are added to make the mouse movements visible. Without delays the movements are too quick to notice what is happening.

Macro Script

```
Get Mouse Position Screen: %N[1]%, %N[2]%
Delay 2 Seconds
Mouse Move Screen 300,300
Delay 2 Seconds
Mouse Move Screen %N[1]%, %N[2]%
```

Mouse Click on Control

This command allows the user to send a click to the specified Windows control. This is most useful with buttons because it will make the button think that the user clicked on it, even if the mouse is not on the control.



Commands: Mouse Click on Control

Single Click

Control: %Image%

Button: Left Button

Script Editor > Expand Mouse Category > Get Mouse Position

General Tab

There are three steps to setting up this command correctly.

1. Select whether to have the mouse single or double click on the control.
2. Choose the Control variable previously defined.
3. Decide whether to click with the Left, Right or Middle mouse button.

Advanced Tab

This tab determines the area of the control to click. By default, this command sends the click to the center of the control. Or select the Coordinates option and manually insert the X and Y coordinates. The coordinates are relative to the control.

The coordinates 0, 0 refer to the top left corner of the control. Or change the coordinates to move a number of pixels to the right and down from the top.

Mouse Clicks

The following mouse commands do not display a Macro Express window when editing and are inserted directly into the macro script. Open the Mouse category of commands and double click on one of the options listed below to insert into the script. This list also represents how the commands are displayed in the Script Editor.

- Mouse Left Button Down
- Mouse Left Button Up
- Mouse Left Click
- Mouse Left Double Click
- Mouse Middle Button Down
- Mouse Middle Button Up
- Mouse Middle Click
- Mouse Middle Double Click
- Mouse Right Button Down
- Mouse Right Button Up
- Mouse Right Click
- Mouse Right Double Click

These commands execute mouse clicks during the playback of the macro. First, use the Mouse Move command or another command that positions the mouse before inserting a mouse click command.

Left Mouse Button Options

- Mouse Left Button Down - Holds down the left mouse button without releasing it. This can be used for dragging items during macro playback.
- Mouse Left Button Up - Releases the left mouse button from its held down position.
- Mouse Left Click - Clicks the left mouse button one time.
- Mouse Left Double Click - Double clicks the left mouse button.

Middle Mouse Button Options

- Mouse Middle Button Down - Holds down the middle button of a three button mouse without releasing it.
- Mouse Middle Button Up - Releases the middle button of a three button mouse from its held down position.
- Mouse Middle Click - Clicks the middle button of a three button mouse one time.
- Mouse Middle Double Click - Double clicks the middle button of a three button mouse one time.

Right Mouse Button Options

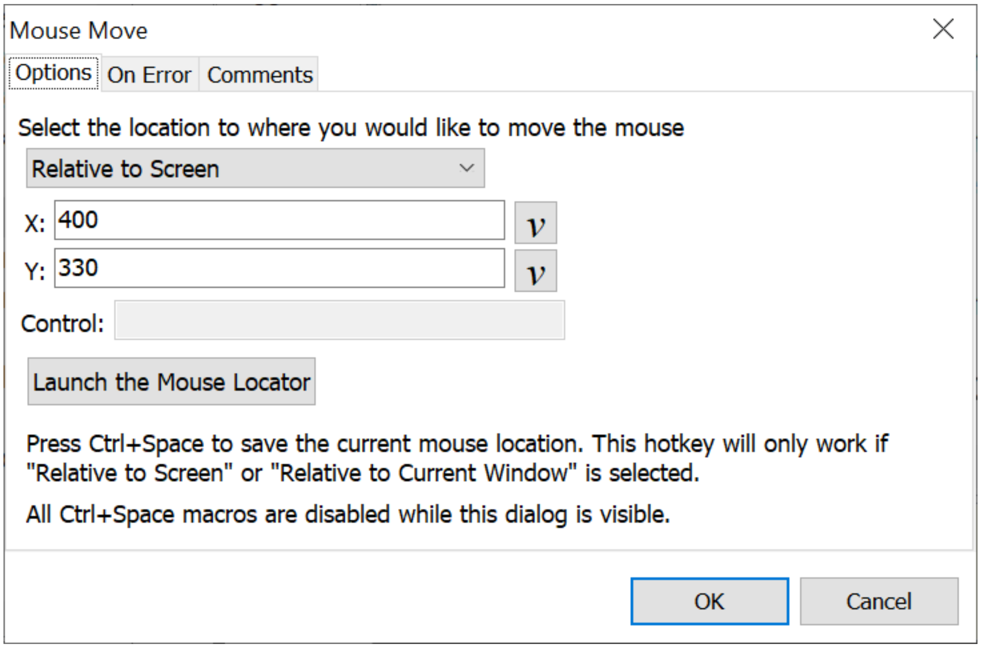
- Mouse Right Button Down - Holds down the right mouse button without releasing it.
- Mouse Right Button Up - Releases the right mouse button from its held down position.
- Mouse Right Click - Clicks the right mouse button one time.
- Mouse Right Double Click - Double clicks the right mouse button one time.

✓ **Note:** If using one of the Mouse Button Down commands, make sure to include a corresponding Mouse Button Up command in the macro. Otherwise the mouse may become stuck in the down position.

Mouse Move

The most common use of the mouse in macros is to move the cursor to a particular spot and click on a button or menu command. The first step in this process is to click on the Mouse Move command.

For the mouse functions to be accurate, use the Mouse Locator to determine the exact coordinates of the button or menu command on the window or screen.



Select the location to where you would like to move the mouse

Relative to Screen

X: 400

Y: 330

Control:

Press Ctrl+Space to save the current mouse location. This hotkey will only work if "Relative to Screen" or "Relative to Current Window" is selected.

All Ctrl+Space macros are disabled while this dialog is visible.

Script Editor > Expand Mouse Category > Mouse Move

Select the location to where you would like to move the mouse

First select from the drop down menu the frame of reference for the mouse movement. The five options are discussed in detail below.

To the Text Cursor Position

This option instructs the macro to move the mouse to the location of the text cursor. The text cursor is the flashing bar that appears on screens where text input is to be made. This can be particularly useful for applications that do not support the SHIFT and Arrow Keys for highlighting text for clipboard copying.

To copy text to the clipboard with the mouse, first use the keyboard to tab to the desired field and the arrow keys to position the cursor. Use the Text Cursor Position function to move the mouse to the same spot as the text cursor. A Mouse Left Click Down command and a Mouse Move Relative to Position command followed by a Mouse Left Click Up highlights text for a clipboard capture.

✓ **Note:** This function does not work on programs that create and use their own text cursor. Programs with their own text cursor do not use the built in Windows Text Cursor functions. Ironically, MS Word appears to fall into this category.

Relative to Screen

This option instructs the macro to move the mouse cursor relative to its position on the computer screen. Use the "Screen Coordinates" numbers from the Mouse Locator with this option.

Relative to Window

This option instructs the macro to move the mouse cursor relative to its position within the window that is active. Use the "Active Window" numbers from the Mouse Locator with this option.

Relative to Last Position

This option instructs the macro to move the mouse cursor relative to the last position of the cursor. In this case do not input new coordinates, just the number of pixels for the mouse to move.

For example assume that the mouse position relative to the screen is X = 100, Y = 200. This means the cursor is 100 pixels from the left edge of the screen and 200 pixels from the top edge of the screen. If we were to plug in X = 20, Y = 50, then the mouse would move to an X coordinate of 120 and a Y coordinate of 250. Or, if we plugged in X = -20, and Y = -50 then the mouse would move to an X coordinate of 80 and a Y coordinate of 150 based on our original mouse position.

Use the Mouse Locator and some simple math to determine how many pixels are needed to move. Start with the first position and jot down the X and Y coordinates. Move the cursor to the new location and subtract the difference. If the starting position is X = 400 and Y= 450 and the new position is X = 500 and Y = 500, then plug in the values of X = 100 and Y = 50.

Select coordinates by either entering the coordinates directly or assigning the coordinate value to a variable. An example of using variables might be to save the original mouse location into variables and then move the mouse back to the original location when finished processing the macro.

Relative to Control

This option assumes using a previously defined Window Control saved to a variable. Enter the name of the control variable or select it from the variable list and then select the coordinates. The coordinates 0, 0 refer to the top left corner of the control. Change the coordinates to move a number of pixels to the right and down from the top.

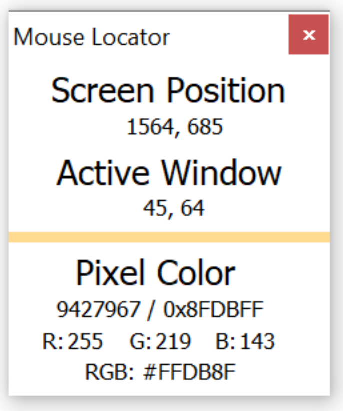
Assume that 10 is assigned as the X coordinate and 15 is the Y coordinate. In this case the mouse cursor would point10 pixels to the right of the left edge of the control and 15 pixels down from the top.

- ✓ **Note 1:** When using the Relative to Screen or Relative to Window options, open the Mouse Locator and position the mouse to the desired location. Then press the Ctrl Key and Space Bar to insert the coordinates into the X and Y fields.

- ✓ **Note 2:** With each of the above selections, except for the Text Cursor Position, optionally use Integer variables for the coordinate values.

Launch Mouse Locator

Clicking on the Launch Mouse Locator button activates the Mouse Locator utility. If this window pops up in an area of the screen that covers important data, click on the window, hold the left mouse button down on the window title bar and drag to another part of the screen.



Mouse Locator ×

Screen Position
1564, 685

Active Window
45, 64

Pixel Color
9427967 / 0x8FDBFF
R: 255 G: 219 B: 143
RGB: #FFDB8F

Screen Position

The set of numbers directly under Screen Position lists the position of the mouse relative to the computer screen. As the mouse is moved, the numbers on the Mouse Locator window change in relation to the mouse cursor position. The number to the left is the X coordinate, or how many pixels from the left edge of the screen. The number on the right is the Y coordinate, or how many pixels the mouse is positioned below the top edge of the screen.

Active Window

The set of numbers directly under Active Window lists the position of the mouse relative to the active window or the window that has the focus. The Mouse Locator window takes focus when it is first launched. Give focus back to the window and the Mouse Locator window remains on top in order to read the coordinates of the window.

As the mouse moves, notice the numbers changing in relation to the mouse position. The number to the left is the X coordinate, or how many pixels the mouse is positioned from the left edge of the active window. The number on the right is the Y coordinate, or how many pixels the mouse is positioned from the top of the active window.

Pixel Color

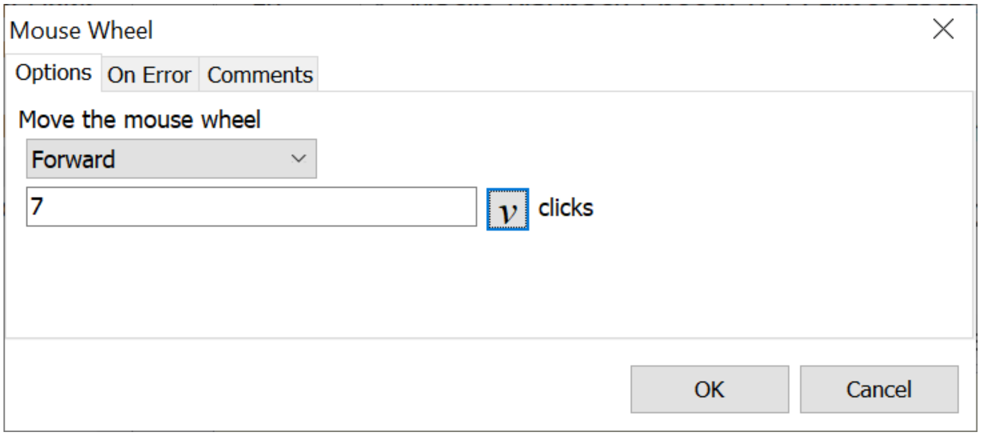
The color of the pixel underneath the mouse pointer is displayed on the Mouse Locator window.

Windows assigns a numeric value to each color, normally from 6 to 8 digits in length. The left side is the decimal value. The right side displays the hexadecimal value in BGR format. The bottom row displays the pixel color in separate values for Red, Green and Blue. The pixel color is used with the Get Pixel Color command and is not needed for the Mouse Move command.

Click the X in the upper right corner to close the Mouse Locator window.

Mouse Wheel

This command moves the mouse wheel forward or backward the number of clicks assigned.



Move the mouse wheel: Forward

7 clicks

Script Editor > Expand Mouse Category > Mouse Wheel

Move the Mouse Wheel

Forward

Moves the wheel forward the number of clicks specified when using a mouse with a wheel.

Backward

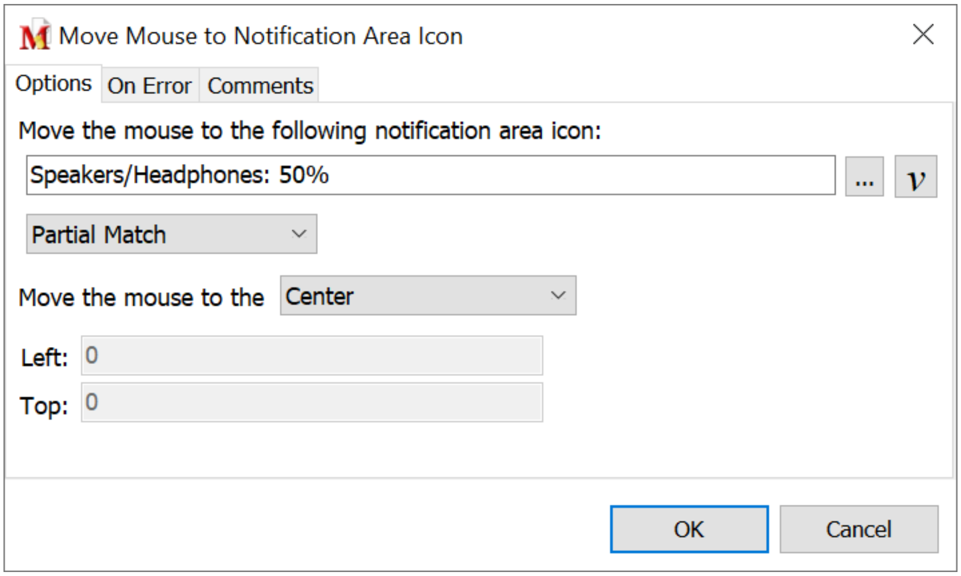
Moves the wheel backward the number of clicks specified when using a mouse with a wheel.

Clicks

Insert the number of clicks that the wheel should move backward or forward.

Move Mouse to Notification Area Icon

This command selects a specific notification area icon and moves the mouse cursor over that icon. Many applications that place icons in the notification area need to be activated by clicking on the icon. This command moves the mouse cursor to the icon. Then follow with a Left or Right Mouse Click to open the application window.



Move the mouse to the following notification area icon:

Speakers/Headphones: 50%

Partial Match

Move the mouse to the Center

Left: 0

Top: 0

Script Editor > Expand Mouse Category > Move Mouse to Tray Icon

Move the mouse to the following notification area icon

Press the Browse button and select the desired notification area icon. The names of each displayed icon are shown in the drop down list. Or type in the name of the icon into the field.

Exact Match / Partial Match

This option is for use when manually inserting the icon name in the edit field. When hovering the mouse cursor over an icon, a yellow text box appears displaying the icon name to help know what name to enter. Decide whether to match the name exactly or just use a partial text match.

Move the Mouse to

The mouse cursor points to the position on the icon selected. This command defaults to the center of the icon. The other option is to select specific coordinates. The coordinates 0, 0 are in the top left corner of the icon. Change the coordinates to move a number of pixels to the right and down from the top.

- ✓ **Note 1:** Windows has options to collapse the notification area. The Move Mouse to Notification Area

Icon command will not work if the icon is hidden behind the collapsed tray.

- ✓ **Note 2:** When moving to the tray icon for a Macro Express Popup Menu macro, use the Partial Match method. Exact Match does not work because the name associated with the Popup Menu notification area icon is changed when a macro is running.

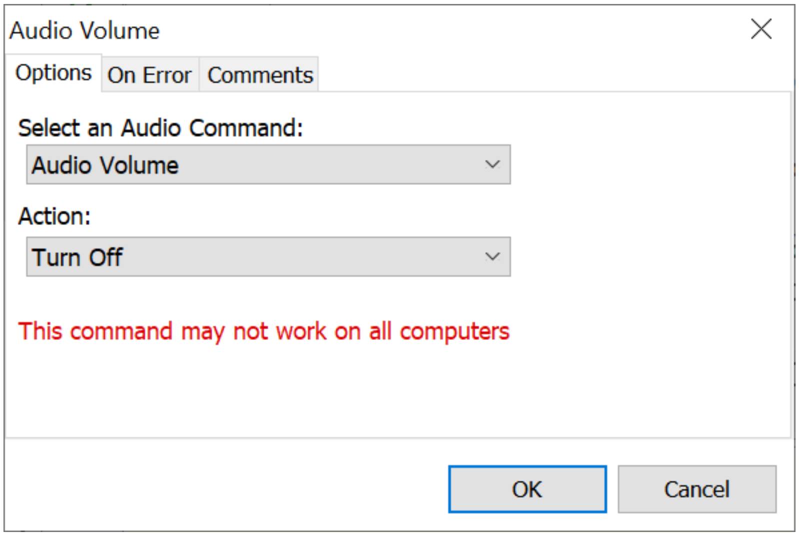
Multimedia

Audio Controls

These audio functions adjust the volume on the computer's sound system. Not all functions may be available on a particular computer. Speaker support is dependent upon the audio device in the computer.

Some Audio commands may not be supported on certain computers or versions of Windows. If your computer does not support a specific command, the message 'This command is not supported on your computer.' will be displayed.

These commands may still be added to a macro, but they would only work on another computer that supports these capabilities.



Select an Audio Command: Audio Volume

Action: Turn Off

This command may not work on all computers

Script Editor > Expand Multimedia Category > Audio Controls

Audio

The audio controls apply to the computer's speakers. The treble and bass are audio controls. Issue macro commands to turn the volume off, set it on its highest setting, nudge it up, nudge it down, toggle it between muted and not, nudge the balance to the left or right, center the audio balance, nudge the treble up or down or nudge the bass up or down.

For some of the commands that affect the volume, a message of 'This command only affects Macro Express's volume.' may appear on certain computers. When Macro Express is running, there is an entry in the Volume Mixer for Macro Express. This is the only volume option that will be affected when the warning message is displayed.

MIDI

The MIDI options control the volume for any MIDI instruments installed on the computer. Issue macro commands to turn the volume off, set it on its highest setting, nudge it up, nudge it down, toggle it between muted and not, nudge the balance to the left or right or center the balance.

- ✓ **Note:** For most computers, the audio range, from lowest to highest, is comprised of 20 increments. Each volume command inserted changes the status by approximately 5%.

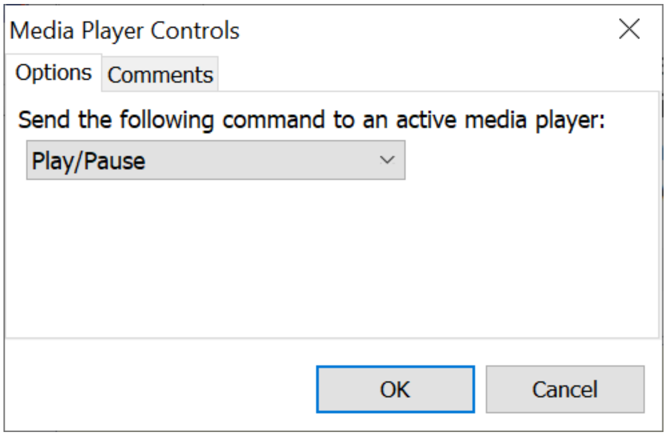
Beep

This command generates a "Beep" sound during the playback of a macro. Use it to call attention to a special action that needs to be taken or to simply alert that the macro has completed playback.

There is not a Macro Express window that opens when using this command. Just expand the Multimedia category of commands in the Editor, double click on the Beep command and insert it into the script. It is displayed as Beep in the Script Editor and as <BEEP/> in the Direct Editor.

Media Player Controls

The Media Player Controls send commands to the media player currently running, such as Windows Media Player or Real Player.



Send the following command to an active media player:
Play/Pause

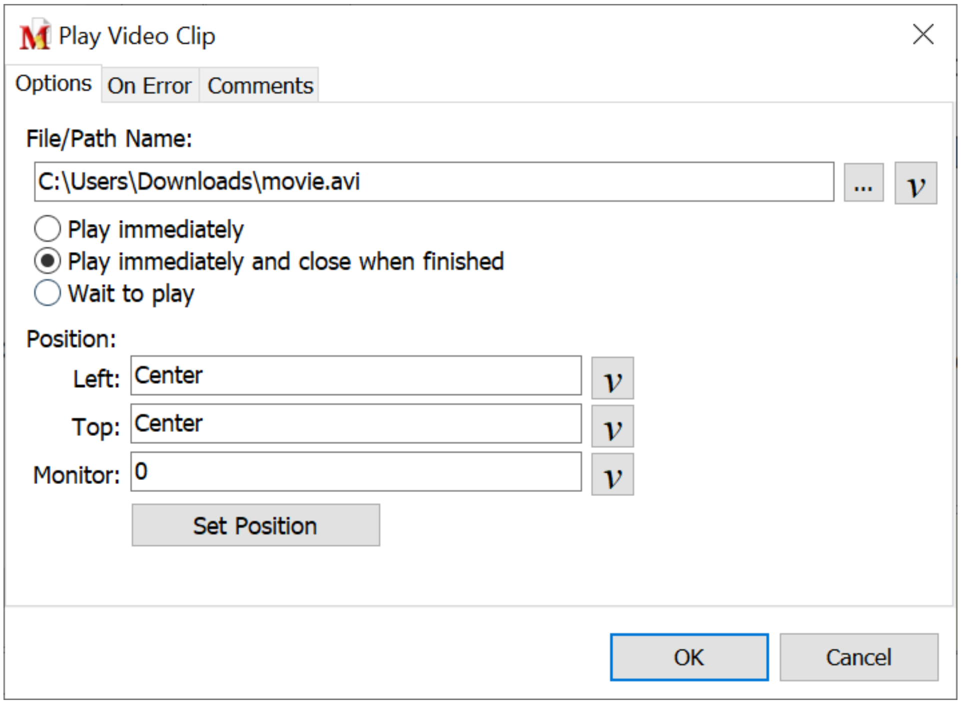
Script Editor > Expand Multimedia Category > Media Player Controls

Options available from the drop down list include:

- Play/Pause the media player
- Move to next track
- Move to previous track
- Stop the player.

Play Video Clip

Use this function to incorporate the playback of .avi format video files into macros.



Play Video Clip

File/Path Name: C:\Users\Downloads\movie.avi

○ Play immediately
● Play immediately and close when finished
○ Wait to play

Position:

Left: Center
Top: Center
Monitor: 0

Script Editor > Expand Multimedia Category > Play Video Clip

File/Path Name

Click on the browse button to find the video file to play back in the macro. After finding the file, click on the Open button to insert the file into the File/Path Name box. Or manually type the file path and name into this field.

Play Immediately

This option launches the AVI player and plays back the video.

Play Immediately & Exit When Done

This option launches the AVI player, plays back the video and then closes the AVI player.

✓ **Note:** When using this option, the macro cannot be stopped and the program cannot be closed until the video finishes.

Wait to Play

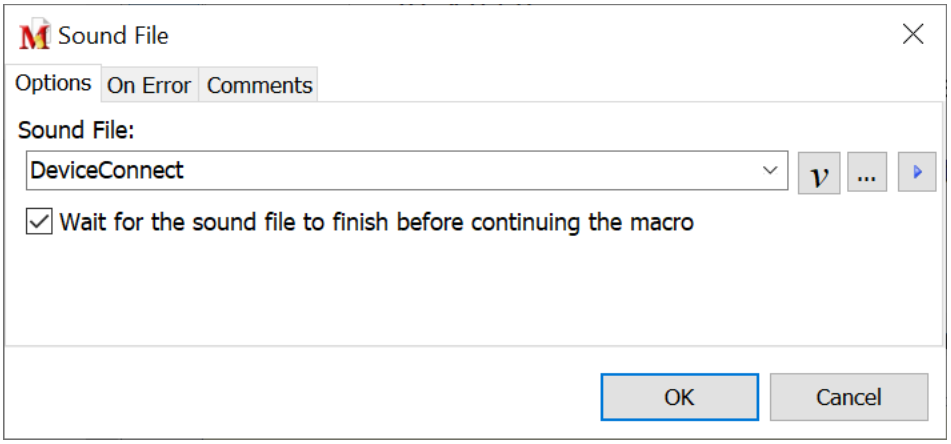
This option loads the AVI player and waits until instructed (click on the Start button of the AVI player) to begin the video playback.

Position

These options define the position on the screen where the AVI player will display during playback of the macro. Insert exact coordinates or use the Set Position Manually button to set the position. See the Window Positioning topic for additional information.

Sound File

The Sound File command adds sound to macros to help bring attention to a particular action or to just liven up the macro as it plays back.



Sound File: DeviceConnect

☑ Wait for the sound file to finish before continuing the macro

Script Editor > Expand Multimedia Category > Sound File

Sound File

This command plays a sound from a .wav file or one of the defined system sounds.

To select a sound file click the Browse button to locate a .wav file on your computer or type the full path and file name of the .wav file. You may also select one of the system sounds available on your computer by clicking on the drop down.

A system sound is a sound assigned to specific Program Events within Windows such as Asterisk, Device Connect or Windows Logoff. The sounds assigned to these events are selected in the Windows Control Panel either individually or as part of a theme. Not all system sounds listed in the Control Panel are available in this command.

The Default Beep sound, defined in the Control Panel, plays if a system sound is not defined or no sound is assigned to the specified system sound in the Control Panel.

Test

Click on the Test button to listen to the .wav file. This is a handy way to quickly test if the correct sound was selected. (The sound file runs immediately upon clicking the button.)

Wait for the sound file to finish before continuing the macro

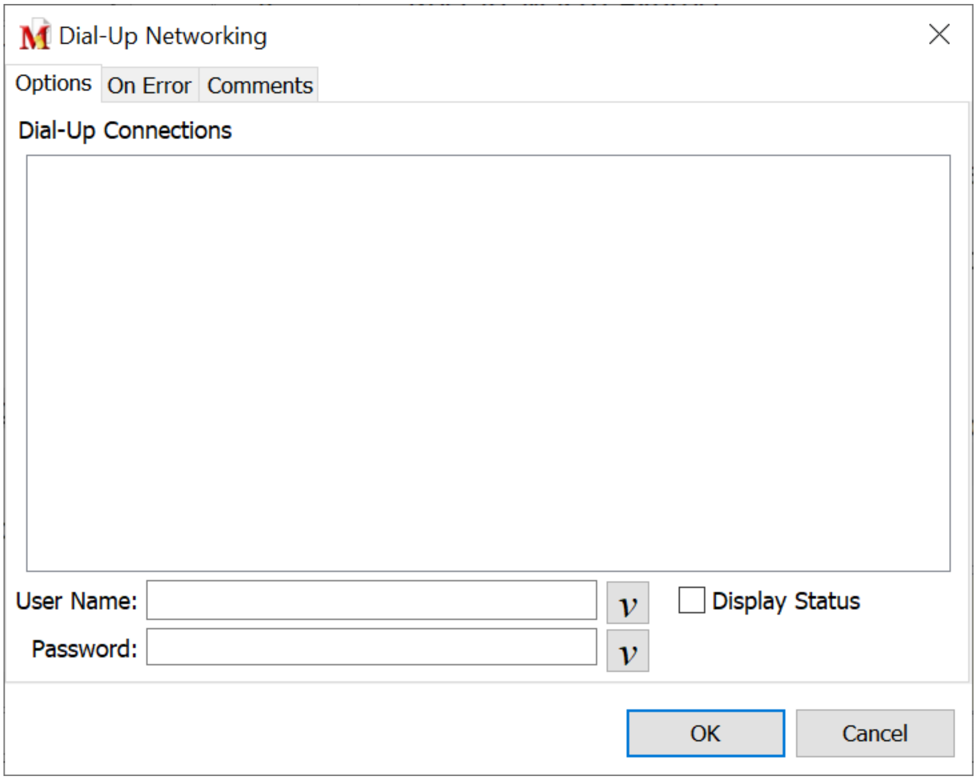
With this option checked, the execution of the macro stops until the .wav file has played in its entirety. If not checked, the sound file plays while subsequent macro commands execute.

Dial-Up Networking

*Commands deprecated - See note below

Dial-Up Networking connects the computer to another computer or to the Internet using a modem.

Dial-Up networking must be set up and configured in Windows in order to use this function. Use the Create a Dial-Up Connection or Change Dial-Up Settings options in Windows to do this.



Script Editor > Expand Network Category > Dial-Up Networking

Phone Book Entries

The Dial-Up Networking option displays a list of all the phone book entries created in Windows. If none are displayed, a phone book entry will first need to be created in Windows. Choose the entry to dial and connect.

User Name

Enter the user name for the account selected.

Password

Enter the password for the account selected.

- ✓ **Note:** When selecting a phone book entry, the User Name will most likely be displayed without requiring the name to be entered. Depending on the security of the system, the password may also be inserted. The actual password is not shown. A series of asterisks is displayed in its place.

Hangup

Choosing this option simply instructs the macro to hang up or terminate a Dial-Up Connection if one is currently active. There is not an edit window associated with this command. Double click on the Hangup command to insert it directly into the script.

*Deprecated Commands

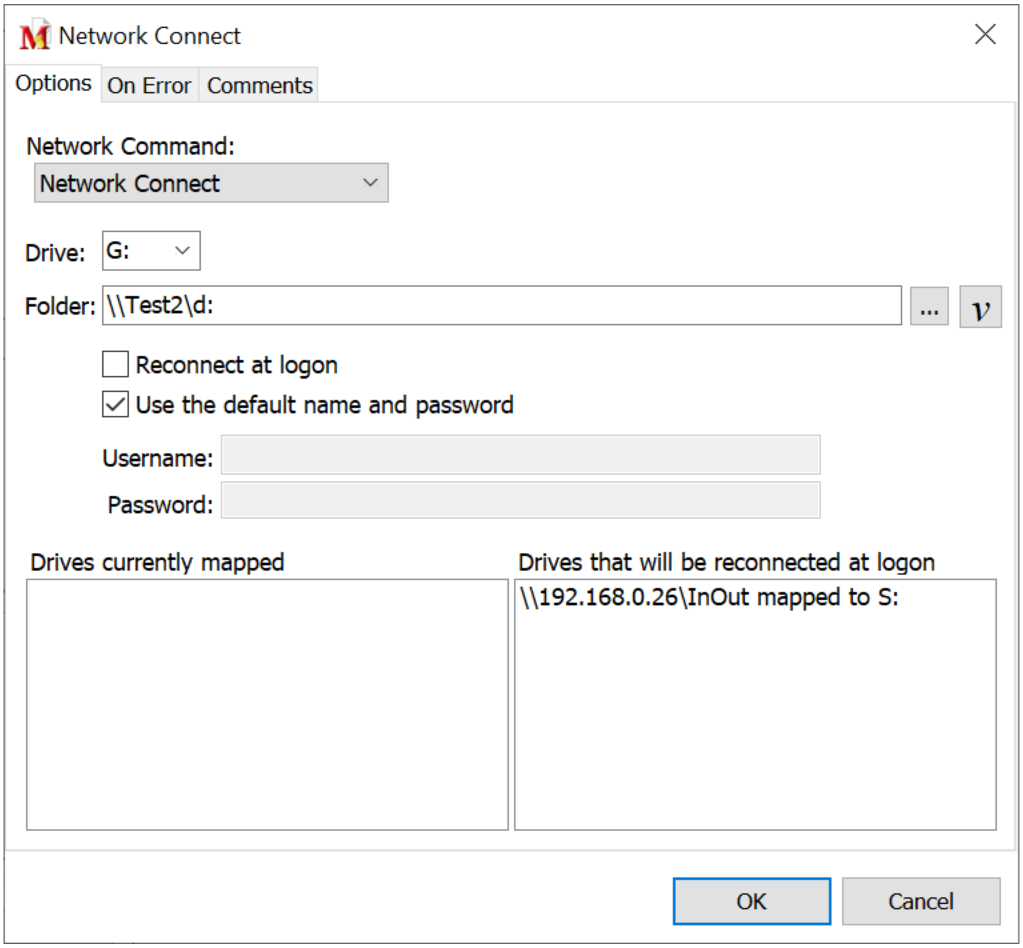
Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future. Where possible we recommend that you avoid the use of deprecated macro commands.

By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option found in Options, Preferences, General - Misc tab.

Network Connect

This macro command connects, disconnects or toggles network drives. Use this feature to access a network server or another computer, bypass the network neighborhood, quickly make temporary network connections or similar. The three network options are described below.



Network Command: Network Connect

Drive: G:

Folder: \\Test2\d:

☐ Reconnect at logon

☑ Use the default name and password

Username:

Password:

Drives currently mapped

Drives that will be reconnected at logon

\\192.168.0.26\InOut mapped to S:

Script Editor > Expand Network Category > Network Connect

Network Connect

The Network Connect command instructs Windows to map a drive letter to a drive on another computer.

Connect

Select from the available drive letters on the computer which drive is to be connected.

To Network Path

Enter the name of the drive and path to be connected. Either type this in directly or click on the Browse button to display all possible network drives available.

Make Connection Permanent

This option instructs Windows to remember the network connection created so that the next time the computer is started, the connection is made.

Use the Default Name and Password

This option instructs Windows to attempt to make the network connection using the home computer's login name and password. If not using this option, enter the login name and password required to access the network drive.

Network Disconnect

This instructs Windows to disconnect a drive from a network drive.

Disconnect

Select from the available drive letters on the computer which drive to disconnect from the network.

Make the Disconnect Permanent (Do not Remap on Startup)

This option instructs Windows to remember that this drive is not to be connected to a network drive the next time Windows is started.

Force the Disconnect Even if the Drive is in Use

This option instructs the macro to disconnect the network drive even if a program is using a file on the drive which is being disconnected. This action should typically not be taken unless absolutely sure that disconnecting will not have any adverse effects on programs currently using the drive.

Toggle Connection

This command is a combination of the Network Connect and Network Disconnect commands. It first checks to see if the drive is already connected to the network. If it is, it disconnects the drive. If the drive is not currently connected, then it makes the network connection.

Connect

Select from the available drive letters on the computer which drive to toggle.

To Network Path

Enter the name of the drive and path to toggle. Either type this in directly or click on the Browse button to display all possible network drives available.

Make Connection Permanent

This option instructs Windows to remember the network connection created so that the next time the computer is started, the connection is made.

Use the Default Name and Password

This option instructs Windows to attempt to make the network connection using the home computer's login name and password. If not using this option, then enter the login name and password required to access the network drive.

Display Connection Status

This option instructs the macro to display a dialog box indicating whether the connection succeeded or not. This occurs when the macro runs.

List Currently Mapped Drives

Use this option to display which drives are currently mapped to network drives.

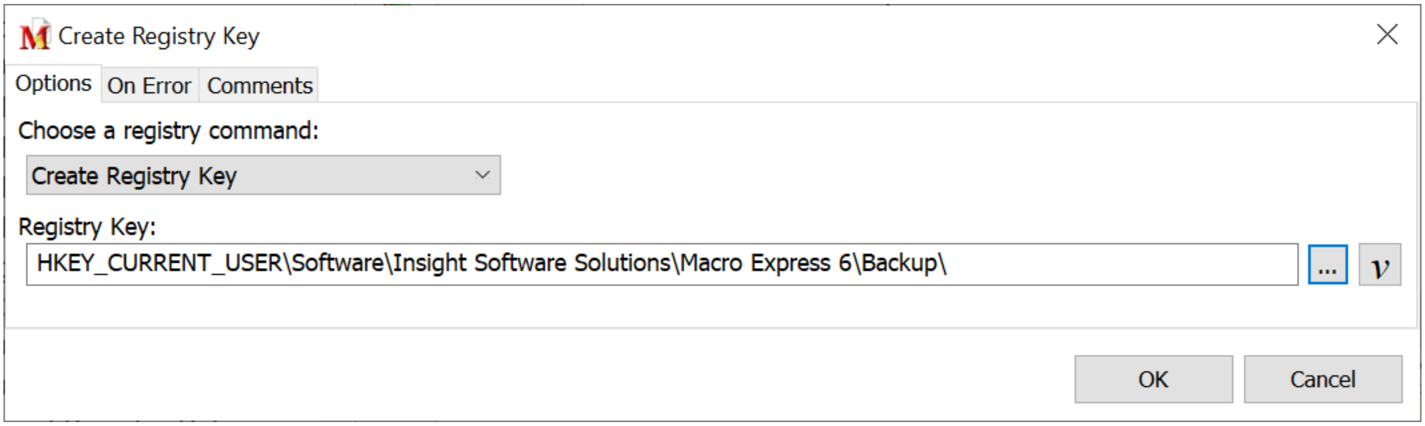
List Drives that Map on Startup

Use this option to display which drives are mapped as permanent connections -- ones that connect when the computer first starts.

Registry Key

These commands create or delete Windows Registry keys.

❖ A word of caution is in order here. Only those who have experience working with the Windows Registry should be using these commands. Otherwise problems may occur with the operating system.



Choose a registry command: Create Registry Key

Registry Key: HKEY_CURRENT_USER\Software\Insight Software Solutions\Macro Express 6\Backup\

Script Editor > Expand Registry Category > Registry Key

Choose a Registry Command

Select from the drop down list whether to create or delete a registry key.

Create Registry Key

Creates the specified Windows registry subkey in the key selected.

Delete Registry Key

Deletes the specified Windows key from the registry.

Registry Key

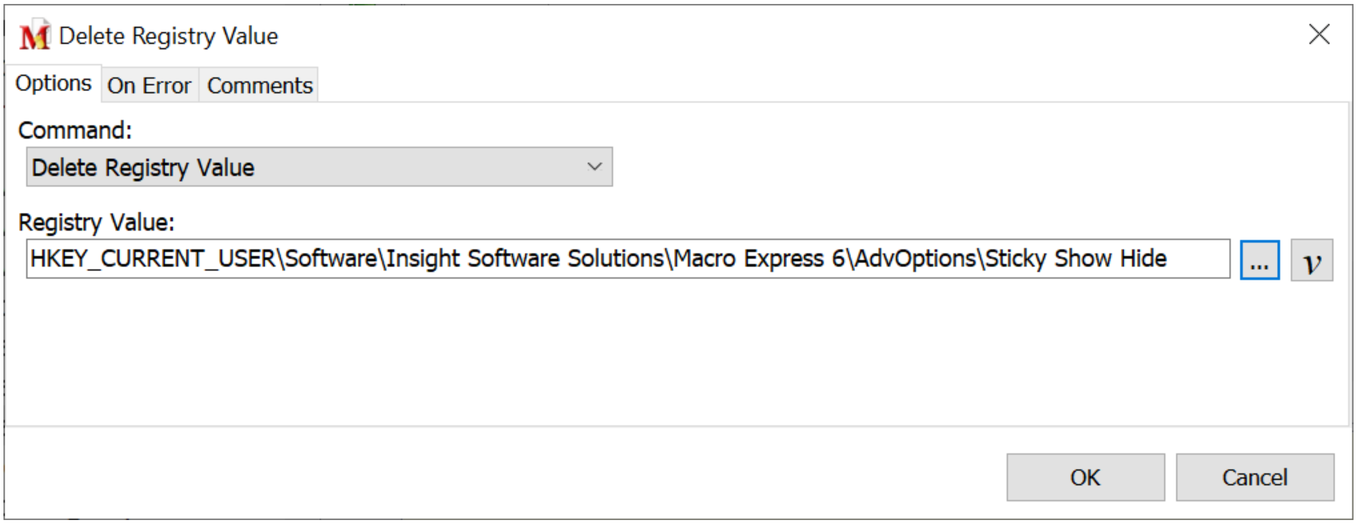
Select the registry path. Either type in the full path or click on the Browse button to navigate the Registry to find the key needed.

✓ **Note:** The Create Registry Key, Delete Registry Key, Write Registry Value and Delete Registry Value commands REQUIRE full Read/Write access to the registry in those areas that the user wishes to modify. The system administrator is responsible for assigning security permissions. Macro Express does not have the ability to modify Access Control Lists.

Registry Values

These commands read values from the Windows Registry and save them to a variable, write to the Windows Registry or delete values from the Windows Registry

- A word of caution is in order here. Only those who have experience working with the Windows Registry should be using these commands. Otherwise problems may occur with the operating system.



Command: Delete Registry Value

Registry Value: HKEY_CURRENT_USER\Software\Insight Software Solutions\Macro Express 6\AdvOptions\Sticky Show Hide

Script Editor > Expand Registry Category > Registry Values

Select from the command choices, choose a variable and then click on the Browse button to locate the registry value needed.

Read Registry Value

This command reads the value from the Windows registry key. This option does not require Write access to the registry.

Registry Value

Insert the Registry Value or click on the Browse button to locate the value needed.

Destination Variable

The registry value is saved to the variable selected.

Write Registry Value

This command writes a value to the specified registry key. This option REQUIRES

Write access to the registry.

Registry Value

Insert the Registry Value or click on the Browse button to locate the value needed.

Destination Variable

The value of the variable will be written to the key selected.

Delete Registry Value

Use this option to delete a value from a registry key. The Delete Registry Value command REQUIRES Write access to the registry.

Registry Value

Insert the Registry Value or click on the Browse button to locate the value needed.

Macro Express variables are stored in these registry Data Types:

Variable Type	Registry Data Type
Boolean	REG_DWORD
DateTimeVar	REG_BINARY
DecimalVar	REG_BINARY
HandleVar	REG_DWORD
IntegerVar	REG_DWORD
LargeIntegerVar	REG_BINARY
TextVar	REG_SZ

✓ **Note:** The Create Registry Key, Delete Registry Key, Write Registry Value and Delete Registry Value commands REQUIRE full Read/Write access to the registry in those areas that the user wishes to modify. The system administrator is responsible for assigning security permissions. Macro Express does not have the ability to modify Access Control Lists.

Continue

The Continue function may be used with the Repeat, Text File Begin Process or ASCII File Begin Process commands. Continue instructs the macro to ignore the remaining commands in the repeat loop. It skips directly to the End Repeat or End Process commands in the script and starts again with the next record or cycle.

Continue is typically used in conjunction with If statements. If a specific condition is met the macro will skip a section of the repeat loop. Otherwise the script continues as normal.

This command does not open an editable Macro Express window. Expand the Repeat category of commands and double click on the Continue command to insert it into the script.

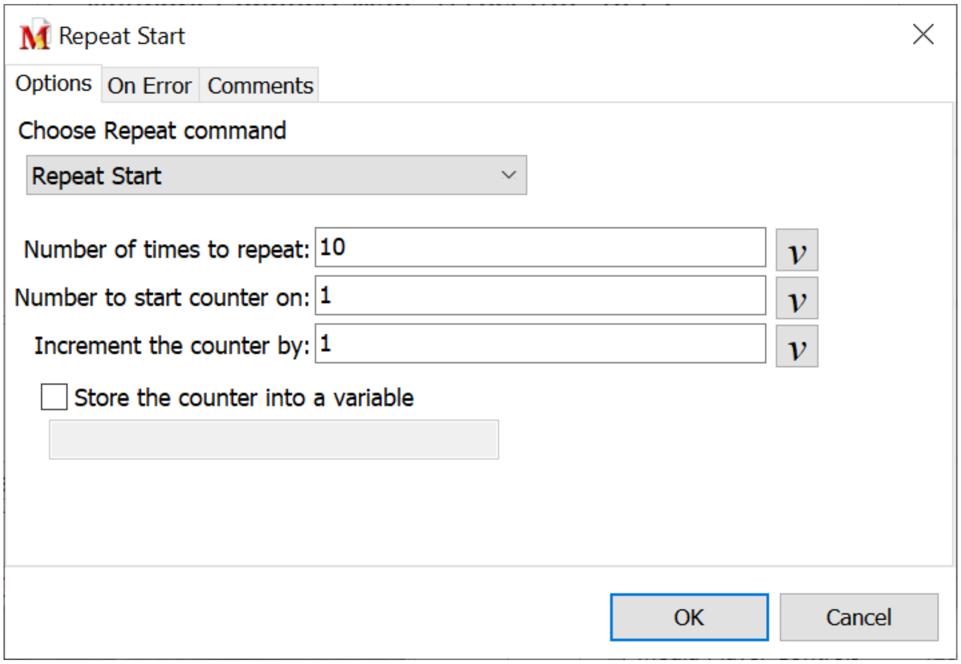
End Repeat

Each Repeat command requires a corresponding End Repeat command. All commands in the script between the Repeat and the End Repeat commands are repeated the number of times designated.

This command does not open an editable Macro Express window. Expand the Repeat category of commands and double click on the End Repeat command to insert into the script.

Repeat Commands

Use the Repeat commands to repeat a section or the entire macro any given number of times. All macro repeat loops must include an End Repeat command. All commands immediately following the Repeat command are repeated until an End Repeat command is encountered.



Choose Repeat command: Repeat Start

Number of times to repeat: 10

Number to start counter on: 1

Increment the counter by: 1

☐ Store the counter into a variable

Script Editor > Expand Repeat Category > One of Repeat Commands in this Topic

Start Repeat

The Start Repeat repeats a macro sequence the number of times requested. Enter the number of times to repeat the macro in the Number of Times to Repeat edit box.. Click on OK to save the settings. All macro commands after the Start Repeat are repeated until an End Repeat command is encountered.

Number of Times to Repeat

Insert the number of times to repeat the macro script.

Number to Start Counter On

This is currently only used in conjunction with a repeat counter. It specifies the starting point of the repeat counter.

Increment the Counter By

This is currently only used in conjunction with a repeat counter. It specifies an increment to apply to the repeat counter each time the repeat loops.

Counter Start and Stop Example

Let's look at an example using the Counter Start and Counter Stop. If we set the counter start to 5 and the counter step to 10 and had a repeat count of 3 then a repeat counter would have the following values:

During the 1st of 3 loops: 5
During the 2nd of 3 loops: 15
During the 3rd of 3 loops: 25

Start Repeat with Prompt

Use the Start Repeat with Prompt option to repeat a script a variable number of times. When the macro reaches this command in the script during playback, a prompt window displays prompting the user to insert the number of times to repeat the macro. Enter the number and the macro string repeats the number of times requested.

Use this option to avoid having to rewrite or edit the macro each time a variable number of repetitions is required.

Start Repeat With Variable

The Start Repeat with Variable option repeats the macro based on the value saved to the Integer variable selected. If the value of the Integer variable is 10, the macro is repeated 10 times.

Repeat Using Variable

Insert the Integer variable name in this field or click on the Variable button to select the variable previously defined.

Repeat Until

The Repeat Until option repeats a macro script again and again until a condition is met. The condition is the comparison of a text, integer or decimal variable to a fixed value. Comparisons can be equal, not equal, less than, greater than, less than or equal to or greater than or equal to. The comparison can also check to see if the variable contains or does not contain a specific text string, number, decimal, etc.

One example of this function is to create a macro that copies a line of text to the clipboard and then saves the clipboard contents to a variable. The Repeat Until function looks for a specific string of text. If the string of text is found, the macro breaks out of the Repeat loop and continues with the rest of the macro. If not, the macro moves to the next line in the document and repeats the process.

This option is also a way to run a macro continuously. For example set the Repeat Until option to repeat until N[1] = 1. Then do not insert any code in the macro to allow variable N[1] to equal 1.

Sample Macro - Repeat Until

In this example, we want to make sure that Microsoft Word is closed before the macro continues. The command to close Word must be placed before the macro commands listed below.

This macro has been set up in a repeat loop. Variable N[1] has been set to 0. The macro will repeat this loop until N[1] has been changed to 1. In this macro we have instructed the N[1] variable to change to 1 when it is detected that Word has closed.

```
Variable Set Integer %N[1]% to 0
Repeat Until %N[1]% = 1
  If Not Program Name "Word" running
    Variable Set Integer %N[1]% to 1
  End If
Repeat End
```

✓ **Note:** The macro commands between the Repeat Until and Repeat End commands are always executed at least once, because the Until condition is not evaluated until after the first iteration.

Repeat With Folder

This command scans through a folder and finds each file or subfolder in the folder. During the repeat process each successive file name or subfolder is placed in a text variable which can then be processed by other macro commands.

Sample Macro - Repeat with Folder

This macro counts the number of files in a particular folder. The count is saved to variable N[1]. Before starting the repeat loop, Integer variable N[1] is set to 0. Each time the macro repeats through the files, N[1] is incremented by one until a final count is made. After the loop is completed, a Text Box Display command has been added to display the file count.

```
Variable Set Integer %N[1]% to 0
Repeat with Folder
Variable Modify Integer: %N[1]% = %N[1]% + 1
Repeat End
Text Box Display: Files in folder
```

Folder

Select the folder to be processed. Either enter the path name directly or use the Browse button to locate the folder.

Return

Choose whether to return files only or folders.

Place Result In

Select the variable that will store the file or folder name.

Return the Full File Path

Use this option to save the full file path to the variable. Otherwise

Process Sub Folders

Optionally choose to have the macro recursively process all subfolders. This places all the file names or sub folders that are located in the folder or subfolders being processed into the variable.

Repeat with Windows

The Repeat with Windows command is similar to the Repeat with Folder command. It scans through all of the windows running on the computer, both visible and hidden. During the repeat process each successive window title is placed in a text variable which can then be processed by other macro commands.

Windows to Retrieve

Select whether to process through all windows, visible windows or hidden windows.

Sort Method

Select whether to process the windows alphabetically or in the order the windows are saved by the operating system.

Repeat with Processes

This command is also similar to the Repeat with Folder command. Repeat with Processes loops through all processes currently running on the computer. During the repeat loop the name of each successive process is placed in a text string variable which can then be acted upon by other macro commands.

- ✓ **Note:** Running processes are displayed in the Windows Task Manager.

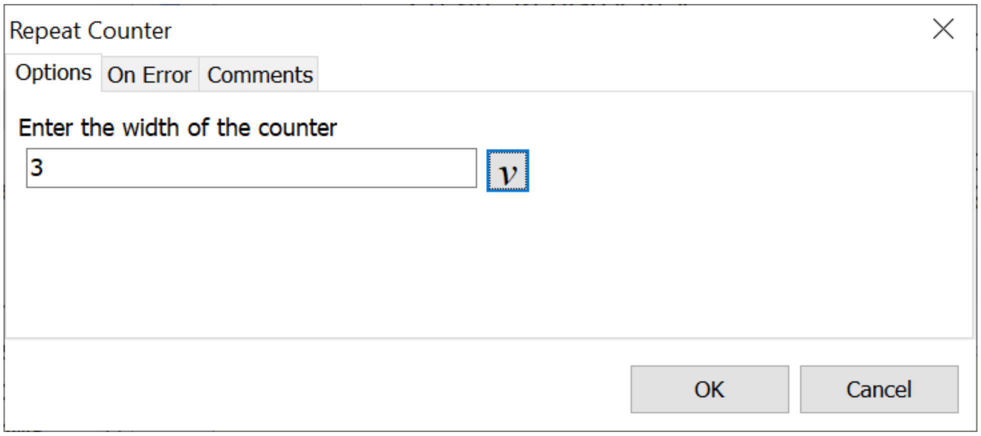
Store the Counter into a Variable

For the Start Repeat, Start Repeat with Prompt and Start Repeat with Variable commands, optionally place the repeat counter into a variable so that the counter value can be used and manipulated during the repeat process.

- ✓ **Notes:**
  1. Repeat commands can be nested (placed inside of each other).
  2. Repeat commands can cause excessively long macros. Remember, there is a way to abort a macro! Press the Scroll Lock and Pause buttons to stop the macro.

Repeat Counter

*Command deprecated - See note below



Enter the width of the counter

3

Script Editor > Expand Repeat Category > Repeat Counter

- ✓ We recommend using the Store the Counter into a Variable option that is part of the Repeat Start command.

A repeat counter command can be placed inside a repeat loop. What this does is type via the keyboard the current count of the repeat loop. For example, if creating a macro that repeated 5 times and the only thing inside the repeat loop was the incremental counter, it would type out "12345" when the macro is run inside a program such as Notepad.

The repeat counter can be especially useful for creating a unique name for each repeated loop. For example, while running a procedure multiple times, each result needs to be saved to a file name. Use the repeat loop and when the procedure saves the file name, add the incremental counter to the end of the file name. This provides a unique file name for each repetition.

If leading zeros are desired on counter values, then specify the "Counter Width". For example, to ensure that all values are at least 3 characters long, enter 3 for the width. This produces 001, 002, etc.

The syntax for the repeat command with a counter would be:

```
Repeat Start
  Repeat Counter: 3 digits wide
  Text Type: This is my macro text.
Repeat End
```

*Deprecated Commands

Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future. Where possible we recommend that you avoid the use of deprecated macro commands.

By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option found in Options, Preferences, General - Misc tab.

Repeat Exit

The Repeat Exit command is used to exit out of a repeat loop before the entire cycle is completed. This command would typically be used in conjunction with a conditional If statement. If the condition is met, the macro exits the repeat loop before the entire cycle is complete. The macro then continues with the command that follows the Repeat Exit.

The Break command may also be used to exit from a repeat loop. Use the Continue command if needing to skip to the end of the Repeat loop and start with a new repeat cycle.

This command does not open an editable Macro Express window. Expand the Repeat category and double click on the Repeat Exit command to insert it directly into the script.

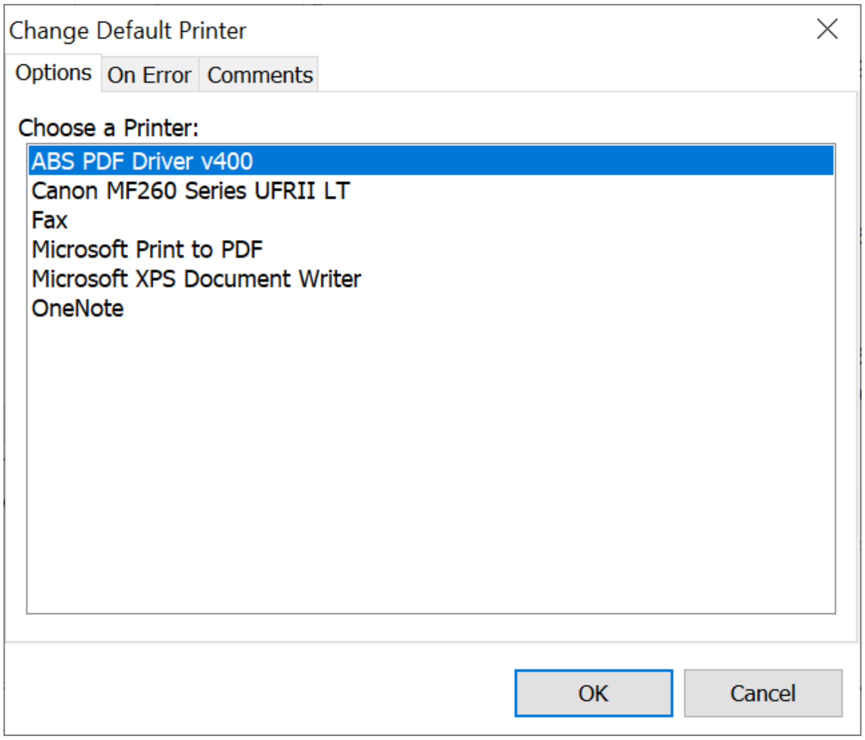
Sample Macro

Below is a modified Copy and Paste Routine macro demonstrated in the Tutorial. If specific text is found during the clipboard copy, the macro exits the repeat loop. Otherwise it continues cycling through the loop.

```
Repeat Start (Repeat 100 times)
  Window Activate: notepad
  Wait for Window Title: notepad
  Text Type (Simulate Keystrokes): <SHIFT><END>
  Clipboard Copy
  If Clipboard Contains "phrase I'm looking for"
    Repeat Exit
  End If
  Text Type (Simulate Keystrokes): <HOME><ARROW DOWN>
  Window Activate: wordad
  Wait for Window Title: wordpad
  Clipboard Paste
  Text Type (Simulate Keystrokes): <ENTER>
End Repeat
```

Change Default Printer

Use this feature when more than one printer is connected to the computer or network. For example, create a macro to quickly change the default printer setting from a monochrome laser to an ink jet for color printing.



Choose a Printer:

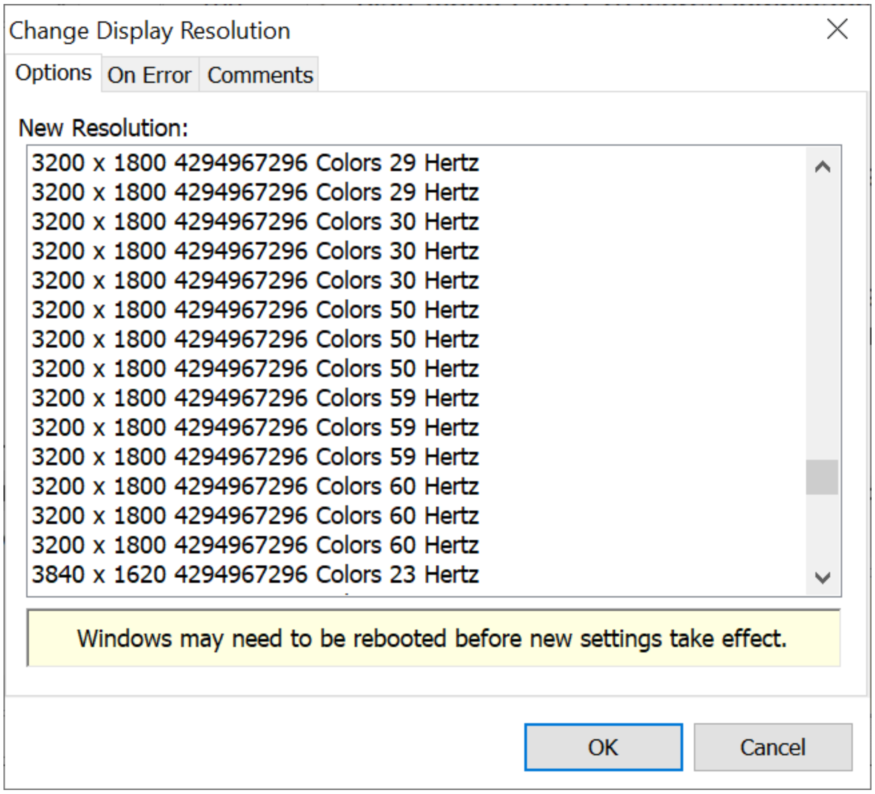
- ABS PDF Driver v400
- Canon MF260 Series UFRII LT
- Fax
- Microsoft Print to PDF
- Microsoft XPS Document Writer
- OneNote

Script Editor > Expand System Category > Change Default Printer

To select a new default, just click on the printer needed to highlight the selection. Click on the OK button to incorporate the printer change into the macro. When the macro is run, the printer selected becomes the new default printer for the computer.

Change Display Resolution

Macro Express provides a convenient method to quickly change the screen display resolution. For example, use this function with projects that require viewing at various screen resolutions. This option provides rapid changes between settings.



New Resolution:

```
3200 x 1800 4294967296 Colors 29 Hertz
3200 x 1800 4294967296 Colors 29 Hertz
3200 x 1800 4294967296 Colors 30 Hertz
3200 x 1800 4294967296 Colors 30 Hertz
3200 x 1800 4294967296 Colors 30 Hertz
3200 x 1800 4294967296 Colors 50 Hertz
3200 x 1800 4294967296 Colors 50 Hertz
3200 x 1800 4294967296 Colors 50 Hertz
3200 x 1800 4294967296 Colors 59 Hertz
3200 x 1800 4294967296 Colors 59 Hertz
3200 x 1800 4294967296 Colors 59 Hertz
3200 x 1800 4294967296 Colors 60 Hertz
3200 x 1800 4294967296 Colors 60 Hertz
3200 x 1800 4294967296 Colors 60 Hertz
3840 x 1620 4294967296 Colors 23 Hertz
```

Windows may need to be rebooted before new settings take effect.

Script Editor > Expand System Category > Change Default Printer

To select a screen display setting, just click on the screen resolution desired to highlight the selection. Click on the OK button to save and incorporate the screen resolution change into the macro.

- ✓ **Note:** Depending on the video driver, it may be necessary to reboot the computer before the new screen resolution settings take effect. If this is the case with the system, place a Reboot command in macro - after the screen setting command. This changes the screen resolution and then reboots the computer to initiate the settings.

Control Panel

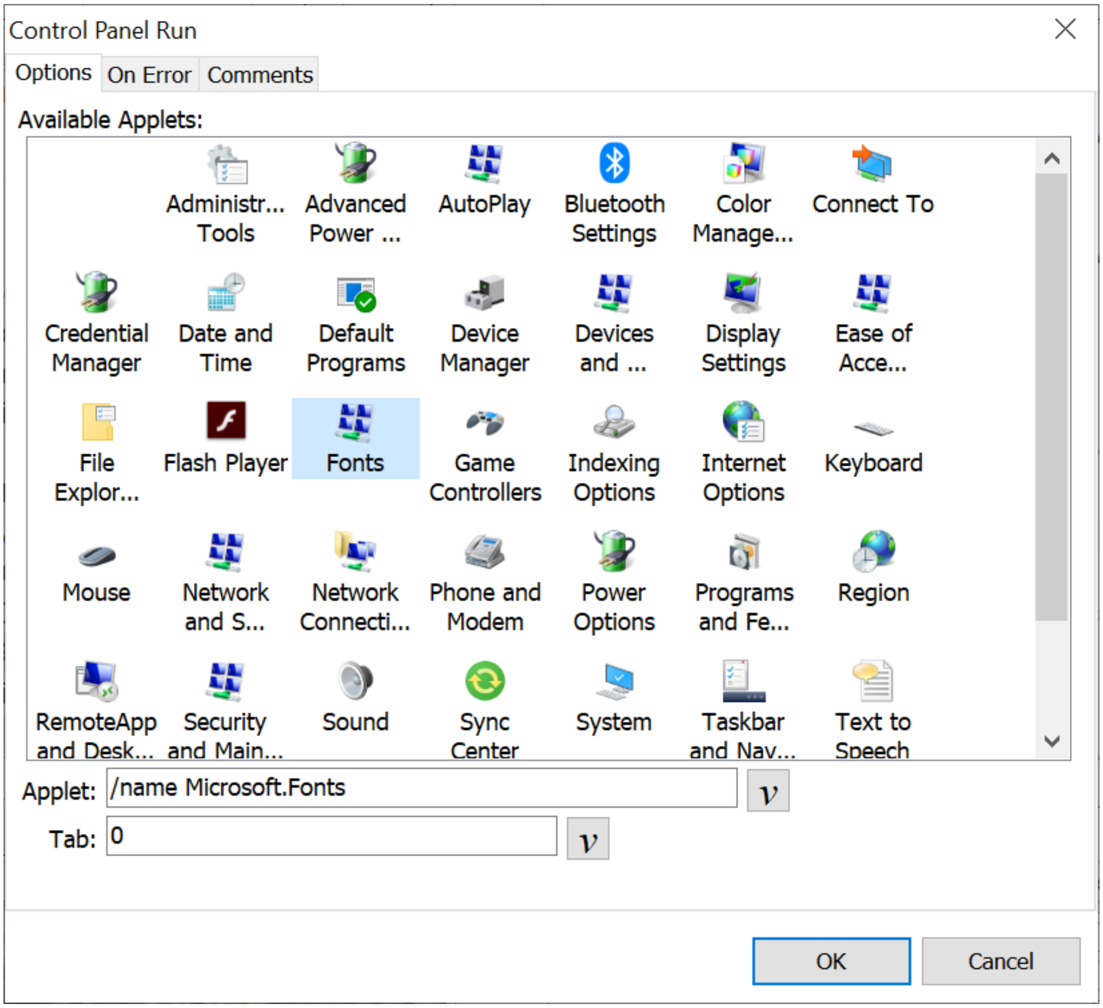
Use these two commands to open the Windows Control Panel or a specific applet in the Control Panel

Control Panel Open

The Control Panel Open function opens the Windows Control Panel. There is not an edit window associated with this command. The command is inserted directly into the script.

Control Panel Run

The Control Panel run command opens the Control Panel and the particular applet selected.



Applet: /name Microsoft.Fonts

Tab: 0

Script Editor > Expand System Category > Control Panel

Available Applets

The available Control Panel applets are displayed by name and icon. Highlight the applet to open.

Applet

If the applet is not displayed in the Available Applets box, enter the applet file name in the Applet edit box. Control Panel applets have a file extension of .CPL.

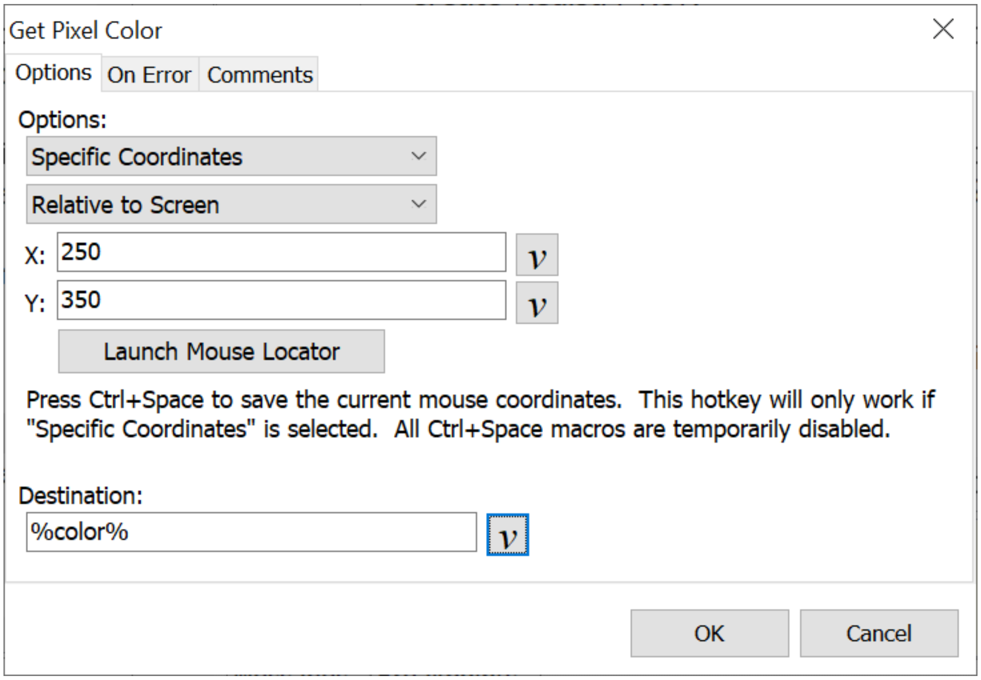
Tab

Many applets open a window with several tabs. Insert the tab number to display when the applet opens. For example, access with 0 set for the Tab opens the Accessibility Options window to the first Tab. If the Tab were set to 1, then the Accessibility Options window would open to the second Tab.

Get Pixel Color

Use the Get Pixel Color option to obtain the pixel color underneath the mouse pointer. The pixel color is assigned a numeric value by Windows, normally 6 – 8 digits in length. This numeric value is then saved to an Integer variable such as N[1]. For details about colors see How colors are displayed.

This command can be used for a variety of tasks. For example, use this command to determine if a check box has been checked or not. Or for a particularly difficult timing problem, the Get Pixel Color command may be the solution.



Options:
Specific Coordinates
Relative to Screen
X: 250
Y: 350

Press Ctrl+Space to save the current mouse coordinates. This hotkey will only work if "Specific Coordinates" is selected. All Ctrl+Space macros are temporarily disabled.

Destination:
%color%

Script Editor > Expand System Category > Get Pixel Color

Options

First select the option for determining the location of the pixel, either beneath the mouse or using specific coordinates.

Beneath the Mouse

This option obtains the color of the pixel located directly underneath the mouse pointer. The pixel color is obtained wherever the mouse cursor is located at the time this command takes place during the macro.

Specific Coordinates

With this option assign the specific mouse coordinates based on the active window or the entire screen. Insert the X and Y coordinates of the pixel to find.

Relative to Screen

Relative to Screen instructs the macro to move the mouse cursor relative to its position on the computer screen. Use the "Screen Coordinates" numbers from the Mouse Locator with this option.

Relative to Window

This option instructs the macro to move the mouse cursor relative to its position within the window that is active or has focus. Use the "Active Window" numbers from the Mouse Locator with this option.

Launch Mouse Locator

The mouse locator is an easy way to obtain the pixel coordinates when using the Specific Coordinates option. If the Mouse Locator window pops up in an area of the screen that covers important data, click on the window, hold the left mouse button down on the blue bar at the top of the window and drag to another location. See the Launch Mouse Locator topic for details on use.

Destination

Assign the pixel color obtained to a variable.

Sample Macro - Get Pixel

Here's an example of how this command could be used. Let's say that there is a web site that uses two alternating graphical images, one red and the other green. We want to determine which image is displayed and take appropriate action based on the results.

The first step is to load the web page. Use the Get Pixel Color command and select the At Specific Coordinates option, Relative to Screen.

Then launch the mouse locator from the Get Pixel Color window. Position the mouse cursor over the red image and press Ctrl Space. This inserts the pixel coordinates to the Get Pixel Color window. Write down the Pixel color on a piece of paper because we'll need this information later. Click OK to save.

With this information we can build the rest of the macro. The macro inserts a password if the image is red. If the image is not red (because the green image is currently displayed), it opens a new web page.

Macro Script

```
Get Pixel
If Variable %N[1]% = 255
Text Type: <TAB> 123xyz
Else
Web Site: https://www.macros.com
End If
```

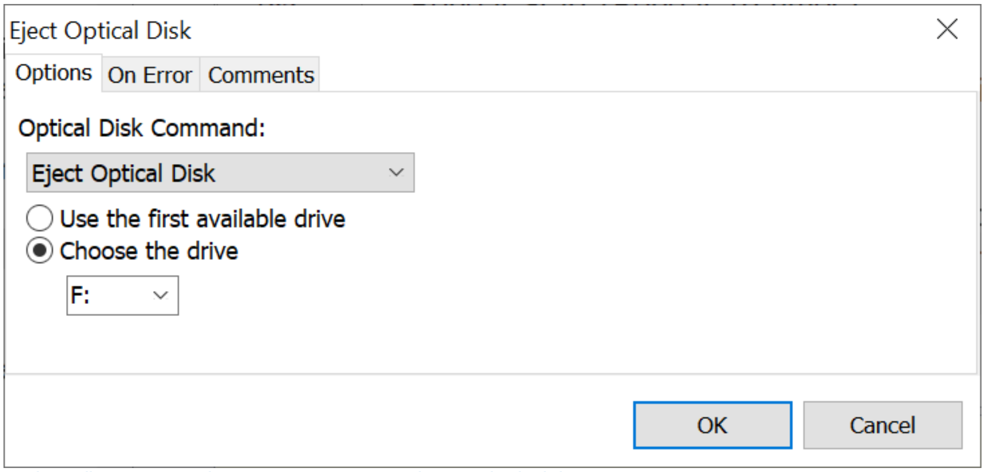
The pixel color for red is 255. This is the value we wrote down on the piece of paper to use with the If Variable command. So what we have done is use the If Variable command, stating if variable N[1] (what we assigned to the pixel color) equals a value of 255, then press a Tab and type in the password of 123xyz.

If the red graphical image is not present when the macro is run, then N[1] will have a different value (a value for green). It will not match the value of 255 in the If Variable statement. The Text Type command is skipped and the www.macros.com web page is launched instead.

Optical Disk

Eject Optical Disk

Use this command to eject an optical disk from the drive selected or to just open the disk tray.



Eject Optical Disk

Optical Disk Command: Eject Optical Disk

○ Use the first available drive

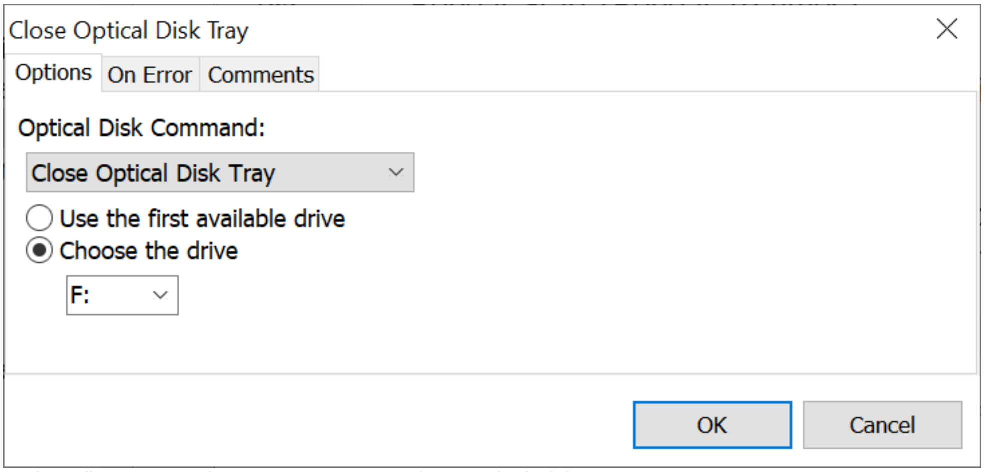
● Choose the drive

F:

Script Editor > Expand System Category > Eject Optical Disk

Close Optical Disk Tray

Use this function to close the disk tray selected.



Close Optical Disk Tray

Optical Disk Command: Close Optical Disk Tray

○ Use the first available drive

● Choose the drive

F:

Script Editor > Expand System Category > Close Optical Disk Tray

Use the first available drive

This option selects the first available disk drive. If there is only one disk drive attached to the computer, then use this option.

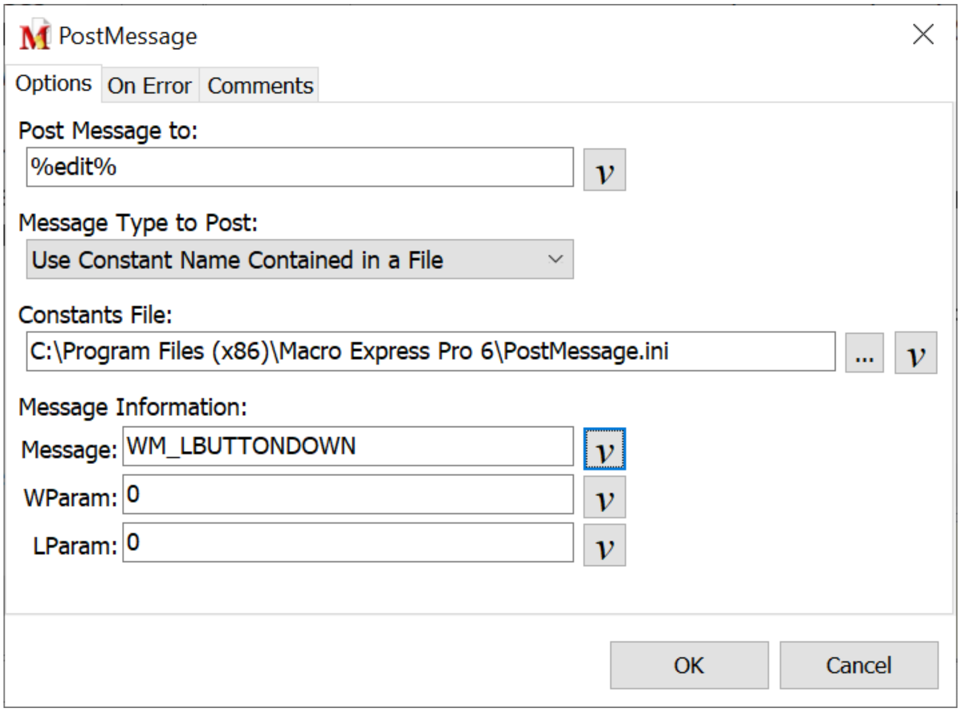
Choose the drive

If there is more than one disk drive on the computer, then choose from the drop down list which drive to use.

Post Message

The PostMessage command allows the user to send a Windows message to the specified control.

- ✓ **Note:** This command is designed for experienced users of Windows messaging. We are unable to provide support for explaining how Windows messages work, what are the various messages available and what they do, etc.



Post Message to: %edit%

Message Type to Post: Use Constant Name Contained in a File

Constants File: C:\Program Files (x86)\Macro Express Pro 6\PostMessage.ini

Message Information:

Message: WM_LBUTTONDOWN

WParam: 0

LParam: 0

Script Editor > Expand System Category > Post Message

Post Message To

Select the variable to receive the message information.

Message Type to Post

Next select the type of message to post. The five types are listed below.

Use Constant Name Contained in a File

The user specifies an ini file containing a list of constant names and their values. The "Message to Post" field should contain a constant name from this file.

An example would be the WM_LBUTTONDOWN message in the PostMessage.ini file. In the file, the following line appears:

WM_LBUTTONDOWN=513

This means that when the user specifies WM_LBUTTONDOWN as the message, message 513 is actually sent.

Constants File

The user specifies in the "Message to Post" field a message number to send to the specified control.

Use Numeric Value

The user specifies in the "Message to Post" field a message number to send to the specified control.

Offset from WM_USER

WM_USER is a Windows constant that marks the beginning of the messages that programmers can use for anything. In this case, the user specifies either a positive or negative number that is added to WM_USER. Currently, Windows defines WM_USER as 1024.

Use RegisterWindowMessage

This command allows the user to enter a constant string that can be used by Windows to generate a message. This is primarily used to communicate with other applications that have registered the same message.

Offset from WM_APP

WM_APP is a Windows constant that marks the beginning of the messages that programmers can use for anything. In this case, the user specifies either a positive or negative number that is added to WM_APP. Currently, Windows defines WM_APP as 32768.

Message Information

The message input into this field comes from one of three sources. The message will be a constant from the Constants file, a string passed to RegisterWindowMessage, or a value from WM_USER.

WParam - LParam

These fields allows the user to specify the WParam and LParam parameters. The contents of these parameters depend upon the message being sent. These values can be found in the Platform API library at https://msdn.microsoft.com.

Microsoft Definition of RegisterWindowMessage

The RegisterWindowMessage function defines a new window message that is guaranteed to be unique throughout the system. The message value can be used when sending or posting messages.

The RegisterWindowMessage function is typically used to register messages for communicating between two cooperating applications.

If two different applications register the same message string, the applications return the same message value. The message remains registered until the session ends.

Only use RegisterWindowMessage when more than one application must process the same message. For sending private messages within a window class, an application can use any integer in the range WM_USER through 0x7FFF.

- ✓ **Note:** When Macro Express is installed, three .ini files (PostMessage.ini, PostCommCtrl.ini and PostRichEdit.ini) are also installed into the Macro Express folder. The default path is C:\Program Files\MacExp3.0. These .ini files contain message definitions that may be used for reference. Double click on the files to open and view in Notepad or Wordpad.

Additional Window Message Information

The following quotes from Microsoft provide additional information about Windows Messages.

"The system passes input to a window procedure in the form of messages. Messages are generated by both the system and applications. The system generates a message at each input event - for example, when the user types, moves the mouse, or clicks a control such as a scroll bar. The system also generates messages in response to changes in the system brought about by an application, such as when an application changes the pool of system font resources or resizes one of its windows. An application can generate messages to direct its own windows to perform tasks or to communicate with windows in other applications.

The system sends a message to a window procedure with a set of four parameters: a window handle, a message identifier, and two values called message parameters. The window handle identifies the window for which the message is intended. The system uses it to determine which window procedure should receive the message.

A message identifier is a named constant that identifies the purpose of a message. When a window procedure receives a message, it uses a message identifier to determine how to process the message. For example, the message identifier WM_PAINT tells the window procedure that the window's client area has changed and must be repainted.

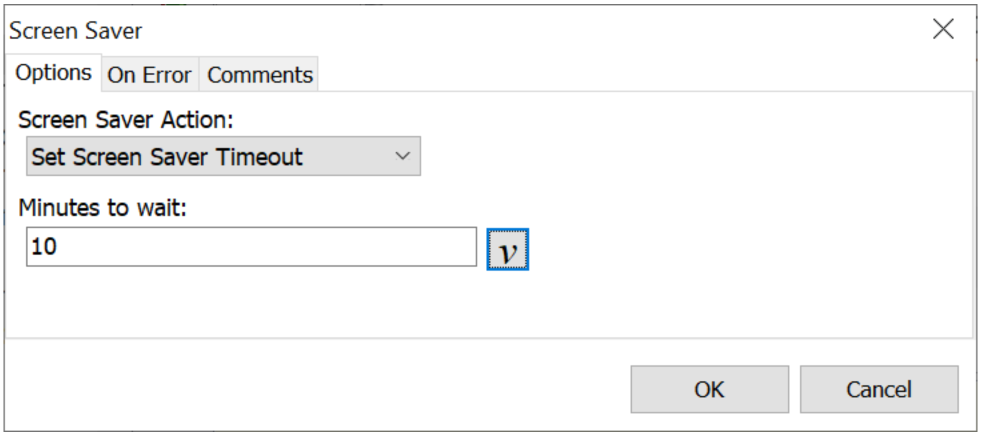
Message parameters specify data or the location of data used by a window procedure when processing a message. The meaning and value of the message parameters depend on the message. When a message does not use message parameters, they are typically set to NULL. A window procedure must check the message identifier to determine how to interpret the message parameters."

- ✓ **Note:** Do not confuse the PostMessage command with the Windows SendMessage command. The Windows SendMessage command allows several different types of data to be sent to an application. For example, a message parameter may contain an integer, packed bit flags, a pointer to a structure containing additional data, and so on.

  In addition, the SendMessage command allows a value to be returned to the program that sent the message. However, the Macro Express PostMessage command is patterned after the Windows PostMessage command. PostMessage only supports integer values in these parameters and does not support a returned value.

Screen Saver

These commands are used to make adjustments to the screen saver.



Screen Saver Action: Set Screen Saver Timeout

Minutes to wait: 10

Script Editor > Expand System Category > Screen Saver

Launch Screen Saver

This command immediately starts the screen saver if the screen saver option is enabled in Windows.

Disable Screen Saver

This command deactivates the screen saver setting. It changes the screen saver setting to read "none" meaning that no screen saver will activate. This is the same as selecting "Display" in the Control Panel and setting the screen saver to "none".

Set Screen Saver Time Out

This option changes the minutes of inactivity required before a screen saver starts. This is the same as selecting the "wait minutes" option in the screen saver tab of the Display Control Panel applet.

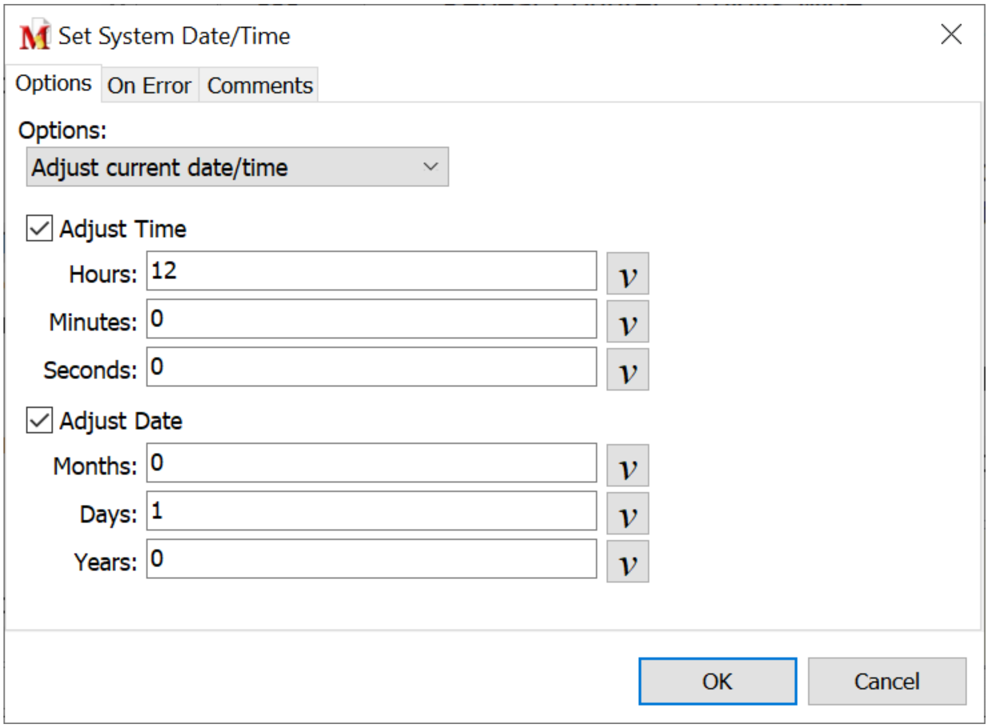
Minutes to Wait

Select the number of minutes of computer inactivity required before the screen saver starts.

Set System Date/Time

Use this option to change the system date and time. There are three ways to adjust the date and time. The first option changes the date and time based on the current settings, moving them ahead a fixed number of hours, days, etc. The second option sets the clock to a specific time and date. The last option syncs with a time server.

- ✓ **Note:** The Set System Date/Time command will not run in Vista or later due to the security features of these operating systems.



Options: Adjust current date/time

☑ Adjust Time

Hours: 12

Minutes: 0

Seconds: 0

☑ Adjust Date

Months: 0

Days: 1

Years: 0

Script Editor > Expand System Category > Set System Date/Time

Options

First select the date/time option desired.

Adjust Current Date/Time

Adjust Time

Use this option to set the system clock ahead by the number of hours, minutes or seconds specified. Or move it back by inserting a negative value in the field.

Adjust Date

Use this option to set the system date ahead by the number of months, days or years specified. Or move it back by inserting a negative value in the field.

Set a Specific Date/Time

Set Time

Select a specific time. When the macro is run, the system clock is changed to the time designated.

Set Date

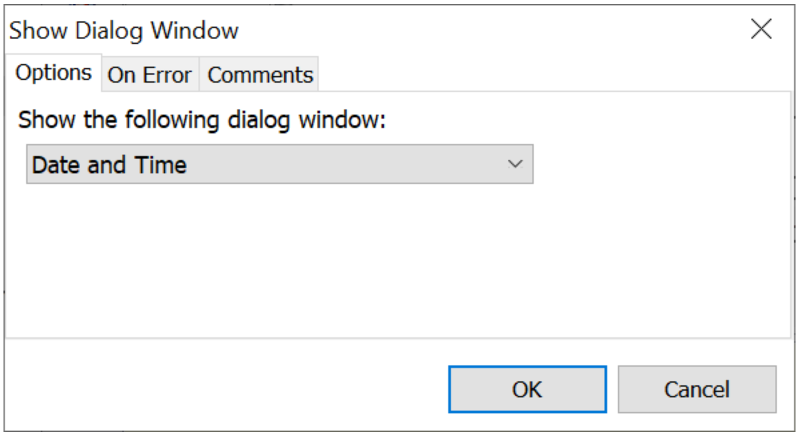
Manually enter a date or click on the arrow to display a calendar for picking a date. When the macro is run, the system date changes to the date selected.

Sync with Time Server

This option synchronizes the system time with a Time Server (web site). Enter the server address in the field provided. The system time on the computer is updated to match the time of the server.

Show Dialog Window

This command allows for a quick method to activate several standard Windows dialog boxes.



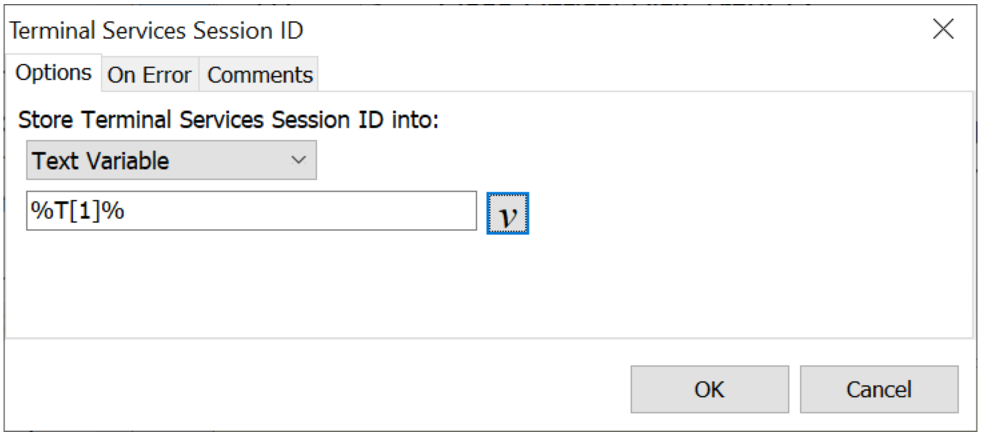
Show the following dialog window: Date and Time

Script Editor > Expand System Category > Show Dialog Window

Command	Description
Date and Time	Activates the Date and Time window for resetting the computer's clock, adjusting the date or changing the time zone.
Find Computers	Activates the Find Computer window in order to locate a computer on a network.
Find Files	Activates the Find Files window for searching the computer for a particular file.
Run Programs	Activates the Run Programs window in order to run a program, install a new one, etc.

Terminal Services Session ID

Windows Terminal Services provides the ability for several users to run Windows-based applications on a single server. The applications for each user run in a session. Each session is assigned a unique number. The "Store Terminal Services Session ID" command allows a macro to obtain this session ID number.



Store Terminal Services Session ID into:

Text Variable

%T[1]%

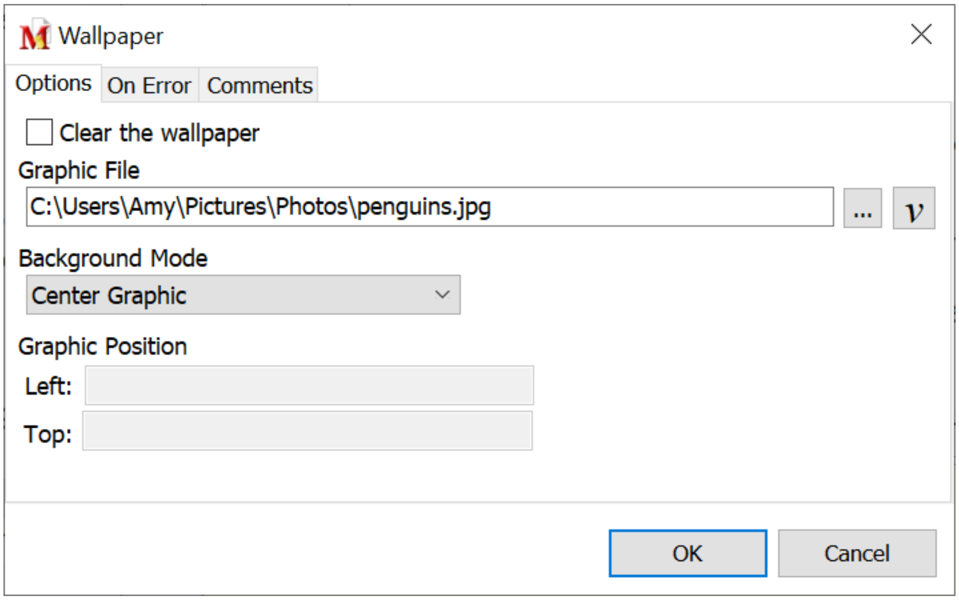
Script Editor > Expand System Category > Terminal Services Session ID

Some resources, such as the hard drive, **may be** shared between sessions. If a macro attempts to write to and then read from a file, macros used by different sessions may attempt to write to or read from the same file. This can lead to file corruption. By using the Session ID as part of the filename, the macro running on each session is assured of using a unique filename.

The session ID can be saved in either a text string variable or as an integer variable.

Wallpaper

If tired of looking at the same Desktop Wallpaper day after day, set up macros to incorporate images into the wallpaper. After setup, just press a hot key to change the wallpaper. Or create several timed macros to change the wallpaper at regularly scheduled intervals. This option changes the desktop wallpaper or background. The wallpaper is automatically updated in the Active Desktop as well, if the Windows operating system supports it.



□ Clear the wallpaper

Graphic File

C:\Users\Amy\Pictures\Photos\penguins.jpg

Background Mode

Center Graphic

Graphic Position

Left:

Top:

Script Editor > Expand System Category > Wallpaper

Clear the Wallpaper

Select this option to remove the current wallpaper. No image is displayed on the desktop.

Graphic File

Click on the browse button to find an image on the hard drive to use as the background wallpaper. A Windows Bitmap is the default file format. Other image file types that can be used are .jpg, .gif and .png. After locating the image, click on the Open button to save the file path/name.

Background Mode

Determine how to display the image on the desktop.

Command	Description
Tile Graphic	Makes copies of the image to tile across and fill the entire screen.
Center Graphic	Centers the image on the desktop
Stretch Graphic	Stretches the image to fill the entire screen.
Position Graphic	Position the image anywhere on the screen.

Graphic Position

When selecting the Position Graphic option, enter the X and Y screen coordinates to position the image on the desktop.

Windows System Commands

The following Windows system commands do not open an edit window when using in the editors. The commands are inserted directly into the editor. Expand the System category of commands and double click on the command to insert it into the macro script.

Command	Description
Empty Recycle Bin	This command essentially does what its name implies, empt Recycle Bin.
Hibernate Computer	Places a computer in hibernate mode, which saves everythir turns off the monitor and hard disk, and then shuts down th computer is restarted, the desktop is restored exactly as it w in hibernation.
Lock Computer	This function locks a computer workstation.
Log Off Computer	Closes all programs currently running and logs off of the com
Power Off Computer	Shuts down the computer and also shuts down the power to ✓ **Note:** The Power Off function is only available for computers that supp
Reboot Computer	Closes all running programs, shuts down and restarts the co
Restart Computer	Closes all running programs, shuts down and restarts the co
Sign Out Computer	Closes all programs currently running and signs out of the co
Sleep Computer	Puts the computer into a sleep mode, which places all work memory. A small amount of power is used during sleep mod
Shut Down Computer	Shuts down the computer.
Suspend Computer *Command Deprecated	Places a laptop in a suspend mode to save on battery power use the Hibernate Computer or Sleep Computer command.
Undock Computer	Allows a laptop or notebook computer to undock from a dock

*Deprecated Commands

Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future. Where possible we recommend that you avoid the use of deprecated macro commands.

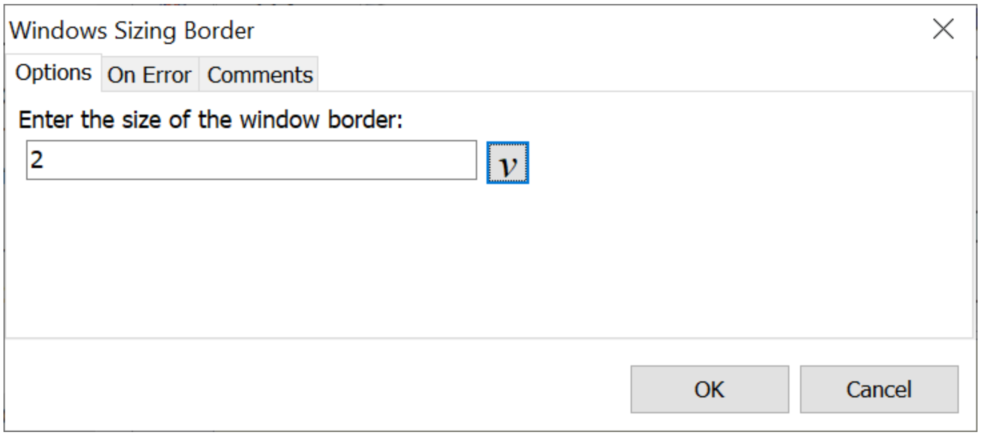
By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option found in Options, Preferences, General - Misc tab.

Windows Sizing Border

*Command deprecated - See note below

This command sets the sizable borders that appear around any re-sizable window. This may be useful when having difficulty positioning the mouse on the edge of a window.

- ✓ **Note:** This is a system command that will affect any window currently open or any that will subsequently be opened.



Enter the size of the window border:

2

Script Editor > Expand System Category > Window Sizing Border

Enter the size of the Window Border

Enter a value from 1 to 10 in the edit field. The number 10 places the thickest border around the window, while 1 places the thinnest border.

---

*Deprecated Commands

Some macro commands have been designated as deprecated. These commands may be infrequently used, use out of date terminology, or are no longer supported by newer versions of Windows. In some cases deprecated macro commands have been replaced by new ones.

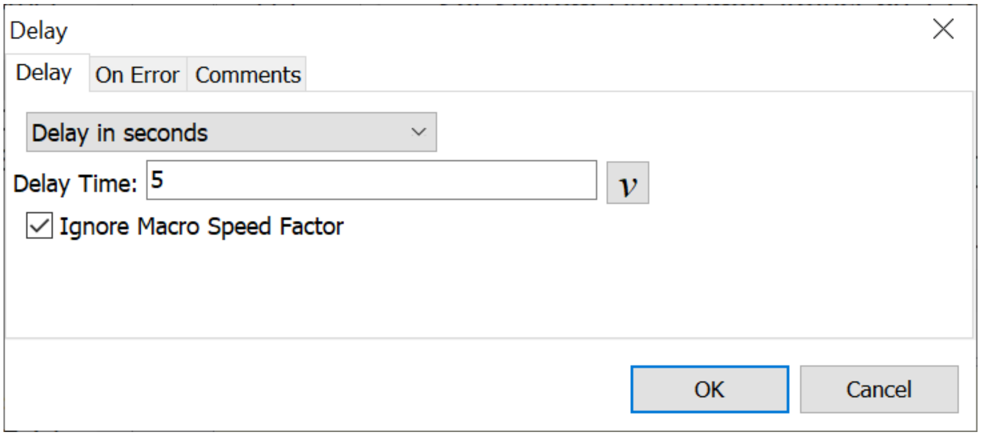
Deprecated commands are still supported in existing macros for backward compatibility. However, they may be removed from Macro Express in the future. Where possible we recommend that you avoid the use of deprecated macro commands.

By default deprecated commands are not displayed or accessible from the list of commands in the Macro Express Editor. To allow adding deprecated commands to a macro enable the Show deprecated commands option found in Options, Preferences, General - Misc tab.

Timing

Delays

Use this function to add Delay timing to macros. Delays may be necessary to allow time for a window to gain focus before the macro can proceed correctly or for any number of other reasons. The delay helps ensure that the macro doesn't play back faster than Windows can keep pace.



Delay in seconds

Delay Time: 5

☑ Ignore Macro Speed Factor

Script Editor > Expand Timing Category > Delays

Timed Delay in Seconds / Timed Delay in Milliseconds

The Timed Delay in Seconds option puts Macro Express into a "wait state" where virtually no CPU cycles are used, thus allowing CPU intensive programs to continue uninhibited. This is similar to the Wait for Time to Elapse command. Select either the Timed delay in seconds or Timed delay in milliseconds option and then enter the number of seconds or milliseconds in the Delay Time edit box.

- ✓ **Note:** This command suspends Macro Express for the amount of time specified. During this time the macro cannot be terminated by pressing the Scroll Lock + Pause keys or by right clicking on the "Running Man" icon in the notification area.

Delay in Seconds / Delay in Milliseconds

Delays can be set in seconds (up to 99999 seconds allowed) or in milliseconds (thousandths of a second). Just click on the option preferred and enter the number of seconds or milliseconds in the Delay Time edit box. Seconds can be in fractions of a second by using a decimal point such as .5 to signify a half second delay. Or 350 milliseconds equates to a little over 1/3 of a second delay.

Ignore Macro Speed Factor

This option instructs the Delay command to ignore the Macro Playback Speed so that the delay time remains constant. If this option were not checked and a Macro Speed command instructs the macro to play back at four times its normal speed, the delay would also be sped up. For example a one second delay would be reduced to a one quarter second delay.

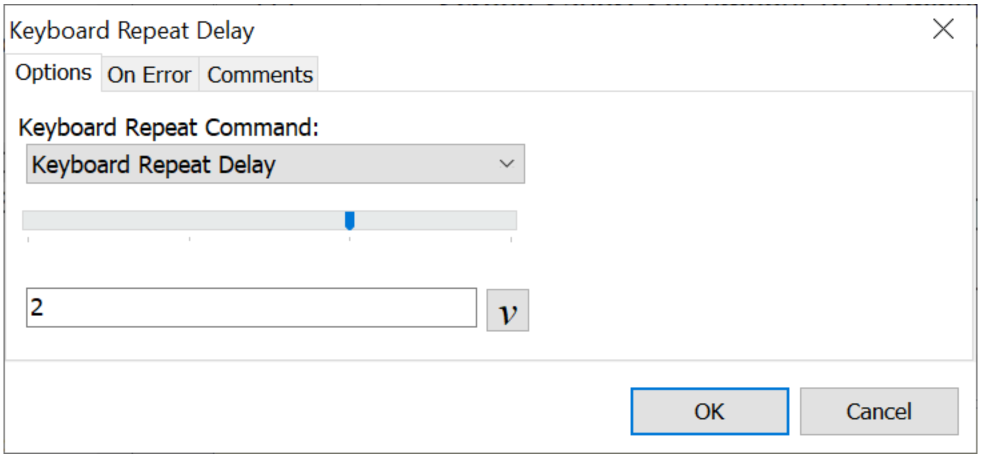
- ✓ **Note:** Delay timing halts the macro for at least the amount of time specified. The delay works by continually checking to see if the specified time has elapsed. While checking, any waiting or new Windows commands are processed. If a Windows command that takes a bit of time happens to be processed, then the delay may last longer than the time specified.

Keyboard Repeat

These commands adjust the keyboard speed.

Set Keyboard Repeat Delay

The Set Keyboard Repeat Delay changes the amount of time it takes before holding a key down on the keyboard to start repeating the key. Valid values are 0 through 3 with 0 being the shortest delay. This is the same as selecting "Repeat Delay" under the speed tab when selecting "Keyboard" in the Control Panel. Insert a value from 0 to 3 or adjust the slider bar to the desired setting.



Keyboard Repeat Delay

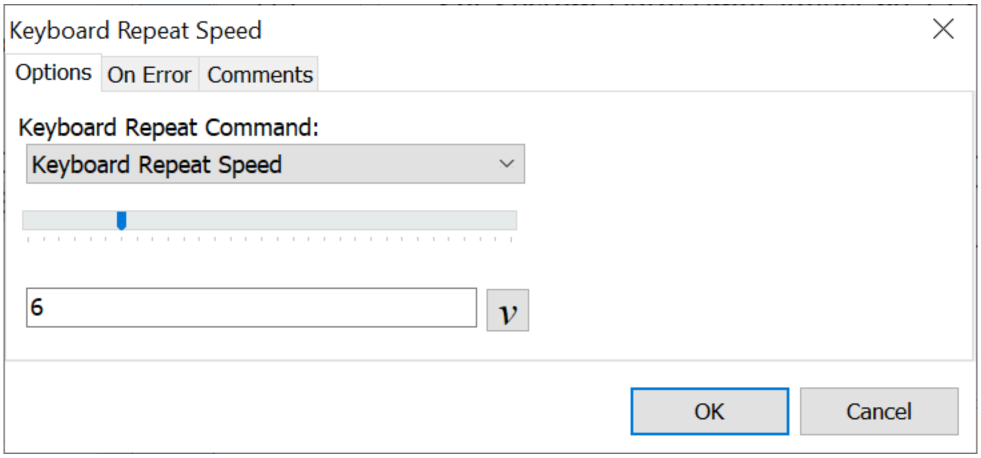
Keyboard Repeat Command: Keyboard Repeat Delay

2

Script Editor > Expand Timing Category > Keyboard Repeat Delay

Set Keyboard Repeat Speed

This option changes the speed of keys being repeated when it is being held down on the keyboard. Valid values are 0 through 31 with 0 being the slowest speed. This is the same as selecting the "Repeat Rate" under the speed tab when selecting "Keyboard" in the Control Panel. Insert a value from 0 to 31 or adjust the slider bar to the desired setting.



Keyboard Repeat Speed

Keyboard Repeat Command: Keyboard Repeat Speed

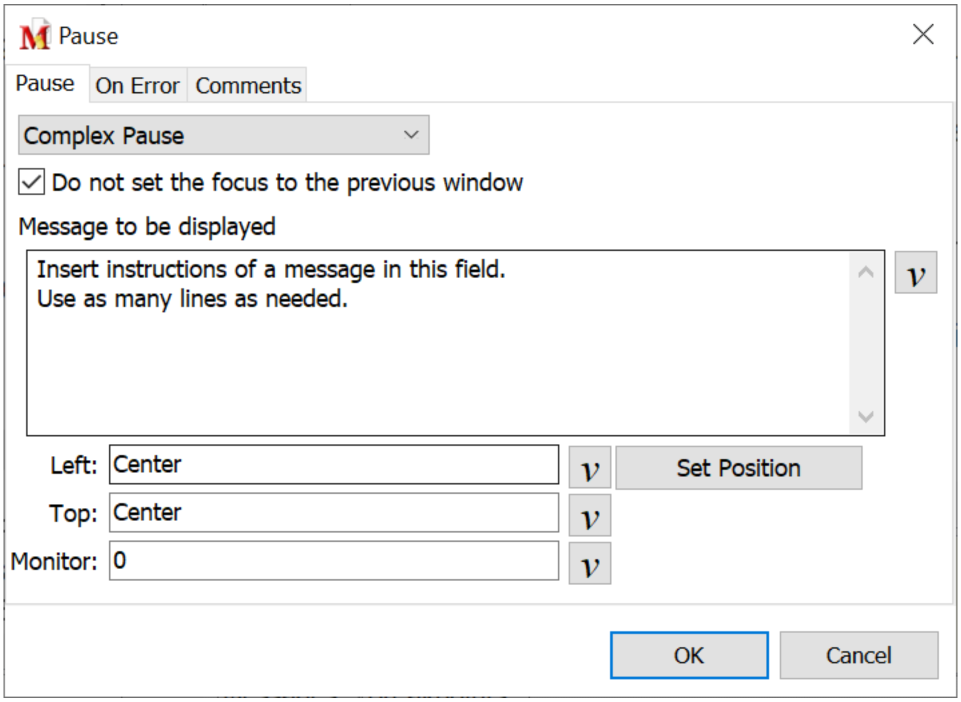
6

Script Editor > Expand Timing Category > Keyboard Repeat Speed

Pause

Macro Express allows for pauses to be placed in macros. When a macro is invoked, it plays back until the script reaches the pause command. When the pause is reached, playback stops and a small Pause window opens indicating that the macro has been paused. While the macro is paused, the mouse or keyboard may be used without adversely affecting the macro.

Resume the macro playback or cancel the macro at any time by clicking the appropriate button on the pause window. Or use the System Macro to resume macro playback. The default hot key setting is Ctrl + Alt + Shift + Y.



Complex Pause

☑ Do not set the focus to the previous window

Message to be displayed

Insert instructions of a message in this field.
Use as many lines as needed.

Left: Center

Top: Center

Monitor: 0

Script Editor > Expand Timing Category > Pause

Select from the drop down list whether to use the Basic or Complex pause.

Basic Pause

The Basic Pause stops the macro and displays a window in the center of the screen stating that the macro pause is in effect. Click on the Resume button, press the hot key assigned to the macro or press the system hot key to continue the macro playback. Click on the Abort button to cancel the remainder of the playback.

Complex Pause

The complex pause displays a message during the pause and allows for positioning the pause window so that it does not interfere with the application window in use.

Message to Be Displayed

Include an optional message to display in the pause window giving additional instructions to the user. Just type a message of up to 180 characters in the edit box.

When the pause message appears during macro playback, click on the Resume button to continue the macro playback or the Abort button to cancel the remainder of the macro. Also resume the playback of the paused macro by pressing the hot key assigned to the macro or by pressing the system hot key.

Location of Pause Window

Use this option to position where the Pause window is to be displayed on the screen during macro playback. See the Size and Position topic for more details about positioning a window.

Do Not Set the Focus to the Previous Window

Select this option to force the Pause window to retain window focus. If this option is left unchecked, the window open before the Pause window was activated keeps the focus. This option is available for either the basic or complex pause.

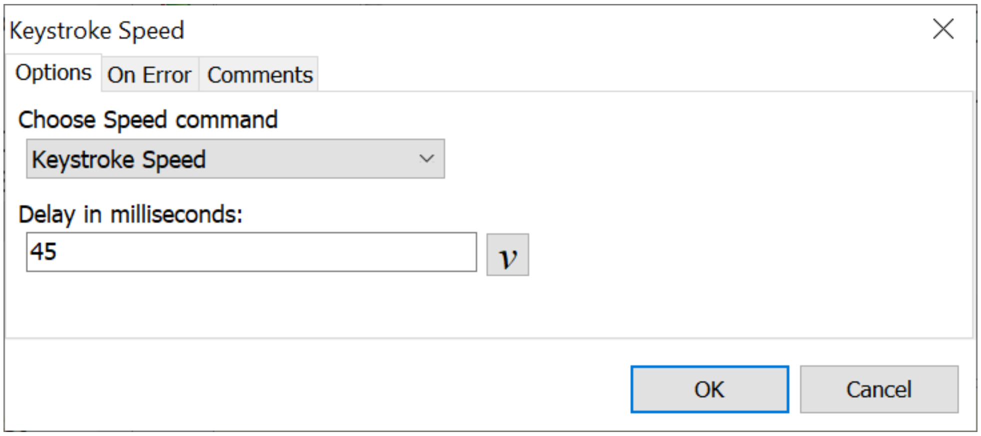
Other Ways to Pause a Macro

1. The Wait for Key Press command causes the macro to pause. It instructs the macro to wait until a specific key is pressed. The macro resumes once the key is pressed. This command does not allow for a message to display while the macro is paused.

2. If the macro is paused so that the user can insert text, another option is the Variable Set String command - Prompt for Value option. This command prompts the user to type text into the prompt. The text is saved to a variable which can then be inserted into another application using the Text Type command. This also applies to the Variable Set Integer and Variable Set Decimal commands.

Speed

The Speed commands adjust the playback speed of keystrokes or mouse movements or adjusts the playback rate of the entire macro.



Choose Speed command: Keystroke Speed

Delay in milliseconds: 45

Script Editor > Expand Timing Category > Speed

Keystroke and Mouse Speed

It is possible for the keystrokes or mouse movements sent by Macro Express to move too quickly in the intended application or for Windows to keep pace. This option slows down the playback speed so that the macro runs correctly.

Delay in Milliseconds

Enter the delay time, in milliseconds (thousandths of a second), to place between each keystroke or mouse playback.

In some cases it may only be necessary to slow down part of the macro. In this situation place the speed command before the section of the macro that needs to be slowed down. Then place another speed command after the slowed down section of the macro with a delay set to "0" zero. The remainder of the macro resumes with no delays between keystrokes and mouse movements.

It is also permissible to place a Keystroke Speed delay and a Mouse Speed delay in the same macro. The delays remain constant throughout the macro until another speed command is inserted with a different delay setting.

Macro Speed

The macro speed function is typically used with capturing (recording) a macro. The Macro Speed command occupies the first line of a recorded macro script. The speed factor is initially set for captured macros in the Capture preference settings.

With all the timing delays between each keystroke and mouse movement recorded, it may be advisable to speed up or slow down the macro. Double click on the Macro Speed command in the script to edit the speed factor.

Multiply Delays By

To slow down a macro, place a number greater than one in the Multiply Delays By field. For example, placing a 4 in this field causes the macro to play back 4 times slower. To speed up the macro, place a value less than one in the Multiply Delays By field. For example, placing 0.5 in this field speeds up the macro by a factor of 2, using 0.25 by a factor of 4, etc.

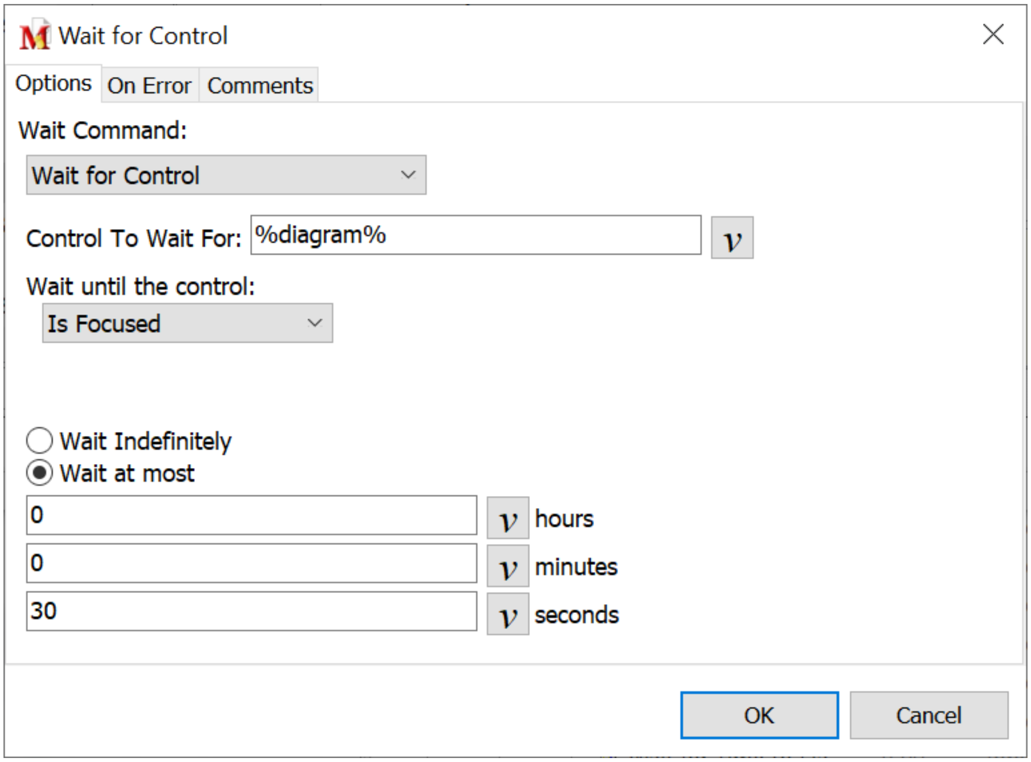
Wait For

Wait for Control

This command allows the user to create a macro that waits for a specific Window Control to gain focus, become visible or become enabled. Or have the macro wait for their opposites - wait for control to lose focus, become invisible or become disabled. So, the user could write a macro that looks like the following:

> Wait for Control (C[1]) to be Enabled
> Mouse Click on Control (C[1])

This would cause the macro to click on the control as soon as the control would allow.



Wait Command: Wait for Control

Control To Wait For: %diagram%

Wait until the control: Is Focused

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
30 seconds

Script Editor > Expand Timing Category > Wait for Control

Control to Wait For

First a Window Control variable needs to be defined in the macro. Then use this function to select the control for which the macro will wait.

Wait Until the Control

Select the status of the control. Choose from the following options - If the Control:

- Exists
- Does not Exist
- Is Enabled
- Is not Enabled
- Is Visible
- Is not Visible
- Is Focused
- Is not Focused

Wait Indefinitely

With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

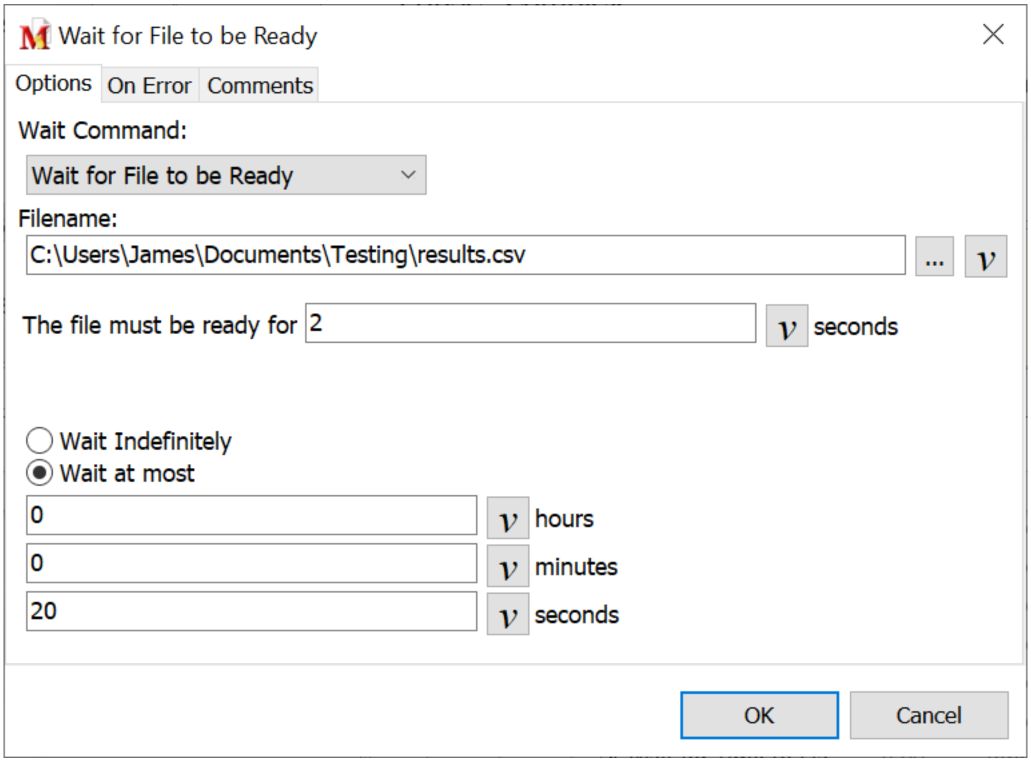
With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for File to Be Ready

This command instructs a macro to wait until a file is ready before continuing. If a macro launches a particular file, it could wait until the application is loaded and the file is ready, before allowing the subsequent steps to proceed. This is a different approach than using a Wait for Window Title command.

Use this command in situations where the macro should continue only after an application has finished writing to the file. The file may exist for some time before the application has finished writing to it. The Wait for File to Be Ready command waits until the file is no longer being written to.

- ✓ **Note:** If the file does not have write permissions then the macro will always flag the file as not being ready. In Windows Vista and later, any files located in the Program Files folder do not have write permission.



Wait Command: Wait for File to be Ready

Filename: C:\Users\James\Documents\Testing\results.csv

The file must be ready for 2 seconds

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
20 seconds

Script Editor > Expand Timing Category > Wait for File to Be Ready

Filename

Insert the entire file path and drive or click on the Browse button to navigate to the location of the file.

File Must be Ready for xx Seconds

Insert the number of seconds that the file should be ready before the macro continues. This helps ensure that the file is ready for other activity before the macro continues.

Wait Indefinitely

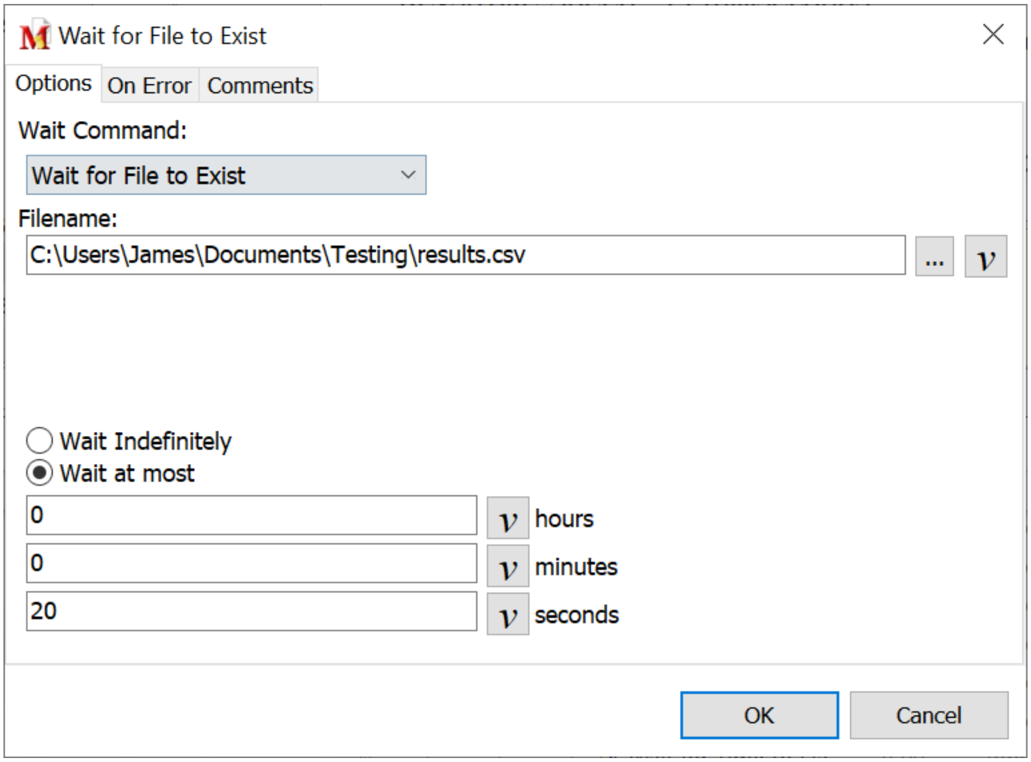
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for File to Exist

This option waits for a specific file to exist. As an example, a macro creates a new file and then uses the Wait for File to Exist command to instruct the macro to wait until the file has been created before continuing.



Wait Command: Wait for File to Exist

Filename: C:\Users\James\Documents\Testing\results.csv

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
20 seconds

Script Editor > Expand Timing Category > Wait for File to Exist

Filename

Insert the entire file path and drive or click on the Browse button to navigate to the location of the file.

Wait Indefinitely

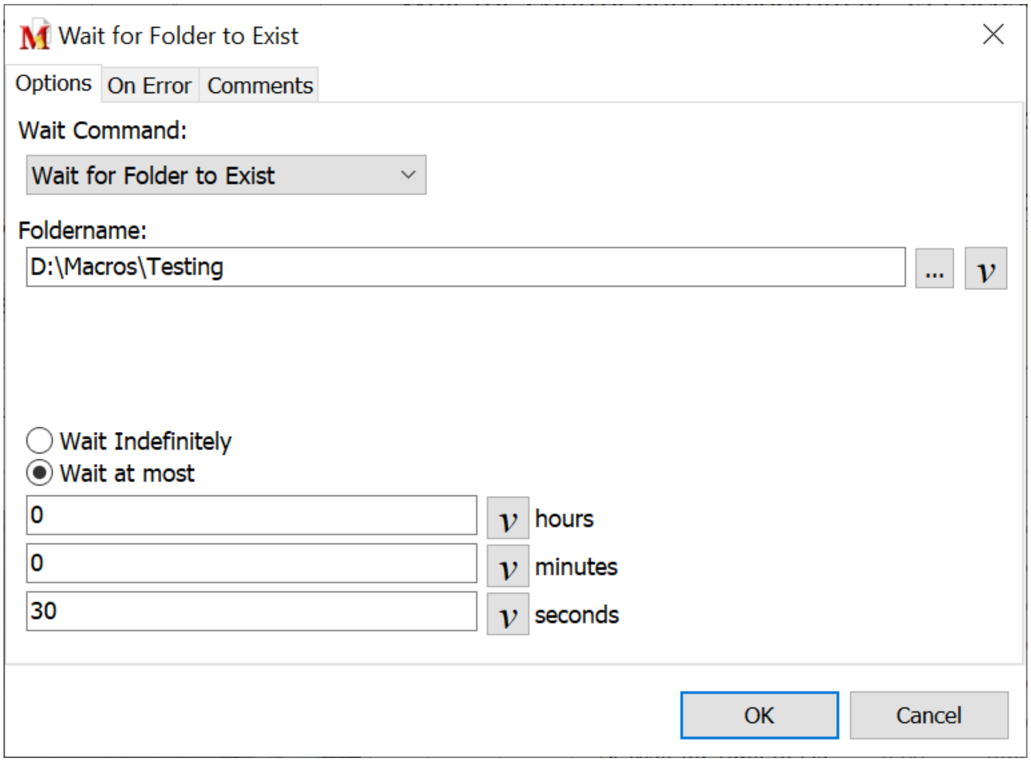
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Folder to Exist

This option is similar to the Wait for File to Exist command. It instructs the macro to wait until a specific folder has been created or exists on the computer before continuing.



Wait for Folder to Exist

Wait Command: Wait for Folder to Exist

Foldername: D:\Macros\Testing

○ Wait Indefinitely

● Wait at most

0 hours

0 minutes

30 seconds

Script Editor > Expand Timing Category > Wait for Folder to Exist

Foldername

Insert the entire folder path and drive information or click on the Browse button to locate the folder.

Wait Indefinitely

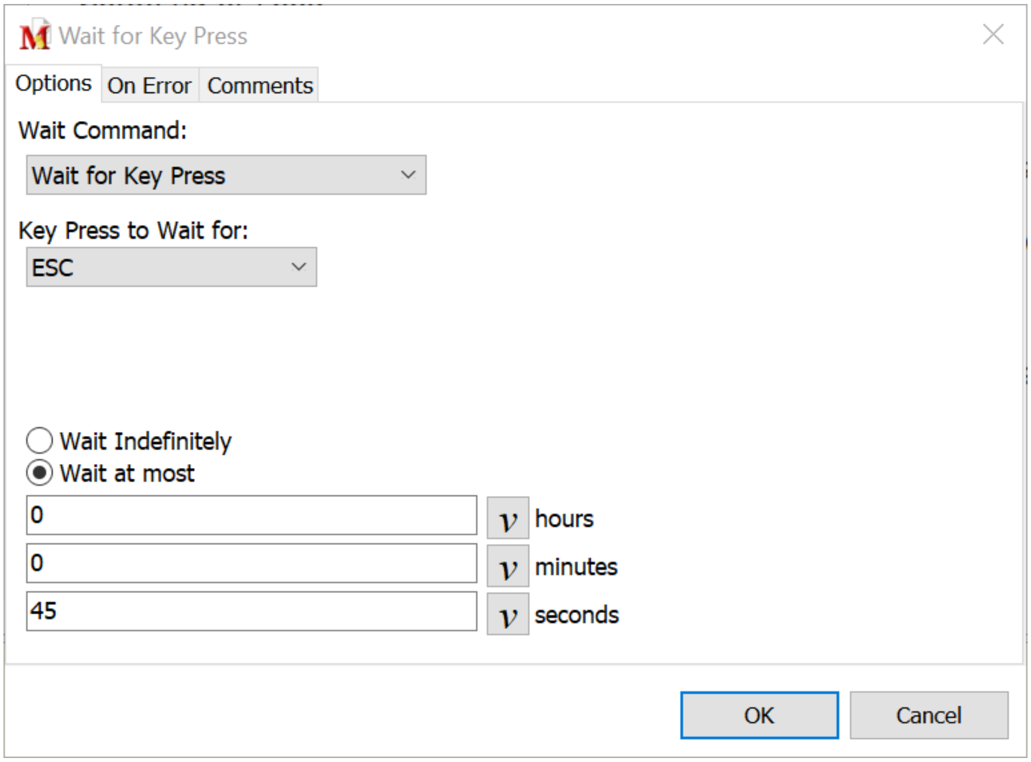
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Key Press

The Wait for Key Press command instructs the macro to wait until a specific key is pressed. The user may input text, move the mouse, etc. until that key is pressed. Once the selected key is pressed, the macro resumes.



Wait Command: Wait for Key Press

Key Press to Wait for: ESC

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
45 seconds

Script Editor > Expand Timing Category > Wait for Key Press

Key Press to Wait for

Select the key to use from the drop down list. During playback, the macro stops when it reaches the Wait for Key Press command in the script and waits for the selected key to be pressed. Choices include letters of the alphabet, the function keys or several other common keys such as Enter, Tab, etc.

- ✓ **Note:** Use caution when this command follows a Text Type command. For example, a macro contains the command "Text Type: <TAB>" and the next command in the script is the Wait for Key Press command. Since the previous command is to play back the TAB key, do not select the TAB key as the Wait For key. Macro Express sees the TAB from the Text Type command, thinks it is finished waiting and continues with the macro. It looks like the Wait for Key Press did not function properly. In this situation, either select a different key or insert a Delay command after the Text Type command. Or use the Wait for Text Playback command. This ensures proper macro functionality.

Wait Indefinitely

With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

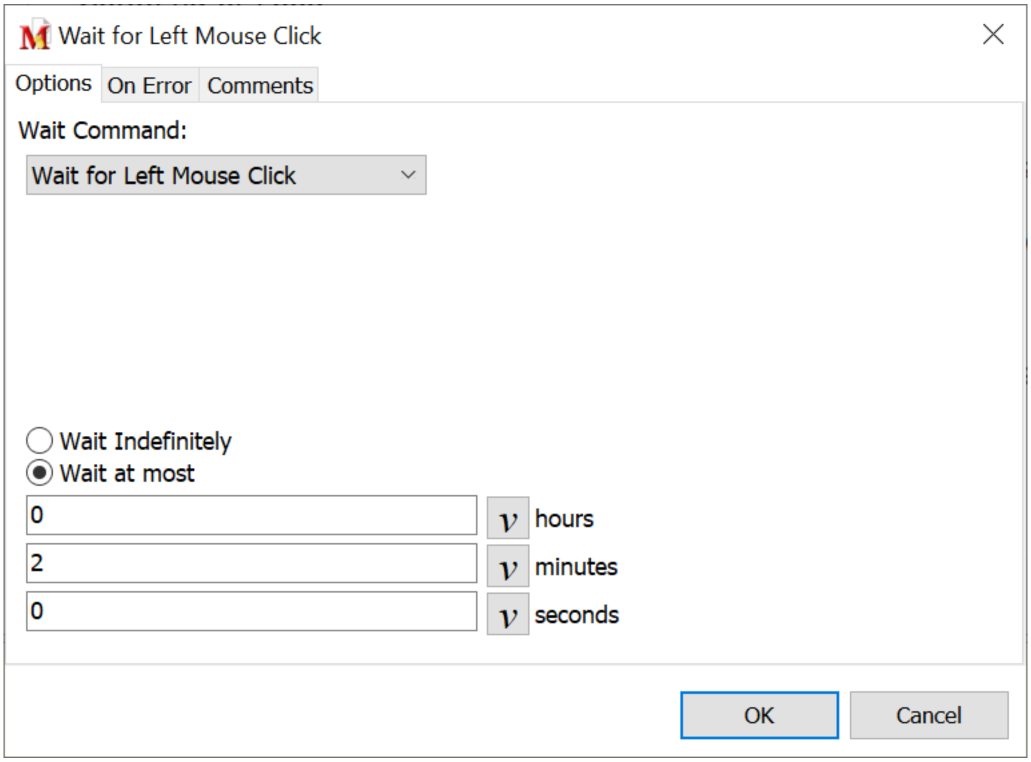
With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Alternative Method for Pausing a Macro

The Pause command is an alternative method for pausing a macro. The pause displays a small window which may include a message for users to follow. Click on the Resume button or use the System Macro - Ctrl Alt Shift Y to resume the macro.

Wait for Mouse Click

These options require a mouse click before the macro resumes playback.



Wait Command: Wait for Left Mouse Click

○ Wait Indefinitely

● Wait at most

0 hours

2 minutes

0 seconds

Script Editor > Expand Timing Category > Wait for Mouse Click

Wait for Left Mouse Click

The macro waits until the left mouse button has been clicked before resuming.

Wait for Middle Mouse Click

The macro waits until the middle mouse button has been clicked before resuming.

Wait for Right Mouse Click

The macro waits until the right mouse button has been clicked before resuming.

Wait Indefinitely

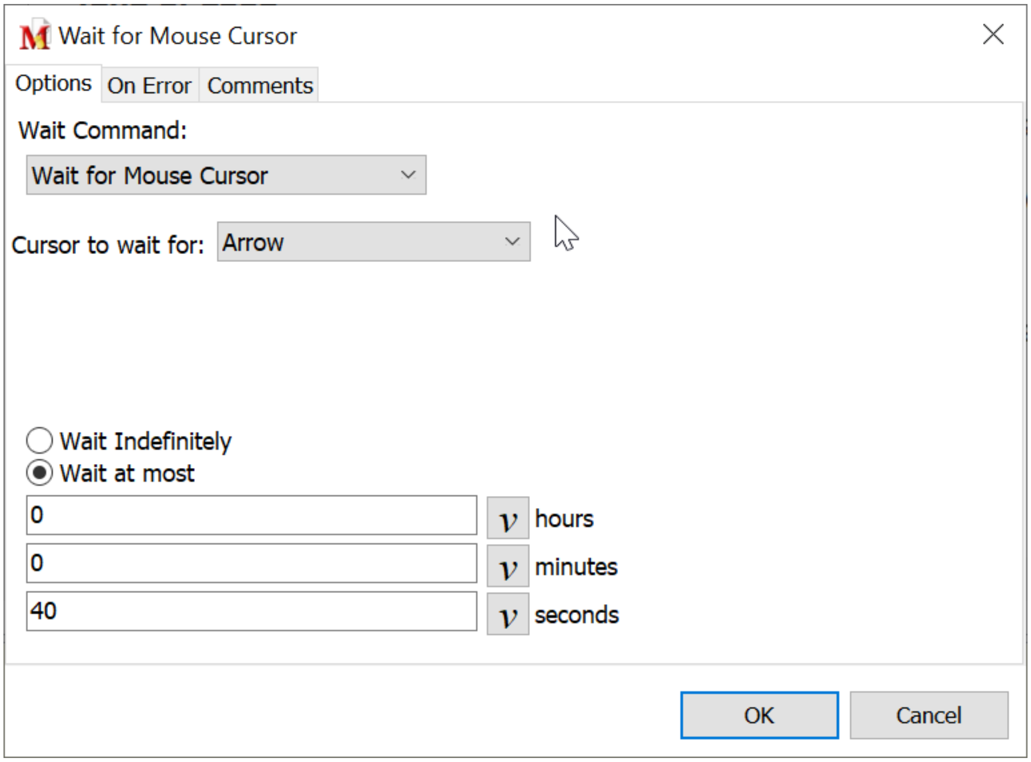
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Mouse Cursor

This option instructs the macro to wait until the mouse cursor type selected appears on the screen. When the specific mouse cursor type appears on the screen, the macro continues.



Wait Command: Wait for Mouse Cursor

Cursor to wait for: Arrow

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
40 seconds

Script Editor > Expand Timing Category > Wait for Mouse Cursor

Cursor to Wait For

Select the mouse cursor from the drop down list. The name as well as a graphical representation of the cursor are displayed to make sure the correct cursor type is selected.

- ✓ **Note:** The mouse cursor must remain in the requested state for at least one half second in order for Macro Express to recognize the cursor type.

Wait Indefinitely

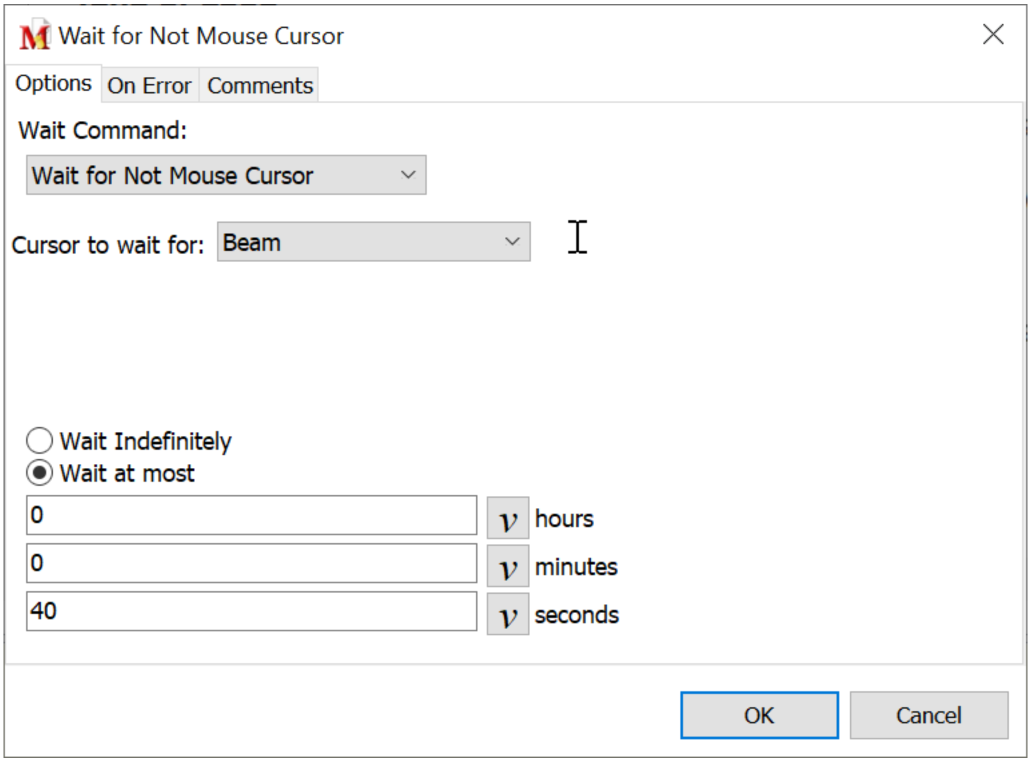
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option selected, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Not Mouse Cursor

This command is essentially the opposite of the Wait for Mouse Cursor option mentioned above. With this command the macro is waiting until the mouse cursor changes to a different type from the one selected. When the cursor changes to something different, the macro continues.



Wait Command: Wait for Not Mouse Cursor

Cursor to wait for: Beam

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
40 seconds

Script Editor > Expand Timing Category > Wait for Not Mouse Cursor

Cursor to Wait for

Select the mouse cursor from the drop down list. The name as well as a graphical representation of the cursor are displayed to ensure the correct cursor type is selected.

Wait Indefinitely

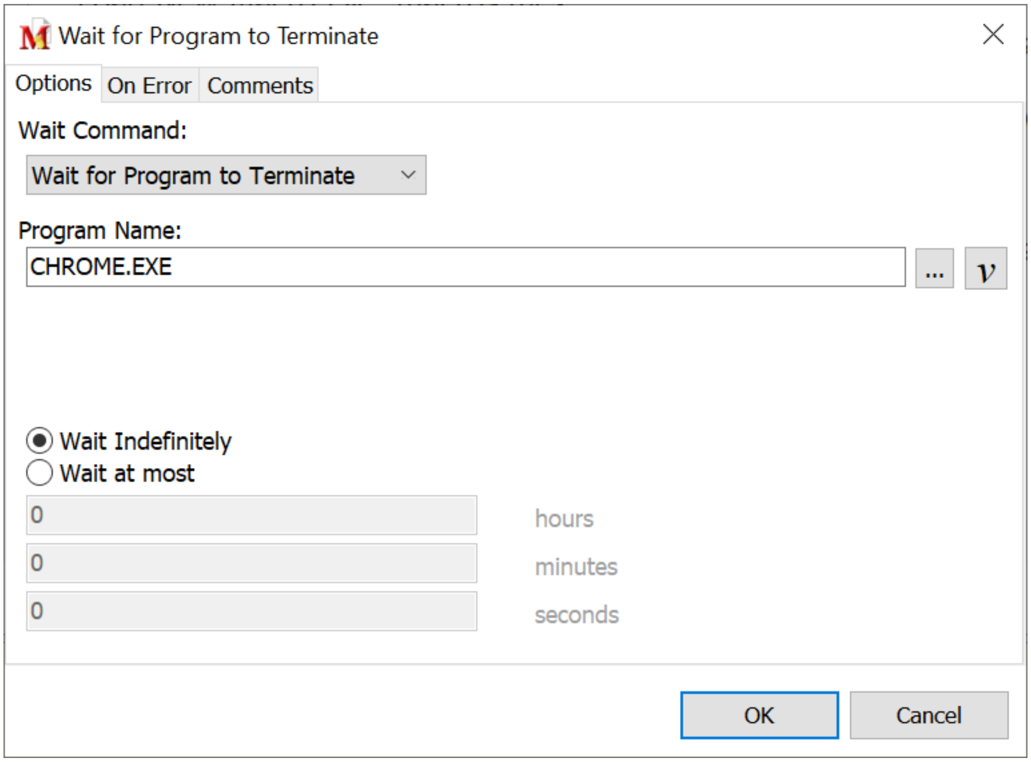
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Program Terminate

The macro waits until the program selected has finished closing, before continuing with the next macro step.



Wait Command: Wait for Program to Terminate

Program Name: CHROME.EXE

● Wait Indefinitely
○ Wait at most

0 hours
0 minutes
0 seconds

Script Editor > Expand Timing Category > Wait for Program Terminate

Program Name

Select the name of the program waiting to terminate. Either insert the name and file path of the program or click on the Browse button to select from the list of programs installed on the computer.

Wait Indefinitely

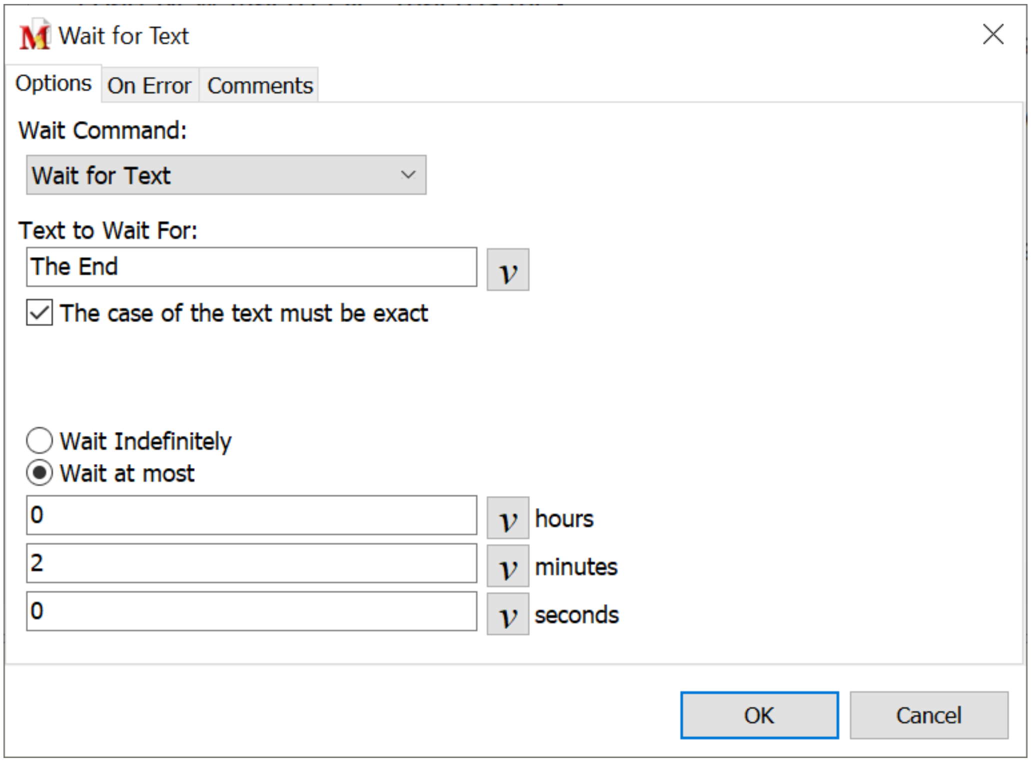
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Text

The macro waits until the specified text is input.



Wait Command: Wait for Text

Text to Wait For: The End

☑ The case of the text must be exact

○ Wait Indefinitely
● Wait at most

0 hours
2 minutes
0 seconds

Script Editor > Expand Timing Category > Wait for Text

Text to Wait For

Insert the text to wait for in this field.

The Case of the Text Must be Exact

Check this box to require that the text matches exactly as originally input, using capital and lower case letters.

Wait Indefinitely

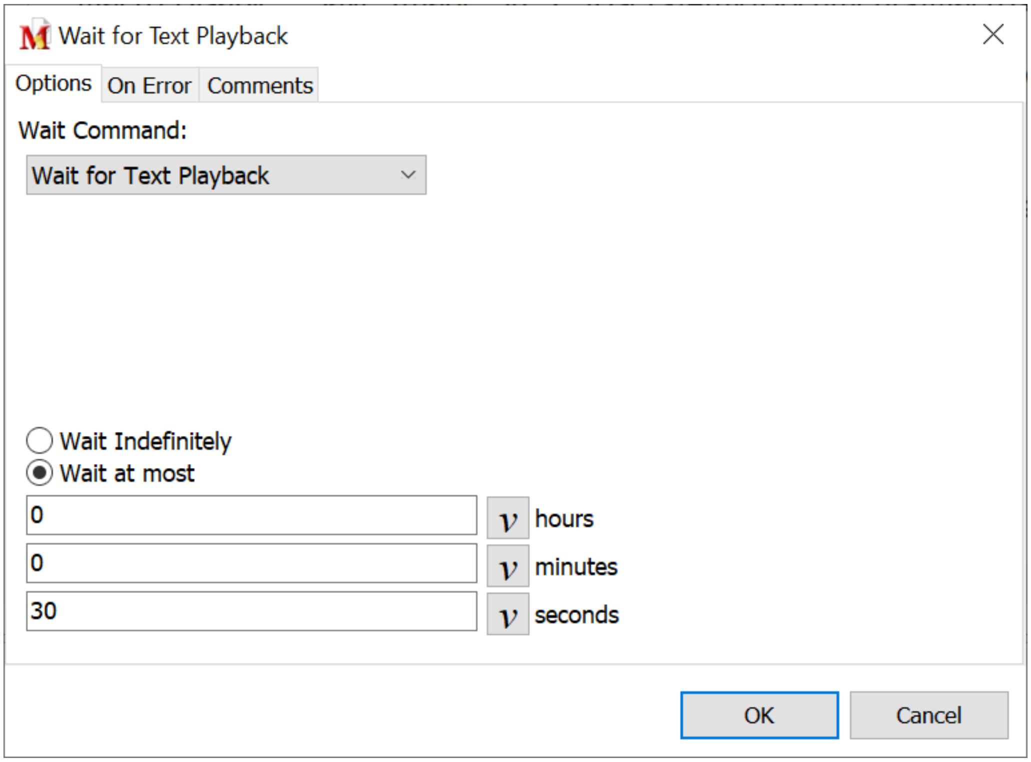
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Text Playback

This option is to be used immediately after a Text command, such as the Text Type function. Wait for Text Playback prevents the macro from continuing until all of the text has finished playing back.



Wait for Text Playback

Wait Command: Wait for Text Playback

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
30 seconds

Script Editor > Expand Timing Category > Wait for Text Playback

Wait Indefinitely

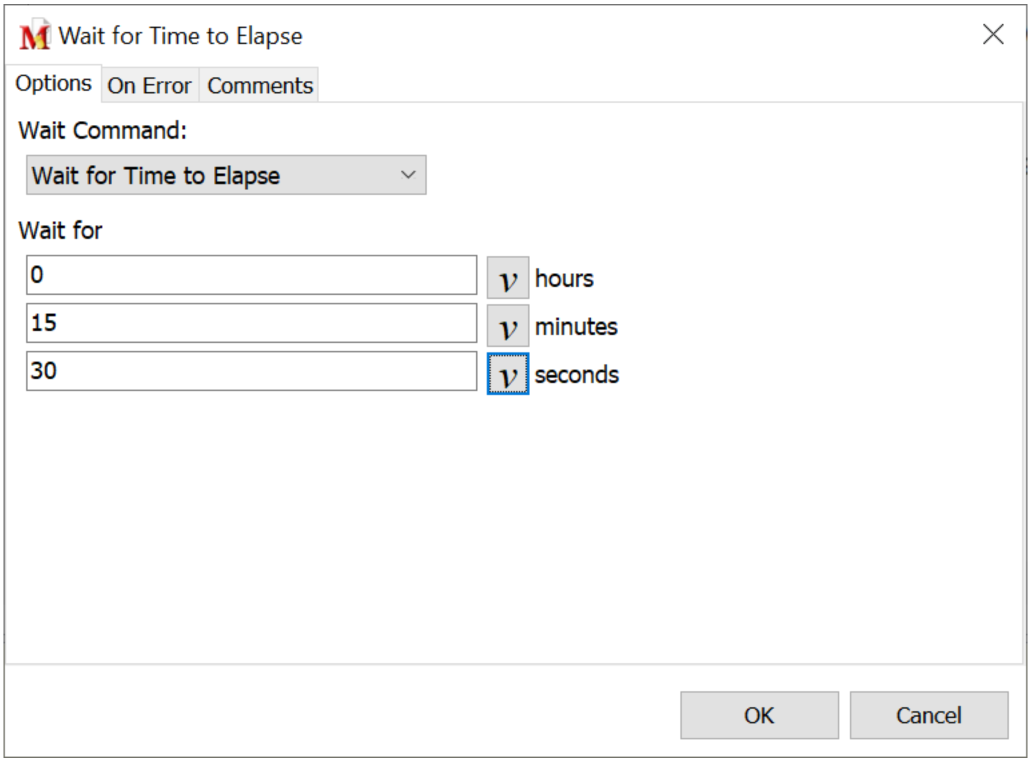
With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Time to Elapse

The Wait for Time to Elapse function instructs the macro to wait a specified number of minutes and seconds before continuing. This command is similar to the Delay command, but with a couple of noticeable differences. First, it allows for a much longer wait period (up to about 7 days). And perhaps more important, this command puts Macro Express into a wait state where virtually no CPU cycles are used, thus allowing CPU intensive programs to continue uninhibited.



Wait Command: Wait for Time to Elapse

Wait for

0 hours

15 minutes

30 seconds

Script Editor > Expand Timing Category > Wait for Time to Elapse

Wait for

Insert the number of hours, minutes, seconds to have the macro wait before continuing.

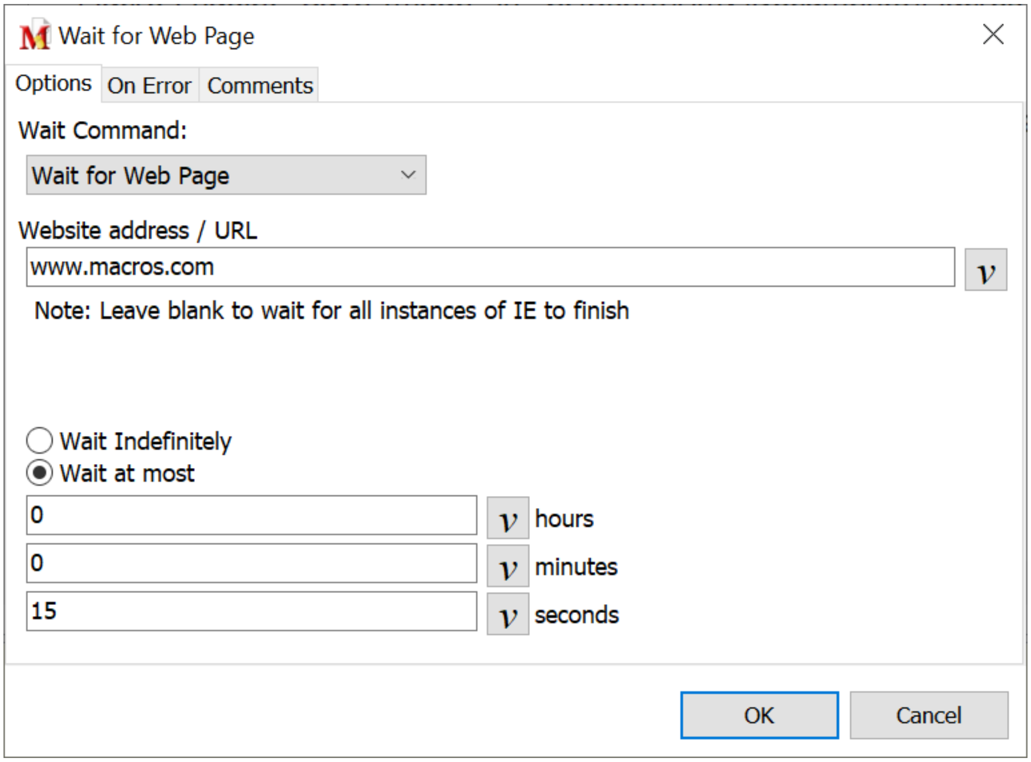
- ✓ **Note:** This command suspends Macro Express for the amount of time specified. During this time the macro cannot be terminated by pressing the Scroll Lock + Pause keys or by right clicking on the "Running Man" icon in the notification area.

- ❖ **Warning:** By putting Macro Express into this wait condition, access to Macro Express functions is not available, including stopping the macro until the specified time has elapsed. Use this command with caution.

Wait for Web Page to Load

The Wait for Web Page to Load command instructs the macro to wait until a web page has finished loading before continuing. This command typically follows a Web Site command. Insert either a partial or complete URL in the URL field of the Wait for Web Page command.

- ✓ **Note:** The Wait for Web Page to Load command only works with Internet Explorer.



Wait for Web Page

Wait Command: Wait for Web Page

Website address / URL: www.macros.com

Note: Leave blank to wait for all instances of IE to finish

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
15 seconds

Script Editor > Expand Timing Category > Wait for Web Page to Load

This command waits for Internet Explorer to report the URL that it last loaded. With some web pages, the URL requested is not the last URL that is loaded. For these web sites there are two choices.

1. Leave the URL field of the Wait for Web Page command empty. Doing this causes the macro to wait until Internet Explorer reports that it is not busy. This works correctly for many web pages.

2. Enter the actual URL that is loaded last into the URL field. To discover what is the last URL loaded, enter some text in the URL field, perhaps "JUNK" and set the "Wait a Maximum of:" field to something short, like 45 seconds or 1 minute. Then run the macro containing the Wait for Web Page command. After 45 seconds or 1 minute a message will pop up that includes the last loaded URL. It may be necessary to repeat this step with a longer wait time

to make sure the web page has actually been loaded. Then, knowing the actual URL loaded last, enter all or part of that URL in the Wait for Web Page command.

Wait Indefinitely

With this option checked, the macro does not resume until the Wait For action specified has occurred.

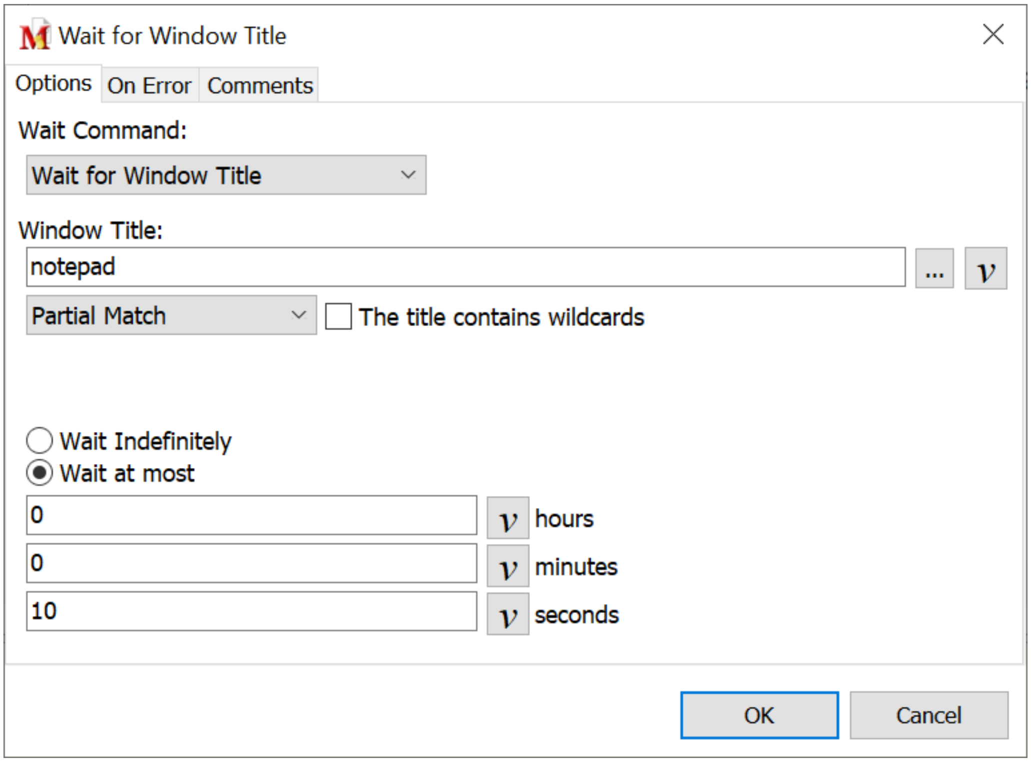
Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Window Title

If writing a macro that changes windows during playback, the macro commands may be ready to execute before the correct window has been activated. This may cause the macro to abort because it didn't find the correct window or subsequent macro commands could play back in the wrong window. To make sure this doesn't happen, insert a Wait for Window Title command instructing the macro to wait until the proper window has focus.

For example, if in a copy and paste routine between two applications, make sure that the macro waits for the second application's window to appear before pasting the copy from the first.



Wait Command: Wait for Window Title

Window Title: notepad

Partial Match □ The title contains wildcards

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
10 seconds

Script Editor > Expand Timing Category > Wait for Window Title

Window Title

Enter the name of the Window (the title in the caption bar of a window). Or click on the Browse button to select from a list of windows currently open on the computer. Choose from Hidden or Visible windows.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro

would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Wait Indefinitely

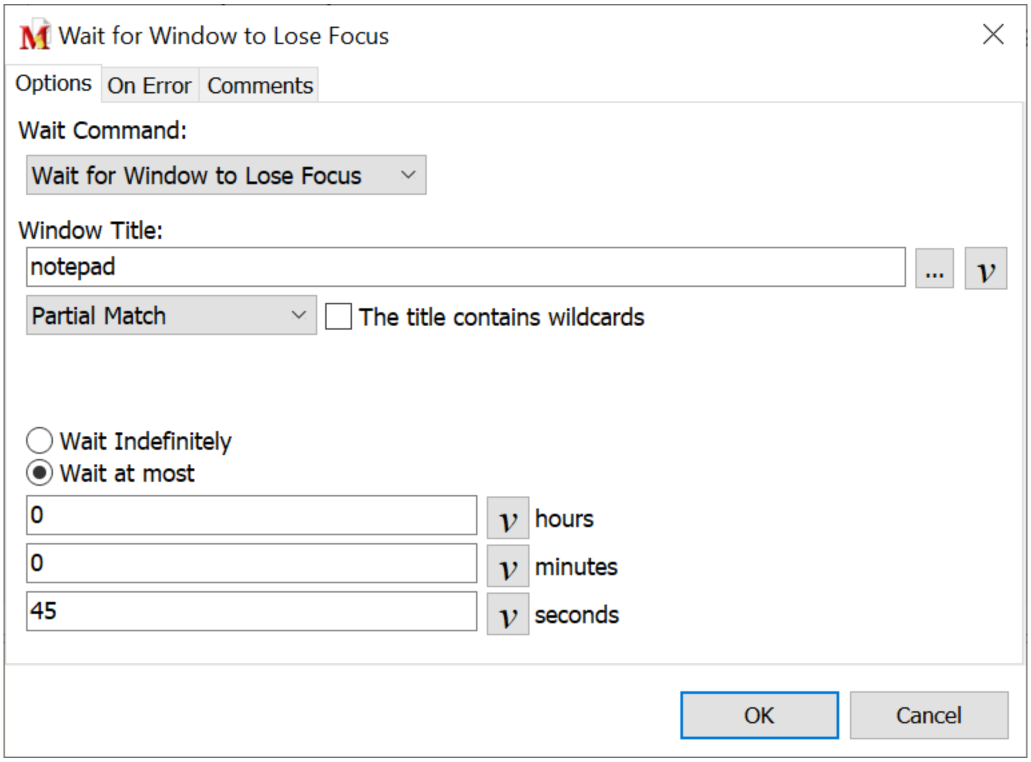
The macro does not resume until the Wait For action specified has occurred.

Wait at most

The macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Wait for Window to Lose Focus

The Wait for Window Title to Lose Focus option instructs the macro to continue only after the Window selected has lost focus or is no longer on top.



Wait Command: Wait for Window to Lose Focus

Window Title: notepad

Partial Match

□ The title contains wildcards

○ Wait Indefinitely
● Wait at most

0 hours
0 minutes
45 seconds

Script Editor > Expand Timing Category > Wait for Window Title to Lose Focus

Window Title

Enter the name of the Window (the title in the caption bar of a window). Or click on the Browse button to select from a list of windows currently open on the computer. Choose from Hidden or Visible windows.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Wait Indefinitely

With this option checked, the macro does not resume until the Wait For action specified has occurred.

Wait at most

With this option checked, the macro waits the number of hours, minutes and seconds specified for the Wait For action to occur. If that action does not take place within the specified time frame, the macro takes the action selected in the On Error tab.

Variables

Variables Overview

About Variables

Variables and conditions add an intelligence element to macros. They also bring an added level of power. Variables provide the capability that software developers find when writing software programs. However, variables in Macro Express are integrated into the commands so that they are easy to implement and use.

A variable is simply something that holds a value. The value can be changed and modified. In other words, it is variable because it can be anything.

Variables allow input of data into the macro from a variety of sources such as from the macro itself, a file, user input, the Windows clipboard, a .INI file, the title of a window and many others. The variables can then be manipulated, processed and tested. Actions can be taken on the results of the test or on the variables themselves. Some specific examples are provided in this section to help cement the idea behind the use and power of variables.

Types of Variables

Macro Express supports several different types of variables. These are Text, Integer, Large Integer, Decimal, Control, Date/Time, Boolean and Handle variables. Variable names may be up to 31 characters in length.

Variable Type	Description
Text String	Holds any type of character, number, word, phrase or paragraph.
Integer	Stores a whole number - values that do not contain fractions (... -2, -1, 0, 1, 2 ...). An integer variable value can range from −2147483648 to 2147483647. When the value exceeds 2147483647, it ‘wraps’. If an integer variable contains 2147483647 and you add 1 to it, the integer variable will contain −2147483648.

Large Integer	Large Integers also store whole numbers. A large integer variable value can range from -9223372036854775808 to 9223372036854775807. When the value exceeds 9223372036854775807, it ‘wraps’. If an integer variable contains 9223372036854775807 and you add 1 to it, the integer variable will contain –9223372036854775808.
Decimal	Holds any type of number, such as 12.432 or -1.3. A decimal variable value can range from 5.0 x 10^ -324 to 1.7 x 10^308 with 15 to 16 significant digits. This first example is a decimal expanded to include 324 leading zeros followed by the number 5, represented as .000(321 more zeros)5. The second range extends to 17 followed by 308 zeros which would look like 1700000000(300 more zeros).0. The numbers are rounded to 15 or 16 significant digits.
Control	Captures information from a window control, such as an edit box, window title, button, etc. This information can then be manipulated to work with other variables in Macro Express.
Date/Time	Contains a date in a specific format so that it can be used for date or time calculations. If only needing to insert the date or time into an application without doing any calculations, then use a Text String variable.
Boolean	Stores either one of two values: true or false.
Handle	Saves the window or control handle information to this variable type.

Examples of using variables

There are virtually an unlimited number of uses for variables in macros. Here are a few examples.

1. Load the mouse cursor position into variables at the beginning of a macro. Restore the mouse location at the end of a macro.

2. Load the position and size of a window into variables at the beginning of a macro. Restore the size and position after manipulating the window.

3. Prompt the user for information and load the information into variables. Save the information to a file and send it via email.

4. Processing a text file or processing an ASCII delimited text file saves values in the files to variables. These values can then be inserted into other applications.

5. Prompt for a text variable. Create boilerplate text that uses the text variable (such as a name) throughout the boilerplate text to customize it. Paste the text into an application.

6. Process a whole set of files by loading in the file names from a text file (i.e. use the process text file command).

7. Manipulate text strings, such as extracting the last 10 characters from a string, finding all characters before the @ symbol, etc.

Defining Variables

There are numerous commands that may be used to define an initial value for a variable. See the Variable Set String, Variable Set Integer and Variable Set Decimal commands, or any number of variable set functions. In addition, variables can be set using the date, time, encrypted text, repeats, processing a text file, processing an ASCII delimited text file macro and practically every other command.

There are also methods available to modify or manipulate the variables once they have been created. See the Variable Modify String, Variable Modify Integer and Variable Modify Decimal commands, or other variable modify functions.

Assign a unique name to each variable or use the variable numbering system from previous versions of Macro Express. The predefined variables are T[1] - T[99] for Text variables, N[1] - N[99] for Integer variables, D[1] - D[99] for Decimal variables and C[1] - C[99] for Control variables.

Create a New Variable

There are three ways to create a variable to be used in a macro.

1. From the Scripting Editor click on the Variables tab. Click on the Add button to open the Variable Properties window and create the variable.

Step by Step Instructions

This example shows how to create an Integer Variable named Insight with an array of 25 elements.

1) From the Script Editor click on the Variables tab to open the Variable Properties window.
2) Enter Insight in the Variable Name field.
3) Click on the drop down menu and select Integer Variable as the Variable Type.
4) Click the check box to create the macro as an array.
5) Enter 25 as the number of elements in the array.
6) Click OK to save and return to the Variables Tab.
7) The new Integer variable, Insight, with 25 elements available is displayed.

2. When editing a macro command click on the Variables button to the right of the edit field. Click on the Add button to open the Variable Properties window as shown below and create the variable.

Step by Step Instructions

This example shows how to create a Decimal Variable named Software from within the Variable Set Decimal command. This variable will not include an array.

1) From the Script Editor expand the Variables section of commands and double click on Variable Set Decimal to open the Variable Set Decimal window.
2) Select the Set Value Now option.
3) Click on the Variables button to the right of Destination Variable to open the Insert Variable window.
4) Click on the Add button to open the Variable Properties window.
5) Enter Software in the Variable Name field.
6) The Variable Type defaults to Decimal Variables.
7) Click OK to save and return to the Insert Variable window.
8) Click OK to save and return to the Variable Set Decimal window.
9) %Software% should now be listed as the Destination Variable.

3. In the macro command window enter the name of the variable in an edit field that accepts variables. After clicking OK to save the information in the command window, a prompt informs that the variable does not exist and asks to create it now. Verify that the Variable Type is correct and create as an array, if necessary.

Step by Step Instructions

This example demonstrates how to create a Text Variable named Solutions, with no array, using the Variable Set String command.

1) From the Script Editor expand the Variables section of commands and double click on Variable Set String to open the Variable Set String window.
2) Select the Set Value Now option.
3) Type Solutions in the Destination Variable field.
4) Enter This is a Test in the Initial Value Edit field.
5) Click OK to save the changes.
6) A dialog box opens informing that The variable "Solutions" is not currently defined. Would you like to create it now?
7) Click Yes to open the Variable Properties window.
8) Solutions should be listed as the variable name and Text Variables selected.
9) Click OK to save the variable and return to the Script Editor.

Variable Name

In the Variable Properties window assign a name to the variable being created. Assign any name to the variable or use the variable assignments defined in previous versions of Macro Express, such as T1 or D95. Macro Express Pro displays them as T[1] or D[95] because they are now part of a variable array. In the sample image above the variable name is sales. Variable names may not exceed 31 characters.

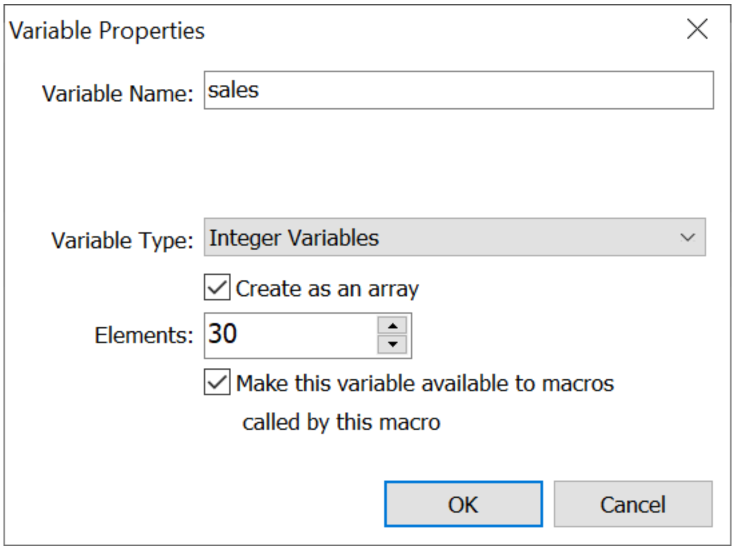
- ✓ **Note:** If assigning original names to variables, rather than using T[1], D[1], etc., then the following characters are not allowed in the variable name: [ ] < > , . - + = \ | / * %

Variable Type

Select from the drop down list the type of variable to create, such as a Text variable, Integer variable, Decimal variable, etc.

Create as an Array

At times it may be appropriate to assign more than one value to a single variable. In such instances, create a variable that contains a series of values. This series is called a variable array. If the variable is named "sales", then the array elements would be designated as sales[1], sales[2] and so on up to the number of elements defined in the array. Array elements start with [1] and not [0]. Each element stores a separate value.



Variable Properties

Variable Name: sales

Variable Type: Integer Variables

☑ Create as an array

Elements: 30

☑ Make this variable available to macros called by this macro

Script Editor > Variables Tab > Add Button
Or Click on any Variables Button next to an Edit Field > Add Button

Select the Create as an array option to set the variable as an array and then choose the number of elements to be available in the array. Variable arrays place brackets around the element number of the variable.

Elements

Select the number of elements needed in the array. In the example above, 30 elements have been selected. This allows the use of up to 30 variables in the array. The variable format looks like the following:

sales[1]
sales[2]
sales[3]
...
sales[30]

In the above example, an array with 30 elements allows assignment of a value to variable sales[1], another value to variable sales[2], etc. - up through sales [30]. There is not a limit to the number of elements that may be assigned to an array variable.

✓ **Note:** The "T", "N", "D" and "C" variables are predefined and always available for use as an array. The variables are assigned as T[1], T[2], T[3], etc. By default, each of these predefined variables are set to an array of 99 variables. There is not a limit to the number of elements in the array and the default arrays may be expanded if desired. Using example #1 above, change the array value. The element of the array is placed in brackets as shown in the Elements example above.

Make this Variable Available to Macros Called by this Macro

Select this option in order to allow the variable value to be passed between macros when using the Macro Run command or when using the Variable Save and Variable Restore commands.

When this option is selected the variables are also referred to as Global variables, because they are available to other macros. When not selected, the variable values are Local and may only be seen in the current macro.

Click OK to save the variable information.

- ✓ **Note:** See the Using Integer Variables and Using Text String Variables tutorials for actual examples of how to create variables and use them in macros.

Variable Scope

Variable Scope describes when the information in a variable is accessible. When created, the stored value is only available for a certain part of a macro. Macro Express Pro allows for variables with either a Local or Global scope.

Local and Variable Scope

When a variable is created, the 'Make this Variable Available to Macros Called by this Macro' option determines whether the variable has a Local or Global scope. Select this option to save the variable with a Global scope. Leave this option unchecked to save the variable with a Local scope.

Local Variable

A Local variable is accessible only inside the macro where it is defined.

For example, suppose you have macros A, B and C, and A is the main macro that calls macros B and C via the Macro Run command. Macro A could have a local variable named %Address%, and macro B could also have a local variable named %Address%. Each of these variables contain separate and distinct values. Changing the variable value in macro B has no effect on the variable in Macro A.

Global Variable

A Global Variable is accessible inside the macro where it is defined and in all other macros called by that macro via the Macro Run command. It is recommended that global variables be defined in the main macro.

For example, suppose you have macros A, B and C and A is the main macro that calls macros B and C via the Macro Run command. After defining a variable named %Address% in macro A both macros B and C can use this variable. Changes made to the value of %Address% in macro B would be available in macros A and B.

Mixing Scope

Separate variables may be defined with the same name and different scopes. Using the example above, a global variable named %Address% may be defined in macro A and a local variable named %Address% defined in macro C. The global %Address% will be used in macros A and B but the local %Address% will be used in macro C.

Macro Threads

Macros are activated in a variety of different ways. Each activation can be thought of as a separate task or thread. This macro thread consists of the main macro and any macros called from that macro by the Macro Run command. If the same macro runs more than once as a result of two separate activations, each activation is running on its own thread.

Variables are not Global between macro threads. This applies even if the same macro is called. For example, suppose you have macros A, B, C and Z. Both macros A and Z call macros B and C via the Macro Run command and both contain the global variable %Address%. The variable %Address% in macro A is separate and distinct from the variable %Address% in macro Z. If macro A sets %Address% to ‘PO Box 106’ and macro Z sets %Address% to ‘120 South Main’ then %Address% in macros B or C would contain ‘PO Box 106’ if called from macro A and ‘120 South Main’ if called from macro Z.

Making information available to different macro threads

Sometimes information needs to be shared between different macro threads. Some examples might include paths to common files, user names, department names, company names or delay values. Each individual or organization will have different needs. There are several effective ways to share values between macro threads.

Environment Variables

Use the ‘Variable Modify String: Save to Environment Variable’ command to save a variable to the environment variables area of Windows. Later, use the ‘Variable Set String: Set from an Environment Variable’ command to read an environment variable into a text variable.

Values saved in the environment variable will remain until Macro Express Pro is terminated and are also available to any programs called by Macro Express Pro after the variable has been saved.

Some older versions of Windows may impose a size limit on an individual environment variable or the size of the environment block.

Environment variables are a good place to store text variables. Numeric information may also be stored, but it must be converted to and from a text string. Avoid trying to store control (non-printable) characters in the environment variables.

Registry

The ‘Write Registry Value’ and ‘Read Registry Value’ commands may be used to save and restore variables. Saving information in the Registry this way is safe as long as the value is not written to a registry key or value that is used by another program or by the operating system. Also, some users can write to certain areas of the registry and others cannot.

There should not be any problems writing registry values to a registry key similar to this:

HKEY_CURRENT_USER\Software\<Your organization or company name>\Macros

All variable types except control variables may be saved to and restored from the Registry.

INI Files

Text variables can also be saved in INI files using the 'Variable Modify String: Save to INI File' and 'Variable Set String: Set Value from INI File' commands.

Values saved in INI files remain until the INI file is modified or deleted. It is possible to store numeric information but it will need to be converted to and from a text string. Avoid trying to store control (non-printable) characters in INI files.

Text Files

Save anything that can be stored in a text variable to a file using the 'Variable Modify String: Save to Text File' command. Later, read this information using the 'Variable Set String: Set Value from File' command.

Values saved in text files remain until the file is modified or deleted. It is possible to store numeric information but it will need to be converted to and from a text string. Unlike storing information in environment variables or INI files, control (non printable) characters may be stored in a text file.

Variable Save / Variable Restore

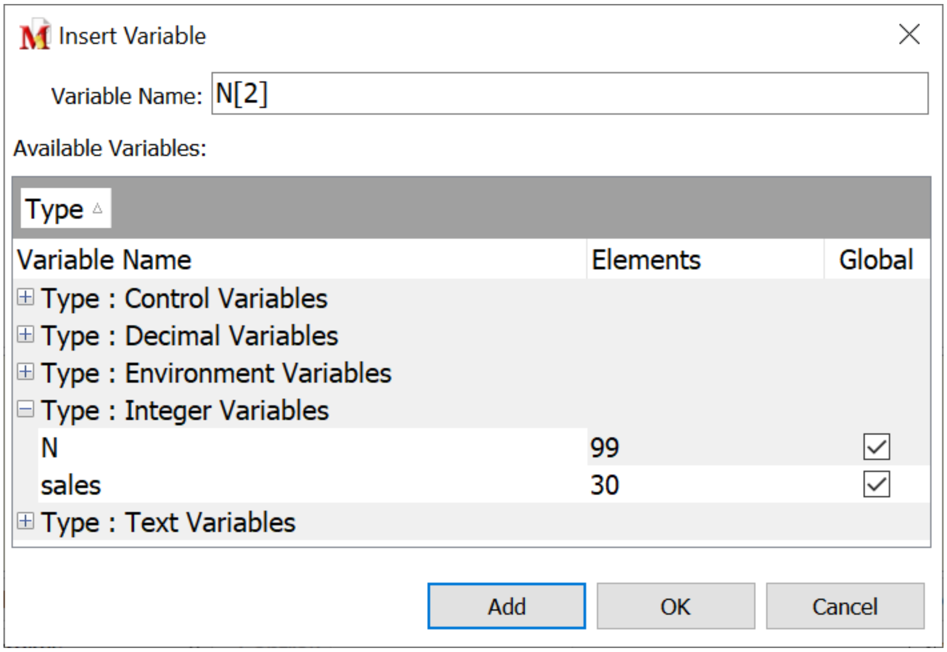
Global variables may be passed from one macro to the next using the Variable Save and Variable Restore commands.

Playback of Variables

Variables can be placed in the input fields of practically all macro commands. There are a couple of methods that may be used for inserting existing variables into command fields.

1. Type the existing variable name directly into the edit field. To do so, place the name of the variable surrounded by percent signs (%). This would look like %T[1]% for the Text variable array T[1] or %results% for a variable previously defined as results.

2. Click on the Variable button to open the Insert Variable window. Expand the variable type needed, select an existing variable and click OK. If using an array variable such as N in the example below, N[ ] will appear in the variable name field. Enter the number of the array to be used such as 2 in this example and save. Array elements start with [1] and not [0].



Insert Variable

Variable Name: N[2]

Available Variables:

Type ▵

Variable Name	Elements	Global
⊞ Type : Control Variables		
⊞ Type : Decimal Variables		
⊞ Type : Environment Variables		
⊟ Type : Integer Variables		
N	99	☑
sales	30	☑
⊞ Type : Text Variables		

- ✓ **Note:** In many of the commands, only one variable type may be available for selection in the Insert Variable window. For example, if using the Variable Set String command only Text Variables are available for use.

Before selecting a variable, it must first be defined, as discussed in the Defining Variables topic.

Example 1

To launch a program, enter %T[1]% in the Program/Path Name field of the Program Launch command. When the macro is executed, it attempts to run the program that is defined by T[1]. Assign a value (file name and path) to T[1] previously in the macro.

Example 2

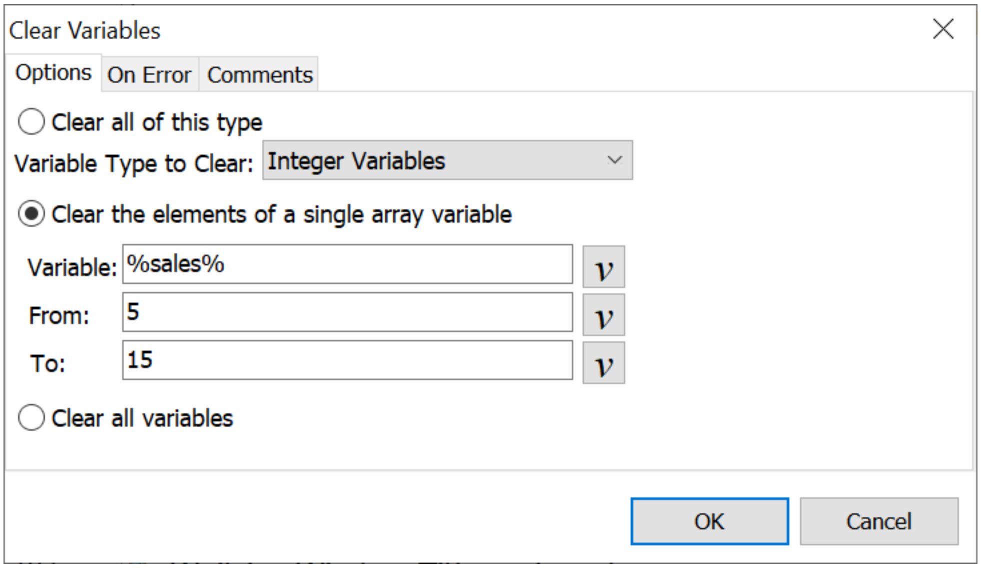
A macro may include several Delay commands each for the same amount of time. Instead of inserting 2 seconds as the delay time, insert %delay%. In this case, the macro delays the amount of time previously defined in the variable named "delay". If a need arises to increase or decrease the delay time, just edit the command that defines the value of %delay% instead of changing each Delay command individually.

There are many ways to use variables. Besides placing variables in commands as described above, there are some functions that accept and use variables directly as part of the command. These include window resizing and repositioning commands, mouse movement commands, repeat commands, processing text or ASCII delimited text files and many others.

Note: See the Using Integer Variables and Using Text String Variables tutorials for actual examples of how to create variables and use them in macros.

Clear Variables

Use this option to clear the value of a variable during macro playback. Select to clear the values of all variables of a specific type, a range of elements in a single array variable or every variable used in the macro.



○ Clear all of this type

Variable Type to Clear: Integer Variables

● Clear the elements of a single array variable

Variable: %sales%

From: 5

To: 15

○ Clear all variables

Script Editor > Expand Variables Category > Clear Variables

Clear all of this type

Select this option to clear all of the variable values of the type selected.

Variable Type to Clear

Select the type of variable to clear. There are eight different variable types to choose from. These include Text, Integer, Large Integers, Decimal, Date/Time, Boolean, Control and Handle variables.

Clear the elements of a single array variable

If using an array variable in a macro, this option clears a range of array elements for the variable selected.

Variable

Select the variable array to be cleared of values. Either enter the variable name such as %Sales%, %D%, etc. or click on the Variables button to find the variable from the list of available choices.

From

Select the first element in the array to be cleared.

To

Select the last element in the array to be cleared.

All variables between the first (From) element chosen and the last (To) element are cleared of their values. For example, consider an array variable with 10 elements and the numbers 2 and 5 are placed in the From and To fields. When the macro is run, only the values of elements 2 through 5 are cleared. The values of the other variables in the array, 1 and 6 - 10 remain unchanged and are not cleared.

Array elements start with [1] and not [0].

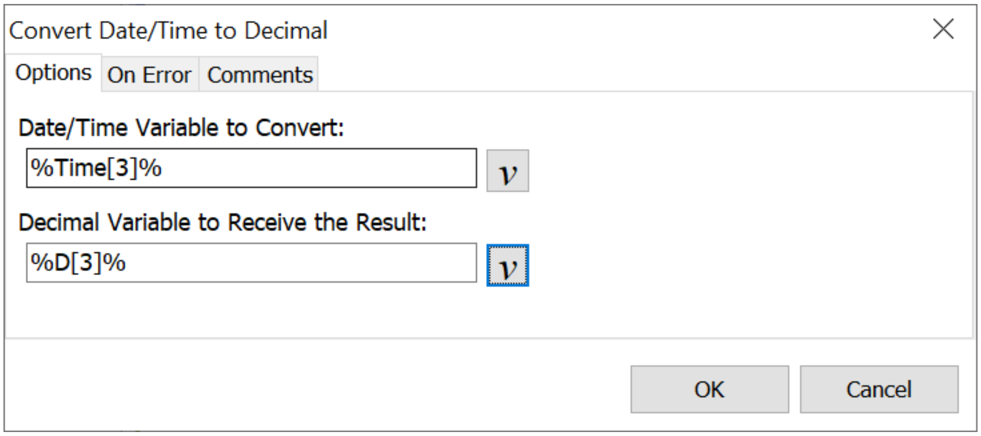
Clear all variables

Select this option to clear the values of all variables, regardless of variable type.

✓ **Note:** The Clear Variables command does not affect variables saved through the Variable Save command.

Convert Date/Time to Decimal Variable

This option converts a Date/Time variable to a decimal value representing the date/time. The value is then saved to a Decimal variable. Create a Date/Time variable through the Variable tab in the Script Editor or through the Date/Time command.



Date/Time Variable to Convert: %Time[3]%

Decimal Variable to Receive the Result: %D[3]%

Script Editor > Expand Variables Category > Convert Date/Time to Decimal

Date/Time Variable to Convert

Select the Date/Time variable to convert to a Decimal variable.

Decimal Variable to Receive the Result

Select the Decimal variable to store the value of the converted Date/Time variable.

Example:

Below is a sample date and time saved to a Date/Time variable:

8/8/2017 11:37:40 AM

The Date/Time variable value above converts to the decimal value shown below:

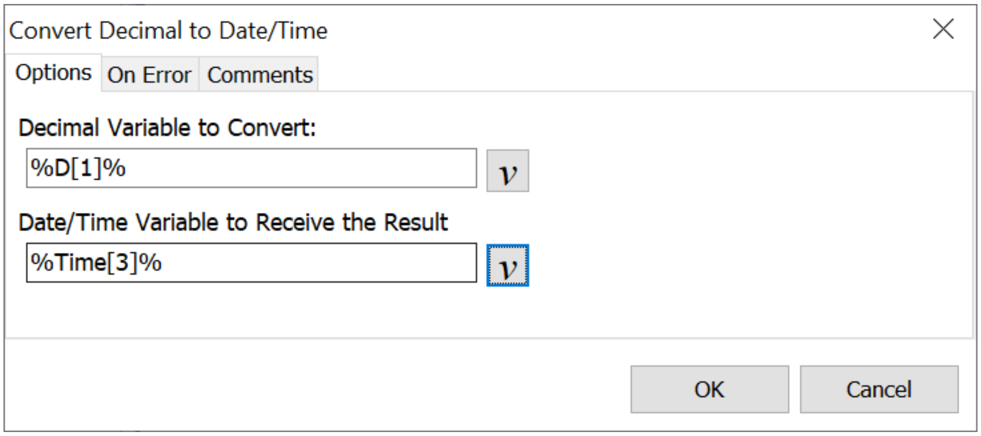
42955.4844978704

The first 5 digits represent the date. The decimal section, to the right of the period, represents the time (or fraction of a day).

Convert Decimal to Date/Time

This option converts a Decimal variable that represents the date/time into a Date/Time variable. A Date/Time variable name may be assigned through the Variable tab in the Script Editor or through the Date/Time command.

- ✓ **Note:** The Decimal value saved to the variable must be greater than or equal to zero. An invalid date warning will appear during macro playback if the decimal value is less than zero.



Decimal Variable to Convert: %D[1]%

Date/Time Variable to Receive the Result: %Time[3]%

Script Editor > Expand Variables Category > Convert Decimal to Date/Time

Decimal Variable to Convert

Select the Decimal variable to convert to a Date/Time variable.

Date/Time Variable to Receive the Result

Select the Date/Time variable to store the value of the converted Decimal variable.

Example:

Below is a sample decimal value. The first 5 digits represent the date. The decimal section, to the right of the period, represents the time (or fraction of a day).

42955.4844978704

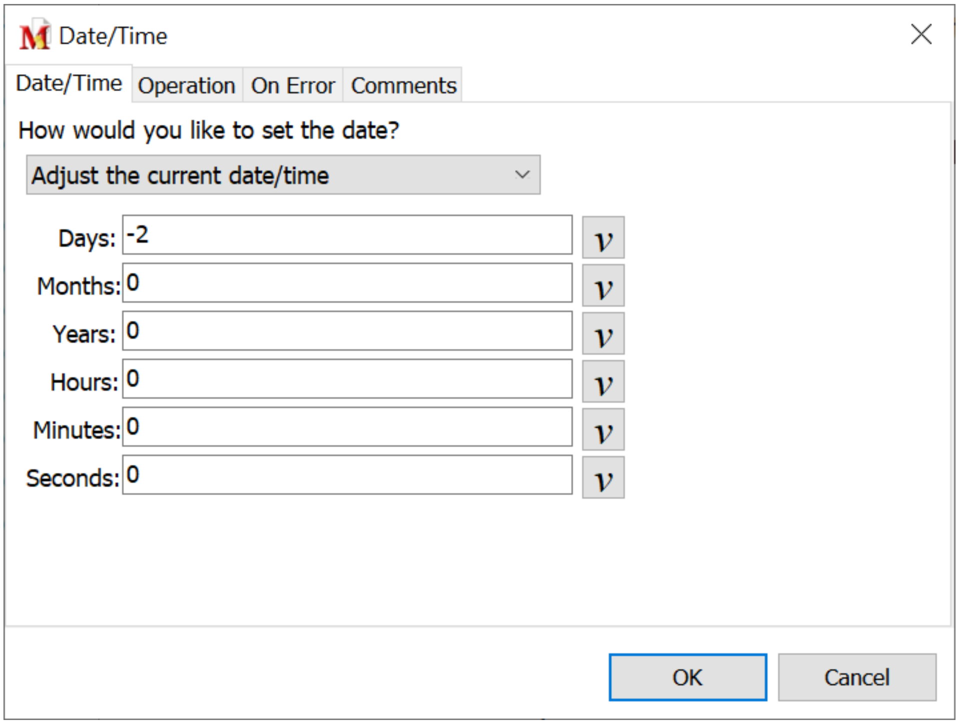
Below is the value saved to a Date/Time variable after the decimal conversion:

8/8/2017 11:37:40 AM

Date/Time

Build date and time formats with this command. Many of these formats are based on the definitions in the Regional Settings of the Control Panel. The month, day, etc. display in the language defined in the Regional Settings. Either save the date and time to a variable or have the macro insert the information into another application.

Date/Time Tab



How would you like to set the date?

Adjust the current date/time

Days: -2

Months: 0

Years: 0

Hours: 0

Minutes: 0

Seconds: 0

Script Editor > Expand Variables Category > Date/Time

How Would You Like to Set the Date?

Select one of the four options from the drop down menu to set the date and/or time.

Set it to the Current Date/Time

This option uses the date and time from your system clock at macro run time.

Use a Specific Date/Time

Use this option to have a specific date or time played back in a macro. Set the value to a specific date or time, either in the past or future.

Adjust the Current Date/Time

Adjust the date or time by Days, Months, Years, Hours, Minutes or Seconds into the future or past. The calculations are based on the computer's current system time and date. Insert negative numbers in the fields to adjust the date or time to the past.

Prompt the User

This option opens a small dialogue window during macro playback requesting the user select a date and or time. The date and/or time is displayed in the format selected.

Prompt

Enter any instructions for the macro user in the Prompt field. The instructions display in the window that opens during macro playback. The current date is displayed in the edit field. The user can either overwrite any of the date information or use the attached calendar to select a date.

If a time prompt is selected, the current time is displayed in the edit field. Overwrite the time by highlighting the hours, minutes or seconds and editing the current entry or use the up and down arrows on the keyboard to change the values.

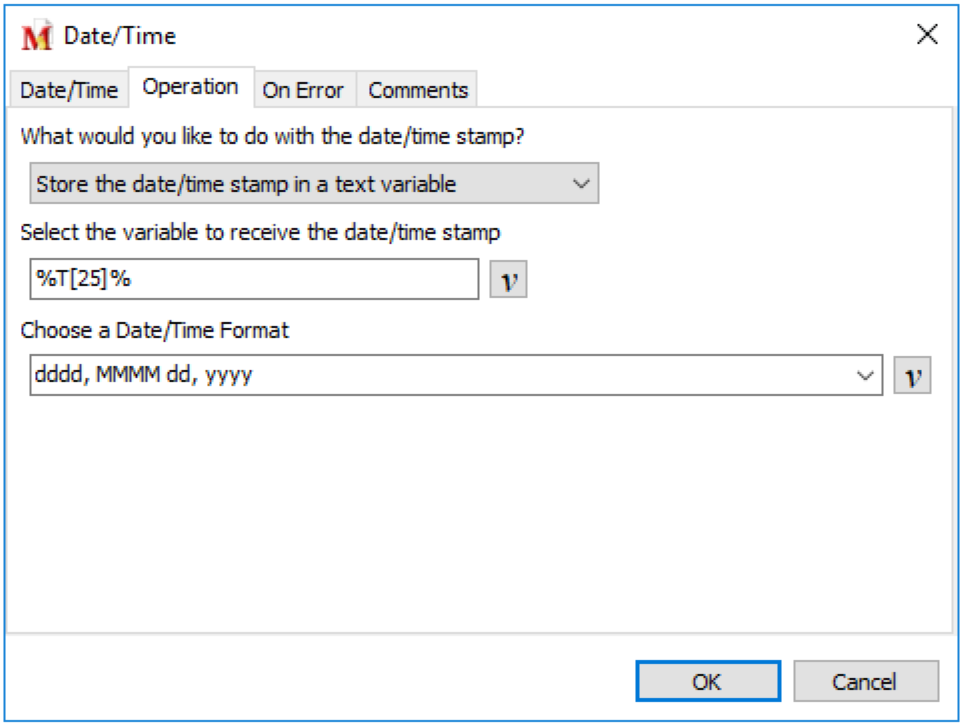
Set Manually

The Set Manually button opens a window to allow positioning of the Prompt window - to set where it will appear on the screen during macro playback. See the Position Window topic for more details.

Prompt for Date/Prompt for Time

Select to have the macro prompt the user for a date, a time or both.

Operation Tab



What would you like to do with the date/time stamp?

Store the date/time stamp in a text variable

Select the variable to receive the date/time stamp

%T[25]%

Choose a Date/Time Format

dddd, MMMM dd, yyyy

What would you like to do with the date/time stamp?

Select from the drop down list what to do with the date and time stamp.

Type the Date/Time Stamp Out

This option types the text (date and/or time) in the format selected into the application that is open when the macro runs.

Store the Date/Time Stamp in a Text Variable

This places the date and/or time into a text string variable. Select or create the text string variable to store the date/time information.

Store the Date/Time in a Date/Time Variable

Select this option to save the date and/or time to a Date/Time variable. Select or create the date/time variable to store the date/time information.

Select the variable to receive the date/time stamp

Choose the variable to store the date/time information. This option is available if either the Store the Date/Time Stamp in a Text Variable or Store the Date/Time in a Date/Time Variable is selected.

Choose a Date/Time Format

Select the date and or time format from the drop down list of pre-defined formats.

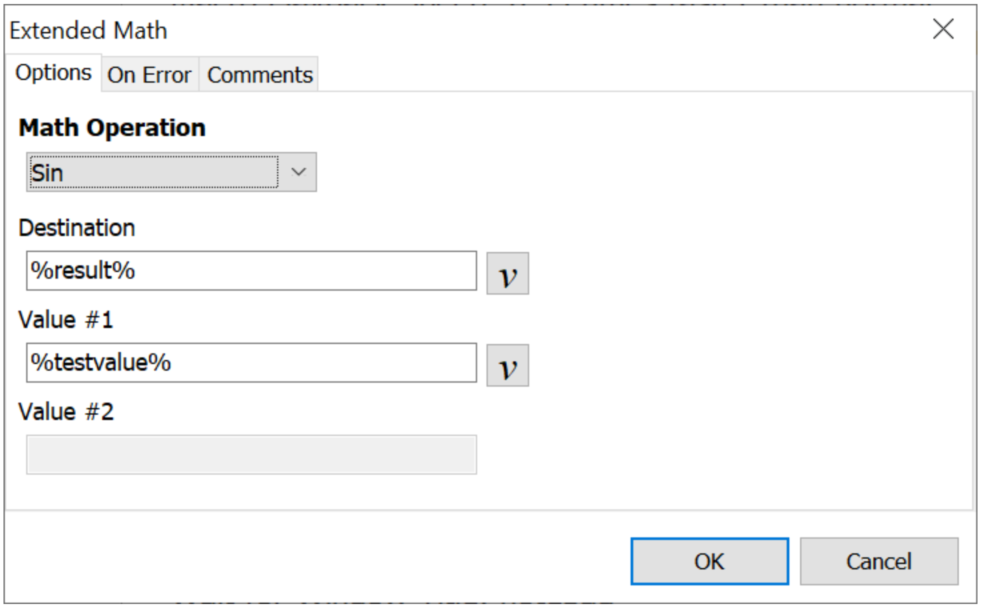
Or create a new format by typing in the date or time characters into this field. This option is not available if the Store the Date/Time in a Date/Time Variable method is chosen.

Modifier	Description
c	Displays the date using the Short Date Format defined in the Control Panel - Regional Settings, followed by the time using the Long Time format defined in the Control Panel - Regional Settings. The time is not displayed if the fractional part of the Date Time value is zero.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat) using the Short Day Name format defined in the Control Panel - Regional Settings.
dddd	Displays the day as a full name (Sunday-Saturday) using the Long Day Name format defined in the Control Panel - Regional Settings.
ddddd	Displays the date using the Short Date format defined in the Control Panel - Regional Settings.
dddddd	Displays the date using the Long Date format defined in the Control Panel - Regional Settings. m Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec) using the Short Month Names format defined in the Control Panel - Regional Settings.
mmmm	Displays the month as a full name (January-December) using the Long Month Names format defined in the Control Panel - Regional Settings.

yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).
h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (00-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
z	Displays the millisecond without a leading zero (0-999).
zzz	Displays the millisecond with a leading zero (000-999).
t	Displays the time using the Short Time format defined in the Control Panel - Regional Settings.
tt	Displays the time using the Long Time format defined in the Control Panel - Regional Settings.
am/pm	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
ampm	Uses the 12-hour clock for the preceding h or hh specifier, and displays AM symbol defined in the Control Panel - Regional Settings for any hour before noon, and the contents of the PM setting defined in the Control Panel - Regional Settings for any hour after noon.
/	Displays the date separator character defined in the Control Panel - Regional Settings.
:	Displays the time separator character defined in the Control Panel - Regional Settings.
'xx'/"xx"	Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

Extended Math

The Extended Math command performs additional math or trigonometric functions beyond the basic addition, subtraction, multiplication and division found in the Variable Modify Integer or Variable Modify Decimal commands.



Math Operation
Sin

Destination
%result%

Value #1
%testvalue%

Value #2

Script Editor > Expand Variables Category > Extended Math

Math Operation

Select from the ten math or trigonometric operations available.

Sin - Sine

Cos - Cosine

Tan - Tangent

ArcSin - ArcSine

ArcCos - ArcCosine

ArcTan - ArcTangent

Abs - Absolute number

Power - Multiply a number by the nth power, such as 2 to the 10th power.

Min - Selects the smaller of two values.

Max - Selects the larger of two values.

- ✓ **Note:** The trigonometric functions require that the values be in radians - (Angle x Pi)/180 = Radians. In most cases, the resulting values of the trigonometric operations will be decimal values.

  ArcSin and ArcCos values must be input as integers between 1 and (-1). Otherwise an error message will result.

Destination Value

Select the variable to store the value of the math operation performed. This variable stores the result of the math operation.

Value #1

Set the value or select a variable to be used in the math operation.

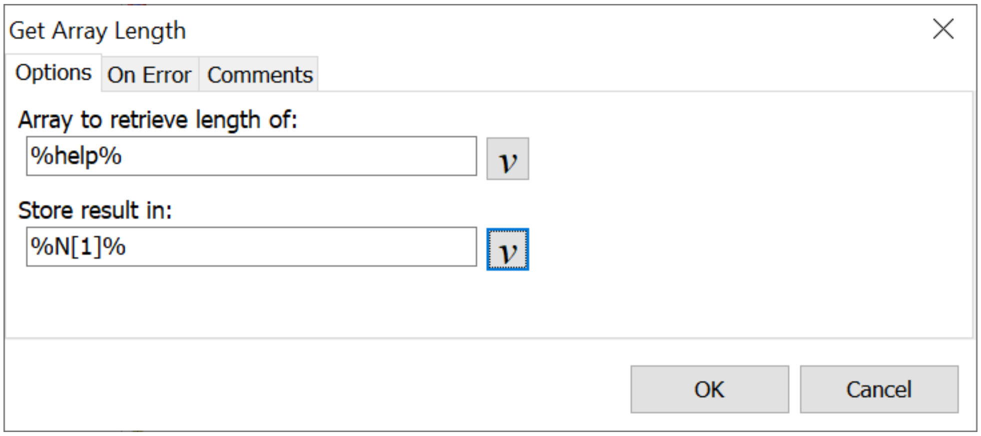
Value #2

Set the value or select a variable to be used in the math operation. This is only available for a few of the above operations, those that require two values.

Get Array Length

At times it may be appropriate to assign more than one value to a single variable. In such instances, create a variable that contains a series of values. This series is called a variable array. If the variable is named "sales", then the array elements would be designated as sales[1], sales[2] and so on up to the number of elements defined in the array. Array elements start with [1] and not [0]. Each element stores a separate value.

Get Array Length returns the maximum number of elements defined for an array variable. This command is useful when used in conjunction with a Repeat command. Use it to change the number of array elements without modifying the macro script.



Array to retrieve length of: %help%

Store result in: %N[1]%

Script Editor > Expand Variables Category > Get Array Length

Array to retrieve length of:

Select the array variable from which to retrieve the number of elements.

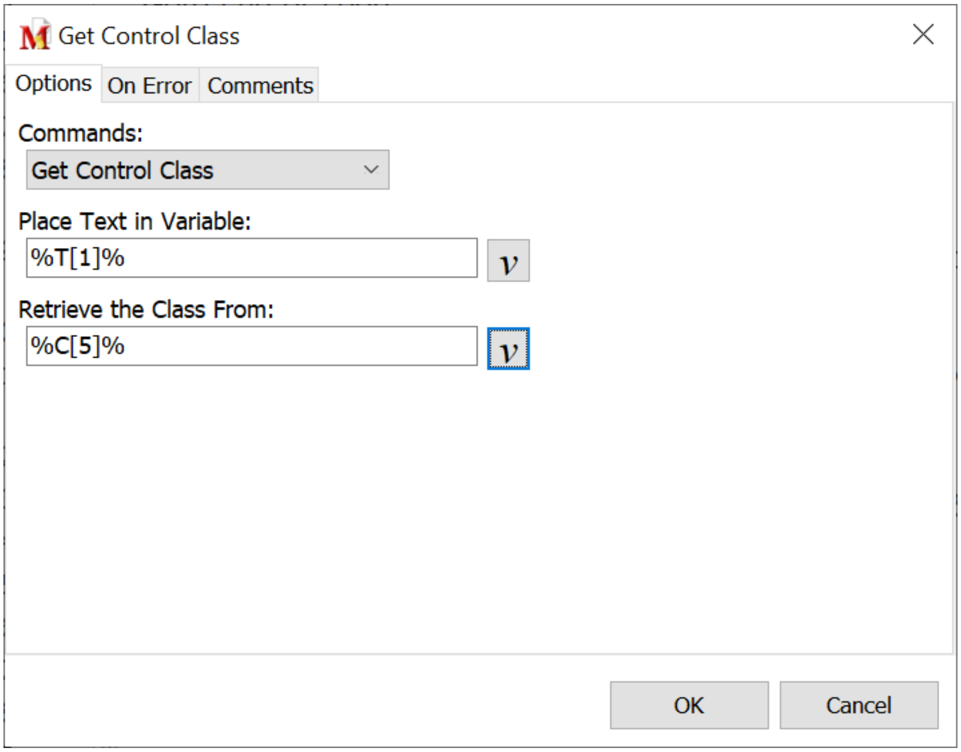
Store result in:

The maximum length of the array is saved to the variable selected. The variable selected must be an Integer variable. The Integer variable stores the number of elements that were defined for the variable - in this case the variable named "sales".

See the Defining Variables topic for more information about arrays.

Get Control Class

This command places the class name of the control into a text string variable. In the example below, when selecting a control, a class name for the control is listed. For example if capturing a button as a control variable, such as %control% in the example below, the class name very likely would be "Button". The class name of "Button" would be saved to text string variable %T[1]% based on the image below.



Commands: Get Control Class

Place Text in Variable: %T[1]%

Retrieve the Class From: %C[5]%

Script Editor > Expand Variables Category > Get Control Class

Place Text in Variable

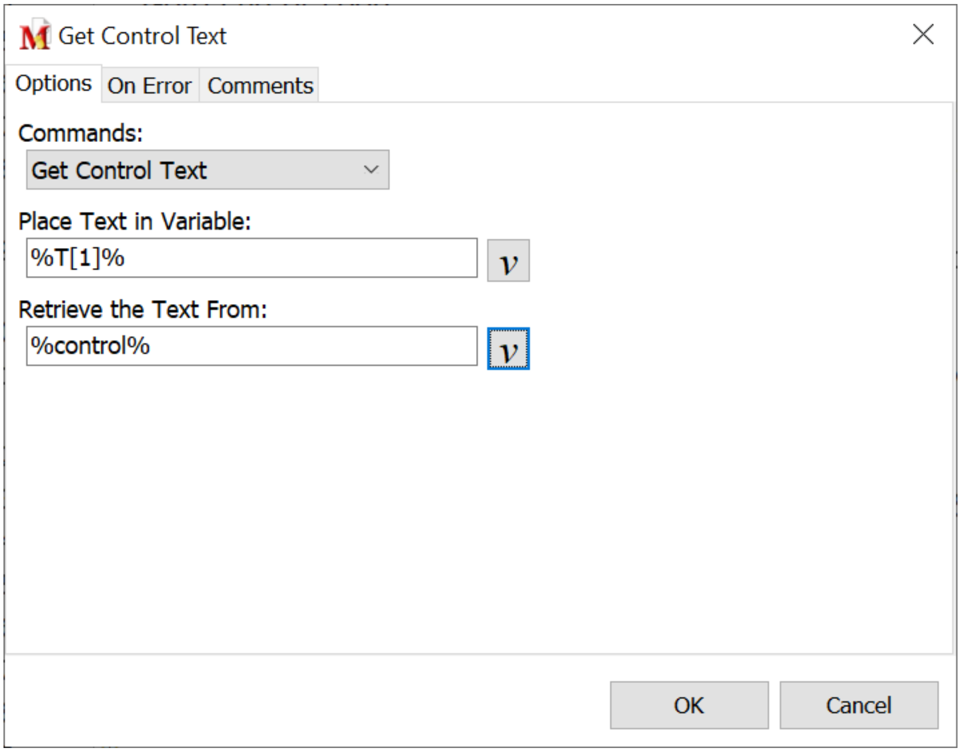
Select the text string variable to receive the control class information.

Retrieve the Class From

Select the control variable that contains the class information to be passed to a string variable.

Get Control Text

This command looks at the text saved to a control variable and saves it to a text string variable. This is helpful for determining the text currently in an edit box and allows for further manipulation of the text string.



Commands: Get Control Text

Place Text in Variable: %T[1]%

Retrieve the Text From: %control%

Script Editor > Expand Variables Category > Get Control Text

Place Text in Variable

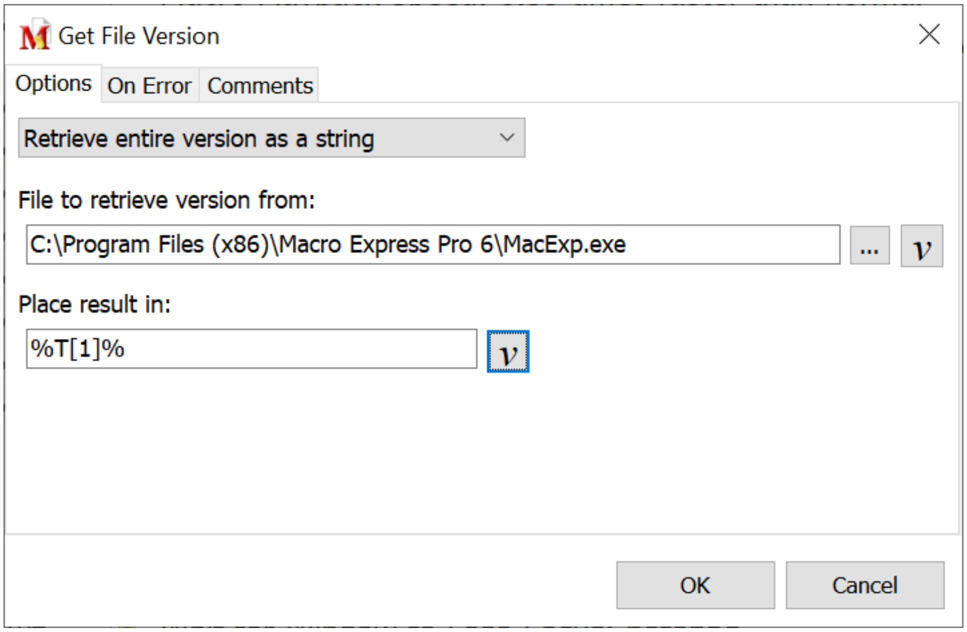
Select the text string variable to receive the control text information.

Retrieve the Text From

Select the control variable that contains the control text information to be passed to a string variable.

Get File Version

This command obtains the file version number of the file selected and saves the information to a text string variable or integer variable. This can be useful in comparing two program files to determine which is the later file. Use this command primarily with executable (.exe) files, .dll files, and similar, as these files normally contain version number information.



Retrieve entire version as a string

File to retrieve version from: C:\Program Files (x86)\Macro Express Pro 6\MacExp.exe

Place result in: %T[1]%

Script Editor > Expand Variables Category > Get File Version

Retrieve Entire Version as a String

This option saves the entire version number to a text string variable. The resulting version number would look something like 6.0.1.1.

File to Retrieve Version From

Press the browse button to locate the file to be examined or type in the full file path and name into the edit box.

Place Result In

Select or create the variable to store the file version number information.

Retrieve Individual Numbers

Use this option to save the components of the version number needed. Select from the major, minor, release or build number.

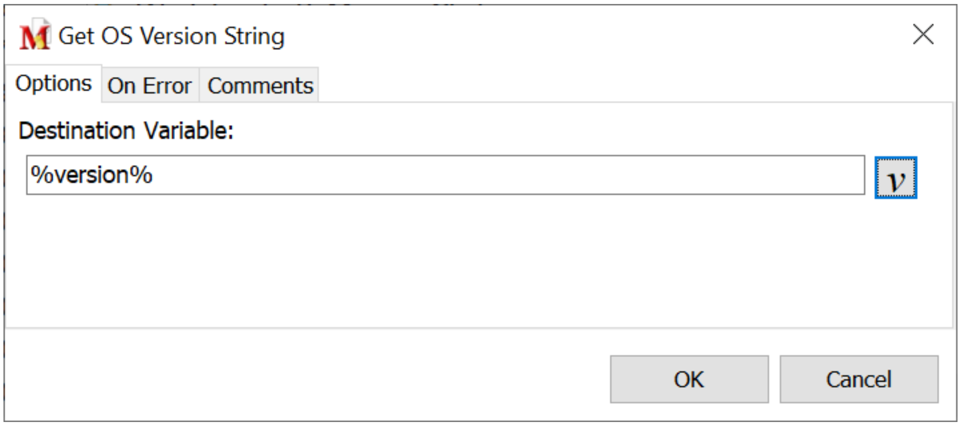
File to Retrieve Version From

Press the Browse button to locate the file to be examined or type in the full file path and name into the edit box.

Select the desired components of the file version number and save variables.

Get OS Version String

This command obtains the operating system information and saves the data to a variable.



Destination Variable: %version%

Script Editor > Expand Variables Category > Get OS Version

Destination Variable

Select the variable to store the operating system information.

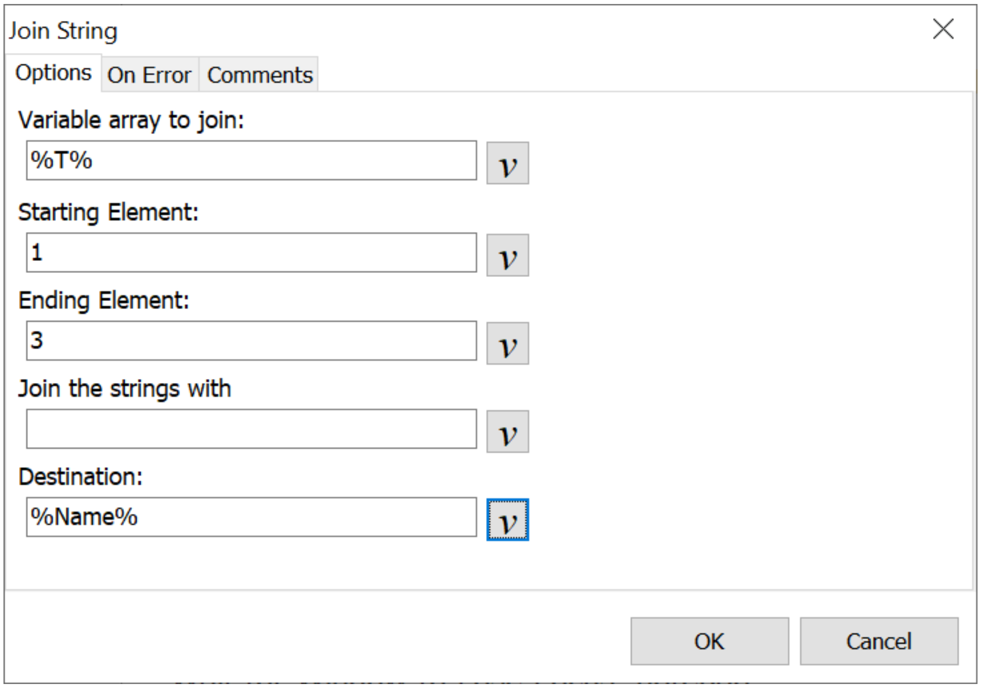
The information saved to a variable will look something like the following:

Windows 10 10.0.18363

Join String

At times it may be appropriate to assign more than one value to a single variable. In such instances, create a variable that contains a series of values. This series is called a variable array. If the variable is named "sales", then the array elements would be designated as %sales[1]%, %sales[2]% and so on up to the number of elements defined in the array.

Array elements start with [1] and not [0]. Each element stores a distinct value. The Join String command is used to combine multiple elements of an array into a single variable.



Variable array to join: %T%

Starting Element: 1

Ending Element: 3

Join the strings with

Destination: %Name%

Script Editor > Expand Variables Category > Join String

Variable Array to Join

Select the variable array from which the array elements are combined into a single variable.

Starting Element

Select the first element that is to be joined into a single variable.

Ending Element

Select the last element of the array variable to combine into the single variable.

Join the Strings With

Insert a character or text as a separator for each of the elements when combined into a single variable. Insert a space, a comma, a hyphen or any type text required. Or leave the field blank for no separation between elements.

Destination

Select the text string variable to store the combined variable elements.

Example

Assume a macro is using the ASCII File Begin Process command. The text file has 10 records on each line. The first record is saved to T[1], the second to T[2] and so on through T[10]. Record one contains a first name, record two a middle name and record three the last name. The remaining records contain address and phone information.

Our objective is to combine the three names into a single string, which the Join String command does quite easily. In this case we just need a space between each of the names, so that the names are not one long jumble. The image above shows how this would be done.

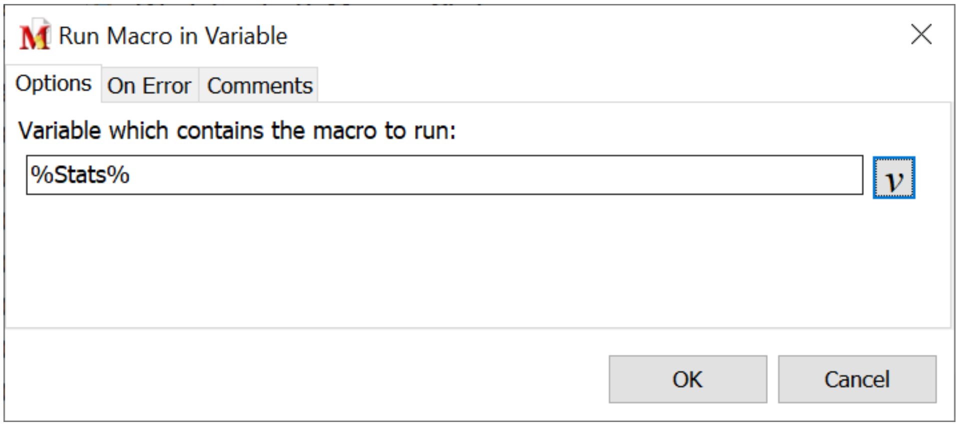
In the Join the Strings With edit box, place the cursor to the far left of the box and press the Space Bar once. This instructs the macro to separate each of the elements with a space.

So in our example, John is saved to T[1], Michael is saved to T[2] and Smith is saved to T[3]. The Join String command then saves John Michael Smith to variable %Name%.

Run Macro in Variable

Macro Express allows the contents of a variable to be loaded during the playback of a macro and executed as part of the macro. This can be used to change the behavior of the macro as it runs.

The "Run Macro in Variable" command is similar to the "Load Macro Text File" command, but because there is no file I/O, the "Run Macro in Variable" command runs much faster.



Variable which contains the macro to run: %Stats%

Script Editor > Expand Variables Category > Run Macro in Variable

Script Editor vs Direct (ASCII) Editor

When creating a macro that will run in a variable it needs to be written in ASCII format.

In the Script Editor, macro commands are displayed in an easily recognized and understood format. To achieve this simplicity, some of the details about each command are only displayed in the edit dialog for the macro command.

In the Direct Editor, all details about each macro command are visible. When describing these two ways of viewing macros, we refer to them as the Script and ASCII representations.

To view the ASCII representation of a macro click View > Direct Editor. Note that when in the Direct Editor view, all open macros are displayed in their ASCII form.

Another way to see the ASCII representation of a macro or macro commands is to highlight the commands you want to view, while in the Script Editor, and paste them into another program such as Notepad.

Examples

Script Format	ASCII Format
Text Type (Simulate Keystrokes): Text Type example	<TEXT TYPE Action="0" Text="Text Type example"/>
Variable Set String %Colors% to "blue"	<VARIABLE SET STRING Option="\x00" Destination="%Colors%" Value="blue"NoEmbeddedVars="FALSE"/>
If Variable %Colors% Equals "Blue"	<IF VARIABLE Variable="%Colors%" Condition="\x00" Value="Blue" IgnoreCase="FALSE"/>

Dynamically Create Macros

Macro Express Pro allows you to dynamically create macros. You can use one or more macros to create other macros that are different depending on events and choices that occur as the macro runs. In order to do this you need to put the ASCII representation of the macro commands that make up your macro into a variable.

For example, to include a Text Type command in the variable %RunThis% you would need to set the content of the %RunThis% variable.

Variable Set String %RunThis% "<TEXT TYPE Action="0" Text="A Text Type example"/>"

Expansion of Variables

Normally commands that deal with variables substitute the contents of a variable for the variable name. For example, %FirstName% will be changed to something like George. When using the Run Macro in Variable command, often the name of a variable is needed instead of the content of the variable. Up until now, the automatic substitution with the content of a variable has made it awkward to use the Run Macro in Variable command.

The option 'Do not process embedded variables' has been added to many of the Variable Set String and Variable Modify String commands. This makes it easier to create a variable that contains variable names for use in the Run Macro in Variable command.

We have added another method of specifying a variable name. If you surround the variable name with {%} then the content of the variable will not be used. Instead the name of the variable will be passed to the Run Macro in Variable command. If you include {%}FirstName{%} it will be changed to %FirstName% in the Run Macro in Variable command. If you include %FirstName%, the content of the variable will be passed to the Run Macro in Variable command.

A) When a variable is included in a macro command, the content of the variable is used when the macro runs. Consider this example:

```
   Variable Set String %My% to "My example"
    Variable Set String %RunThis% "<TEXT TYPE Action="0" Text="%My% of the
Text Type command"/>"
```

When this runs the %RunThis% variable will become "<TEXT TYPE Action="0" Text="My example of the Text Type command"/>"

B) There are times, however, when you want the %FirstName% variable to contain the name of a variable, not its content. To do this change the command to look like this:

```
    Variable Set String %FirstName% "<VARIABLE SET STRING Option="\x00"
    Destination="{%}FirstName{%}" Value="%Name%"
    NoEmbeddedVars="FALSE"/>"
```

The value "{%}FirstName{%}" will be changed to %Name% and will not be changed to the content of variable %Name%.

✓ Note: If you need to include the value {%} in your macro use this {{%}}. It will be changed to {%}.

C) Another way to encode a variable is to use the \c parameter. This option expects two hexadecimal digits to represent a single character. The sample above may be written this way:

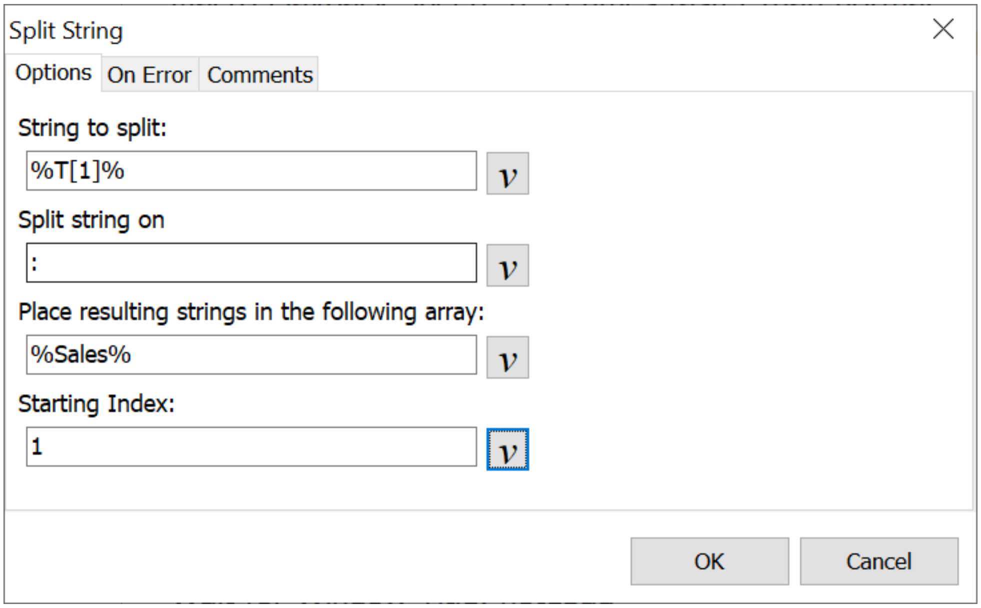
```
    Variable Set String %RunThis% "<VARIABLE SET STRING Option="\x00"
    Destination="\c25FirstName\c25" Value="%Name%"
    NoEmbeddedVars="FALSE"/>"
```

This works because 25 is the hexadecimal value for %. The \c is case sensitive, \C will not work.

✓ **Note:** When including file paths in the Run Macro in Variable command, be careful when including certain characters in the path. For example, if needing to include a path like c:\charlie the \c will be interpreted as a hexadecimal character command. There are two ways to avoid this problem. You can change the case to avoid this situation by using c:\Charlie, for example. Or you may use a double backslash: c:\\charlie.

Split String

The Split String command provides additional power and flexibility to text string variable manipulation. It provides a quick method to break down a text string into smaller components and save each of the components to individual variable array elements.



String to split:
%T[1]%

Split string on
:

Place resulting strings in the following array:
%Sales%

Starting Index:
1

Script Editor > Expand Variables Category > Split String

String to Split

Select the text string to be manipulated and saved to individual variable array elements.

Split String On

Input the value that separates the original text string. This value may be a <space>, a comma, a dash (-) or any other character or text string.

In the image above, all text up to the first ":" symbol in the T[1] string is saved to variable Sales[1]. The text after the first ":" symbol and up to the second ":" symbol is saved to variable Sales[2], and so on. The ":" symbol is not saved as part of the new variable.

- ✓ **Note:** To split the string based on spaces in the string, place the cursor in this field and press the space bar once.

Place Resulting Strings in the Following Array

Select the variable array to store the individual components of the original text string.

Starting Index

Select which variable in the array is to receive the first record. Based on the image above, the first split is saved to variable Sales[1]. Array elements start with [1] and not [0].

Example

Below are a couple of examples that help illustrate this feature.

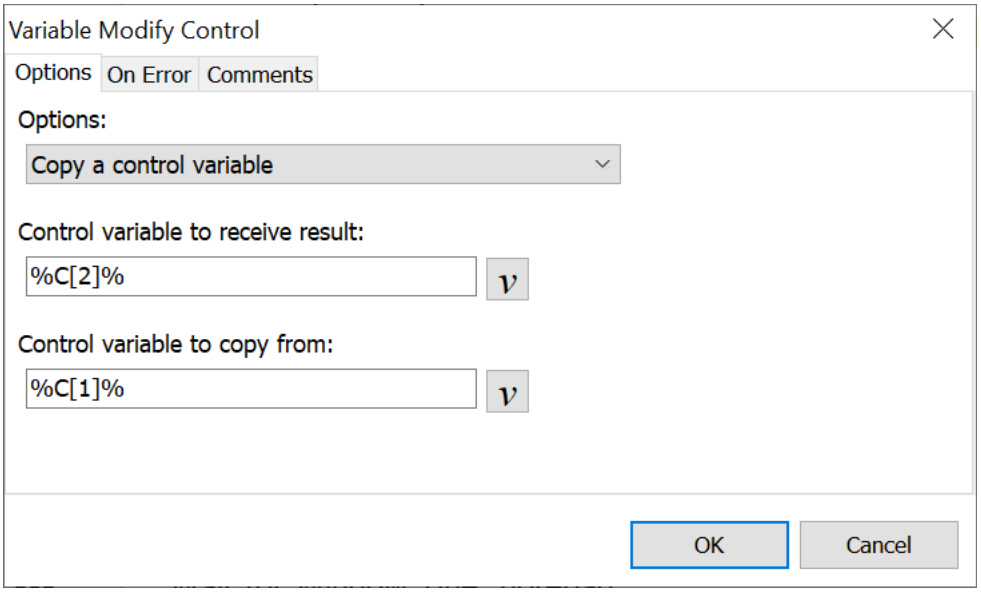
1. The text in the String to Split field is This is a test. The Split String On field is populated with a <Space>. The resulting string array is variable T, starting with Index 1. The values saved to variables T[1] through T[4] are shown below.

T[1] - This
T[2] - is
T[3] - a
T[4] - test

2. The text in the String to Split field is 801-927-5011. The Split String On field is populated with a dash (-). The resulting string array is variable T, starting with Index 5. The values saved to variables T[5] through T[7] are shown below.

T[5] - 801
T[6] - 927
T[7] - 5011

Variable Modify Control



Options: Copy a control variable

Control variable to receive result: %C[2]%

Control variable to copy from: %C[1]%

Script Editor > Expand Variables Category > Variable Modify Control

Copy a Control Variable

This option simply copies the information saved to one control variable to another. This may be useful if needing to retain the value of the original variable, but at the same time manipulate the value. Use this command to make a copy of the variable and now do both.

Control Variable to Receive Result

Select the variable to receive the information from the control variable being copied.

Control Variable to Copy From

Select the control variable that contains the information to copy to another variable.

Copy the Control from the Activation

This command can be used only if the macro is a Control Activated macro. For example, later in the macro script it may be necessary to modify the control that activated the macro. Rather than use the Get Control function and capture the control again, save the activation control to a variable.

Control Variable to Receive Result

Select the variable to receive the information from the control activated macro.

Modify the Top-Level Window Title

When the Get Control or Capture Control commands are used, Macro Express saves a description of the Window Controls in the control variable. This information is used to identify the control. The information includes the name of the program and the name of the top-level window that contains the control.

- ✓ **Note:** This command modifies the content of a control variable. It does not modify the control itself.

Many programs, including but not limited to Internet Explorer, change the title of the top level window depending on what is happening within the program. By using the 'Modify the Top-Level Window Title' command, the Control Variable may be changed so that the Get Control or Capture Control commands are not needed again.

Partial/Exact Matches

The window title can be either a partial or an exact match. For example, if using a control in Internet Explorer, use "Internet Explorer" as the window title, despite what the window title shows, and specify a partial match.

Use the Partial Title option of the 'Modify Top-Level Window Title of Control Variable' command to modify the window title (stored in the Control variable) to work with either Edit or View modes. In the example above, the title could be set to 'Sample #5'. To make this even more flexible, change it to 'Sample #' so that it works in Edit and/or View modes and with all items in the database.

Example

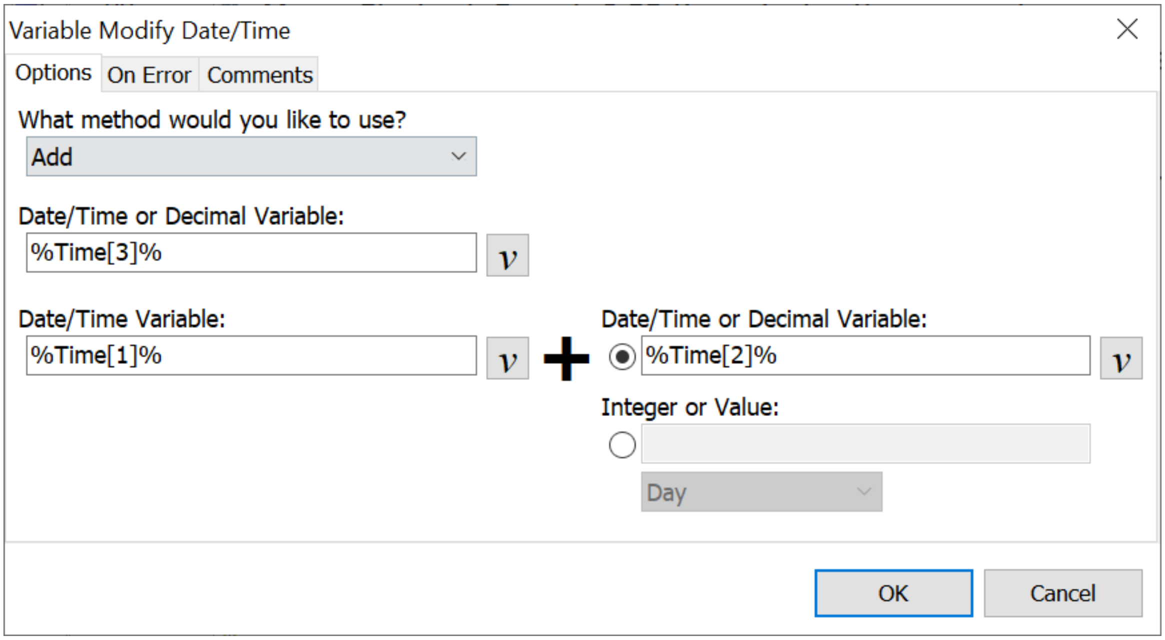
Consider a program that has ‘Edit’ and ‘View’ modes. When in Edit mode the title bar displays ‘Edit Sample #5’. When in View mode the title bar shows ‘View Sample #5’. In this example, there is an OK button on the form.

If Edit mode is open when the macro is created to access the OK button on the form, the macro will not work when in View mode … even though it is the same control. There are two choices. Use two Get Control commands to capture the control in both Edit and View modes. Or, use one Get Control command and use the ‘Modify the Top-Level Window Title" command to change the Window title.

When using controls, the top-level window is the main window that contains the control. This command allows the user to adjust what window title the control looks for. For example, Internet Explorer changes its window title often. This option allows the user to define one control variable and use it to work with different windows in order to perform actions with the desired control.

Variable Modify Date / Time

Use this command to make date and time calculations. The original date and time must be saved as a Date/Time variable in order to process the calculations.



What method would you like to use? Add

Date/Time or Decimal Variable: %Time[3]%

Date/Time Variable: %Time[1]% + Date/Time or Decimal Variable: ● %Time[2]%

Integer or Value: ○ Day

Script Editor > Expand Variables Category > Variable Modify Date/Time

What Method Would you Like to Use?

Select the option to perform from the drop down list.

- **Add** - Add dates or times together.
- **Subtract** - Subtract dates or times from another date or time.
- **Extract Day** - Extract the day from the date variable and save to another variable.
- **Extract Month** - Extract the month from the date variable and save to another variable.
- **Extract Year** - Extract the year from the date variable and save to another variable.
- **Extract Hour** - Extract the hour from the time variable and save to another variable.
- **Extract Minute** - Extract the minute from the time variable and save to

another variable.

- **Extract Second** - Extract the second from the time variable and save to another variable.
- **Convert to Text Variable** - Convert the Date/Time variable into a text string variable.

Add or Subtract Commands

Destination Date/Time or Decimal Variable

Insert the date/time or decimal variable to receive the result of the addition or subtraction.

Date/Time Variable

Select the date and time variable to be used as part of the calculation

Date/Time Variable - Integer Value

Insert a second date/time variable to be part of the calculation or insert an integer value. With the integer value option, the macro adds or subtracts the amount of the integer from the date or time. Select from the drop down list to add or subtract from the day, month, year, hour, minute, second or millisecond.

Extract Commands

Date/Time Variable

Select the variable from which the date or time component is to be extracted and saved to an integer variable.

Destination Integer Variable

Select the integer variable to store the value extracted from the date/time variable.

Convert to Text Variable

This option converts the date/time variable into a text string variable.

Date/Time Variable

Select the Date/Time variable to be converted to a text string variable.

Format to Convert Date/Time to:

Select the date or time format to use with the text string variable.

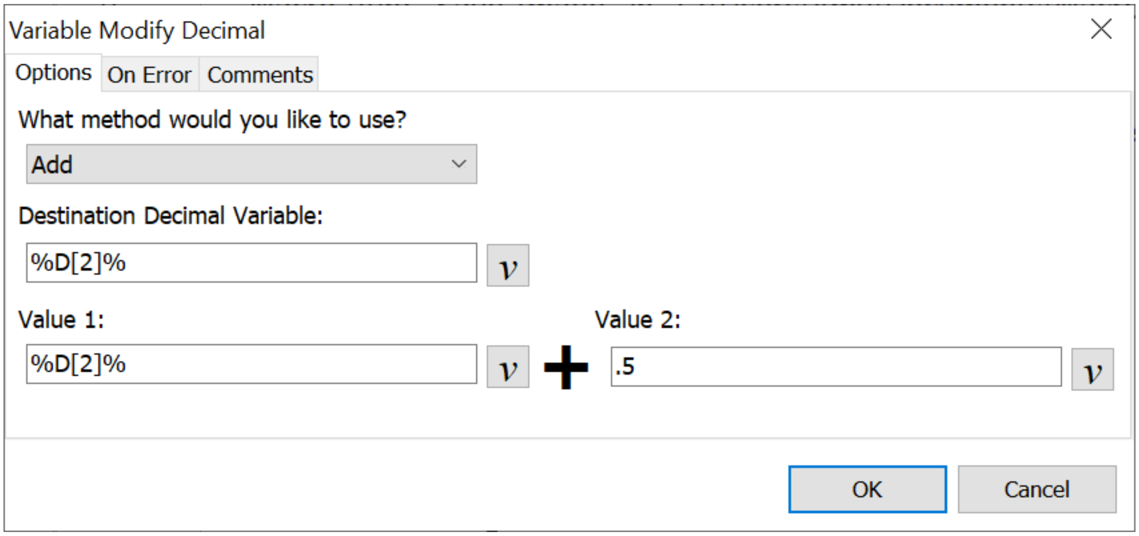
Text Variable to Receive Result

Select the text string variable to store the value from the date/time variable.

Variable Modify Decimal

Use the Variable Set Decimal command to assign a decimal value to a variable. The commands listed below modify or manipulate the original decimal variable.

- ✓ **Note:** A decimal variable (D[1]) can range from 5.0 x 10^ -324 to 1.7 x 10^308 with 15 to 16 significant digits. This first example is a decimal expanded to include 324 leading zeros followed by the number 5, represented as .000(321 more zeros)5. The second range extends to 17 followed by 308 zeros which would look like 1700000000(300 more zeros).0. The numbers are rounded to 15 or 16 significant digits.



What method would you like to use? Add

Destination Decimal Variable: %D[2]%

Value 1: %D[2]% + Value 2: .5

Script Editor > Expand Variables Category > Variable Modify Decimal

Add

Add a decimal variable to a number or to a second decimal variable. Save the results to another decimal variable.

Destination Decimal Variable

Select the variable to receive the results of the addition.

Value One /Value Two

Insert either a decimal value or a decimal variable into these fields. The values are added and saved to the variable selected.

Subtract

Subtract a decimal variable from a number or from another decimal variable or subtract a number from a decimal variable. The result is placed in the decimal variable selected.

Decimal Variable

Select the variable to receive the results of the subtraction.

Value One /Value Two

Insert either a decimal value or a decimal variable into these fields. The values are subtracted and saved to the variable selected.

Multiply

Multiply a decimal variable with a number or with another decimal variable and place it in the decimal variable selected.

Decimal Variable

Select the variable to receive the results of the multiplication.

Value One /Value Two

Insert either a decimal value or a decimal variable into these fields. The values are multiplied and saved to the variable selected.

Divide

Divide a decimal variable with a number or with another decimal variable or divide a number with a decimal variable. The result can be placed in the decimal variable chosen.

Decimal Variable

Choose the variable to receive the results of the division.

Value One /Value Two

Insert either a decimal value or a decimal variable into these fields. The values are divided and saved to the variable selected.

Round

This command rounds the value of a decimal variable to the number of decimal places selected. If copying a decimal value and saving it to a variable or modifying a decimal variable through multiplication or division, use this feature to trim the number of decimal places. Just select the variable to round and insert the number of decimal places.

Decimal Places to Round to

Insert the number of decimal places to save to the variable.

✓ **Note:** The Round Decimal Variable command uses asymmetric arithmetic rounding to determine how to round values that are exactly midway between the two values that have the desired number of significant digits. This method always rounds to the larger value.

The following examples illustrate the use of the Rounding command:

For example:
If %D[1]% is 1.234 then Round Decimal Variable %D[1]% to 2 decimal places results in 1.23
If %D[1]% is 1.235 then Round Decimal Variable %D[1]% to 2 decimal places results in 1.24
If %D[1]% is -1.235 then Round Decimal Variable %D[1]% to 2 decimal places results in -1.23

Convert to Text String

Use this command to convert a decimal variable to a text variable. Specify the number of decimal places to use.

Destination String Variable

Select the string variable to receive the decimal variable value.

Decimal Places

Insert the number of decimal places to save to the text string variable.

Truncate to Integer

Use this command to convert a decimal variable to an integer variable. Any fractional part of the decimal value is truncated.

Integer Variable

Select the integer variable to receive the truncated decimal variable value.

Remove Integer Portion

Use this command to remove the integer value portion (whole numbers) from the decimal variable. For example, if the decimal variable has a value of 1.9087653, the new value would be .9087653.

Destination Decimal Variable

Select the decimal variable to receive the value with the integer portion removed.

Copy Value

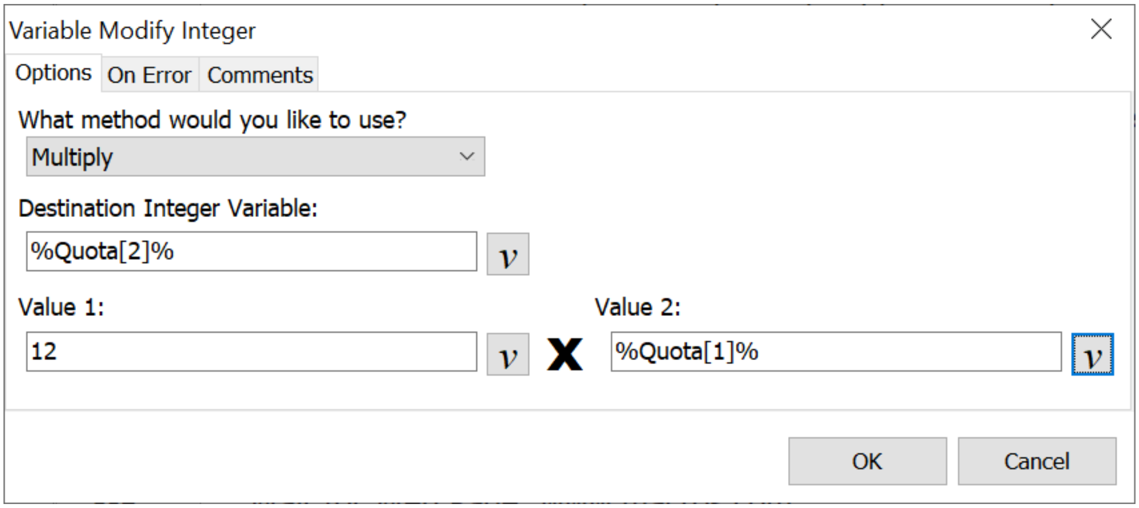
Copy the value of one decimal variable to another decimal variable.

Destination Decimal Variable

Select the decimal variable to receive the value of the copied decimal variable.

Variable Modify Integer

Use these commands to modify or manipulate an integer variable.



What method would you like to use?
Multiply

Destination Integer Variable:
%Quota[2]%

Value 1:
12

X

Value 2:
%Quota[1]%

Script Editor > Expand Variables Category > Variable Modify Integer

Add

Add an integer variable to a number or to another integer variable and place the sum into an integer variable.

Destination Integer Variable

Choose the variable to receive the results of the addition.

Value One /Value Two

Insert either an integer value or an integer variable into these fields. The values are added and saved to the variable selected.

Subtract

Subtract integers and/or integer variables. The result is placed in the integer variable chosen.

Destination Integer Variable

Select the variable to receive the results of the subtraction.

Value One /Value Two

Insert either an integer value or an integer variable into these fields. The values are subtracted and saved to the variable selected.

Multiply

Multiply an integer variable with a number or with another integer variable and place the result into the integer variable selected.

Destination Integer Variable

Select the variable to receive the results of the multiplication.

Value One /Value Two

Insert either an integer value or an integer variable into these fields. The values are multiplied and saved to the variable selected.

Divide

Divide an integer variable with a number or with another integer variable or divide a number with an integer variable. The result is placed into the integer variable chosen.

Destination Integer Variable

Select the variable to receive the results of the division.

Value One /Value Two

Insert either an integer value or an integer variable into these fields. The values are divided and saved to the variable selected.

Convert to String

This command converts an integer variable to a text variable.

Integer Variable

Select the variable to be converted to a text string variable.

New Variable

Select the name of the new variable to contain the contents of the integer variable.

Convert to Decimal

Converts an integer variable to a decimal variable.

Integer Variable

Select the variable to be converted to a decimal variable.

New Variable

Select the name of the new variable to contain the contents of the integer variable.

Copy Value

This command copies the value of one integer variable to another integer variable.

Destination Integer Variable

Select the variable to be receive the copied information.

Variable to Copy

Select the variable to copy its contents to another integer variable.

Increment

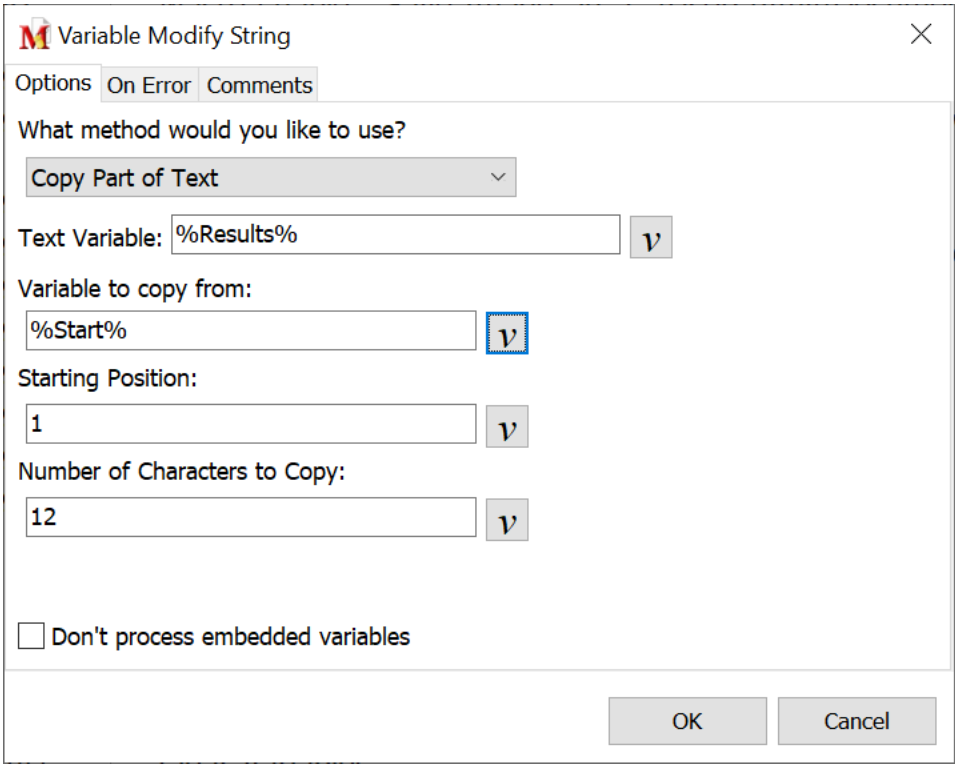
This option increases the value of the selected variable by one. For example, if variable N[1] had an original value of 25, the value would change to 26 after running this command.

Decrement

This option decreases the value of the selected variable by one. For example, if variable N[1] had an original value of 25, the value would change to 24 after running this command.

Variable Modify String

The functions discussed in this topic modify the values of text string variables previously created in the macro.



What method would you like to use?

Copy Part of Text

Text Variable: %Results%

Variable to copy from:

%Start%

Starting Position:

1

Number of Characters to Copy:

12

□ Don't process embedded variables

Script Editor > Expand Variables Category > Variable Modify String

Don't process embedded variables

This option is for use when the variable being modified contains variables or a combination of % symbols. Select this option to have the macro ignore any variables embedded in the variable to be modified.

In the image above, we will assume that the variable %Start% contains another variable - %T[15]%. When the "Don't process embedded variables" option is selected %T[15]% is saved to the %Results% variable rather than the expanded value of the variable.

This feature is available for the following Variable Modify String options: Append Text, Copy Whole Text, Copy Part of Text, Replace Substring, Save to Clipboard, Save to Text File, Append to Text File, Save to INI File and Save to Environment Variable.

Trim

The Trim option removes all spaces and control characters on the left and on the right of the text string variable being modified.

Left Trim

Left Trim removes all spaces and control characters to the left of the text string variable being modified.

Right Trim

Right Trim removes all spaces and control characters to the right of the text string variable being modified.

Strip CR/LF

Strip CR/LF removes any Carriage Returns and Line Feeds embedded in the text string variable being modified.

Convert to Integer

Use this function to convert the value of a text string variable to an integer variable. If the text variable does not contain a valid value to be converted to an integer (i.e. non-numeric characters), then the integer value is set to zero.

Convert to Decimal

Use this option to convert the value of a text string variable to a decimal variable. If the text variable does not contain a valid value to be converted to a decimal (i.e. non-numeric characters), then the decimal value is set to zero.

Append Text

The Append Text function adds text to the end of an existing text string variable.

Text to Append

Insert the text to be added to the variable selected.

Append Text String Variable

The Append Text String Variable command combines the contents of one text variable with another text variable. The text is appended to the end of the receiving variable.

Text Variable

Select the variable to receive the additional text from the second variable.

Text Variable to Append

Select the variable to be appended to the first string variable.

Copy Whole Text

The Copy Whole Text command copies the contents of a text string variable to new text string variable.

Text Variable

Select the variable to receive the copy.

Text Variable to Copy

Select the variable to be copied to another variable.

Copy Part of Text

The Copy Part of Text function copies a portion of the contents of one text string variable to another text string variable. Specify the starting position and the number of characters to copy. The starting position and the characters to copy can be set to either a value or set to an integer variable. Integer variables can be created using the length of a text variable or the position of text in a text variable, providing much greater flexibility in determining what to copy.

Text Variable

Select the variable to receive the copy.

Variable to Copy From

Select the variable from which to copy a portion of the text.

Starting Position

The Starting Position indicates the first character to copy. To start with the 5th character in the variable, insert 5 as the starting position.

Number of Characters to Copy

Insert into this field the number of characters to copy.

Delete Part of Text

The Delete Part of Text command deletes a portion of the contents of a text string variable. Specify the starting position and the number of characters to delete. The starting position and the characters to delete can be set to either a value or set using an integer variable. Integer variables can be created using the length of a text variable or the position of text in a text variable, providing much greater flexibility in determining what to copy.

Text Variable

Select the variable from which to delete part of the text.

Starting Position

The Starting Position indicates the first character to be deleted. If starting with the 5th character in the variable, insert 5 as the starting position.

Number of Characters to Delete

Insert in this field the number of characters to delete.

Uppercase

The Uppercase command converts the entire contents of a text string variable to uppercase letters.

Lowercase

The Lowercase command converts the entire contents of a text string variable to lowercase letters.

Pad Left

The Pad Left function pads the left side (beginning) of a text string variable with spaces. Specify the width that the resulting variable should occupy. For example, if the text variable contained the name "Fred" and the Pad Left width is set to 15, a total of 11 spaces would be added to the left side of (before) "Fred" since "Fred" takes up 4 spaces (4 + 11 = 15). The pad left can be used for right justifying values such as columns of numbers.

Width of Text

Insert the total width that the variable should occupy.

Pad Right

The Pad Right function pads the right side (end) of a text string variable with spaces. Specify the width that the resulting variable should occupy. For example, if the text variable contained the name "Fred" and the Pad Right width is set to 15, a total of 11 spaces would be added to the right side (end) of "Fred" since "Fred" takes up 4 spaces (4 + 11 = 15). The pad right can be used for left justifying values.

Width of Text

Insert the total width that the variable should occupy.

Replace Substring

The Replace Substring option locates a string of text within the text string variable and replaces it with other text.

Text to Replace

Enter the text found in the variable selected that is to be replaced.

Replace Text With

Insert the text to replace the text currently found in the variable.

Replace All Instances

Check this box to have every instance of the text string replaced. With this box not checked, the first instance of the text string is replaced. Any other instances are ignored.

Ignore Case

The Ignore Case option ignores upper and lower case letters in the substring. The replacement takes place even if the case does not match. If this option is not selected, an exact match is required in order for the replacement to occur.

Save to Clipboard

This option saves the contents of the selected text string variable to the Windows

Clipboard.

Save to Text File

Save to Text File saves the contents of the selected text string variable to a text file.

Filename

Select the file to receive the text saved to the variable. Insert the name of the file and path or click on the Browse button to locate the text file.

Add a Trailing CR/LF

This option inserts a carriage return or line feed to the end of the text string that is saved to the text file.

Append to Text File

Append to Text File appends the contents of the text string variable to the end of a text file. The text is positioned after any text that already exists in the file.

Filename

Select the file to receive the text saved to the variable. Insert the name of the file and path or click on the Browse button to locate the text file.

Add a Trailing CR/LF

Add a Trailing CR/LF option inserts a carriage return or line feed at the end of the text string saved to the file. Any subsequent appending to the text file would start on a new line.

Save to INI File

Save the contents of a text string variable to an INI file.

INI File Name

Insert the INI file name and path in this field or click on the Browse button to locate the file.

Key - Insert the Key name in this field.
Value - Insert the Value name in this field.

Save to Environment Variable

Windows maintains "global" variables that the system can access at any point. Some common ones are as follows:

> **temp** -- holds the location of the temporary folder on the computer
> **windir** -- holds the folder name where windows is installed
> **path** -- holds a number of folder names where files should be searched if a file cannot be found in the current folder

Use this command to set up environmental "global" variables for later use or to modify existing variables. A complementary command that can "read" an environment variable is part of the Variable Set String commands.

Additional information about Environmental Variables from Microsoft

To determine which Environment variables exist on your computer, run the Command Prompt. Type set in the Command Prompt window and press Enter to view the Environment variables.

--- Definition #1 ---

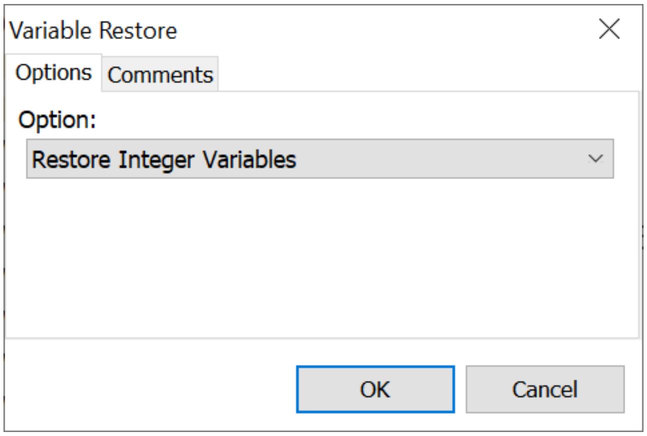
Environment variable
An element of the operating system environment, such as a path, a directory name, or a configuration string. Environment variables are typically set within batch files.

--- Definition #2 ---

Environment variables specify search paths for files, directories for temporary files, application-specific options, and other similar information. The system maintains an environment block for each user and one for the computer. The system environment block represents environment variables for all users of the particular computer. A user's environment block represents the environment variables the system maintains for that particular user, including the set of system environment variables.

Variable Restore

The Variable Restore command used in conjunction with the Variable Save command is the key to passing variables between independently running macros.



Option: Restore Integer Variables

Script Editor > Expand Variables Category > Variable Restore

Used with the Variable Save command, variables can be passed quite easily from one macro to the next on independent running of macros. Where one macro saves out the variables to memory, the next macro can restore variable values used in the first macro.

The Variable Restore command restores values of variables from system memory. Select to restore all variables or a specific variable type, such as Text String, Control, Integer, etc.

Example

At the end of the Macro A script, insert a Variable Save command to save all variable values to memory.

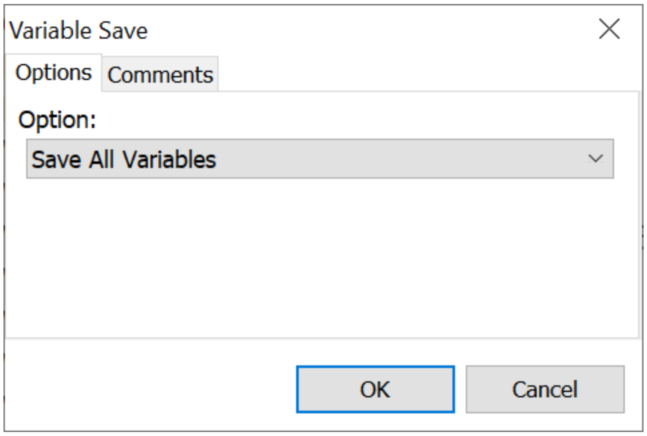
At the beginning of the Macro B script, insert a Variable Restore command. This utilizes the values of the variables created in Macro A in Macro B.

✓ **Note 1**: Upon closing the Macro Express Player or rebooting Windows, the values of the variables saved to memory are lost.

✓ **Note 2:** Only Global variables may be restored. Local variables are not saved with the Variable Save command and therefore cannot be restored.

Variable Save

This command, along with the Variable Restore command, is the key to passing variables between independently running macros.



Option: Save All Variables

Script Editor > Expand Variables Category > Variable Save

Used in conjunction with the Variable Restore command, variables can be passed quite easily from one macro to the next on independent running of macros. Where one macro saves out the variables to memory, the next macro can restore variable values used in the first macro.

The Variable Save command saves the values of variables set in the macro to system memory. Select to save all variables or a specific variable type, such as Text String, Control, Integer, etc.

Example

At the end of the Macro A script, insert a Variable Save command to save all variable values to memory.

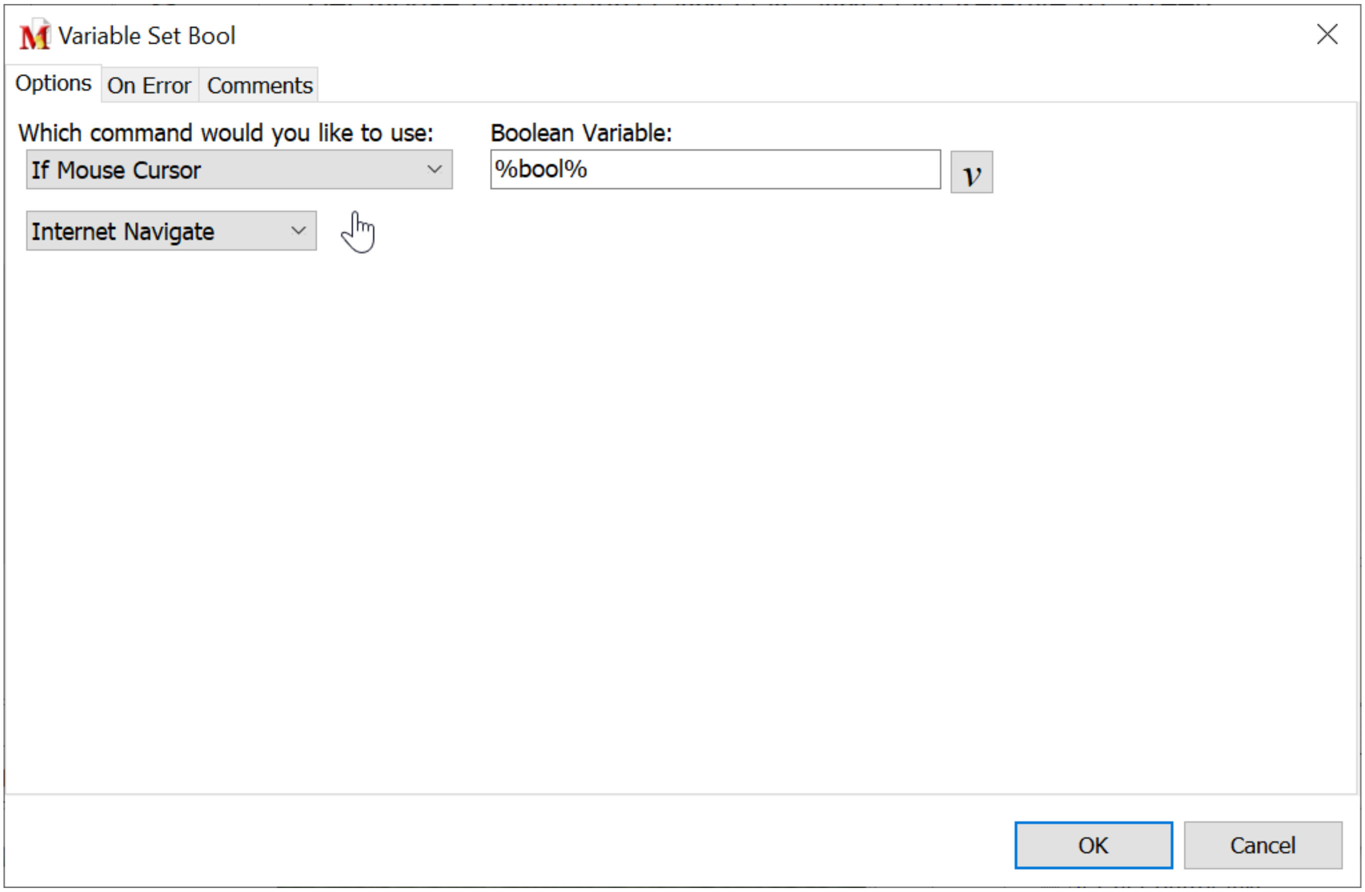
At the beginning of the Macro B script, insert a Variable Restore command. This utilizes the values of the variables created in Macro A in Macro B.

- ✓ **Note 1:** Upon closing the Macro Express Player or rebooting Windows, the values of the variables saved to memory are lost.

- ✓ **Note 2:** Only Global variables may be restored. Local variables are not saved with the Variable Save command and therefore cannot be restored.

Variable Set Boolean

A Boolean Variable holds either one of two values: True or False.



Which command would you like to use: If Mouse Cursor

Internet Navigate

Boolean Variable: %bool%

Script Editor > Expand Variables Category > Variable Set Boolean

Which command would you like to use:

Select the If command to use with the Set Boolean command. When the macro is run, the If Statement either returns a value of "Yes the statement matched" or "No the statement did not match". If the statement matched then it produces a value of "True" in the Boolean Variable. If the statement did not match then a value of "False" is saved to the Boolean Variable.

Boolean Variable

Select the Boolean variable to receive the results.

Sample Macro

Let's look at a quick example using the If Mouse Cursor option as shown in the image above. The Boolean variable was assigned the name of "Bool". The If Mouse Cursor option looks for an Arrow cursor.

For this example, first assume that the macro saves a large file. While the file is saving, the mouse cursor displays as an Hourglass. We don't want the macro to resume until the file has finished saving. The Hourglass changing back to the Arrow indicates that the save has finished.

In the macro script shown below, the Repeat Until command runs in a loop until the value of "Bool" equals True. Based on how we defined the variable in the Variable Set Bool command, "True" signifies that the mouse cursor is an Arrow. When the file save has finished, the mouse cursor returns to an Arrow and the True condition is met. The Repeat loop terminates and the macro continues.

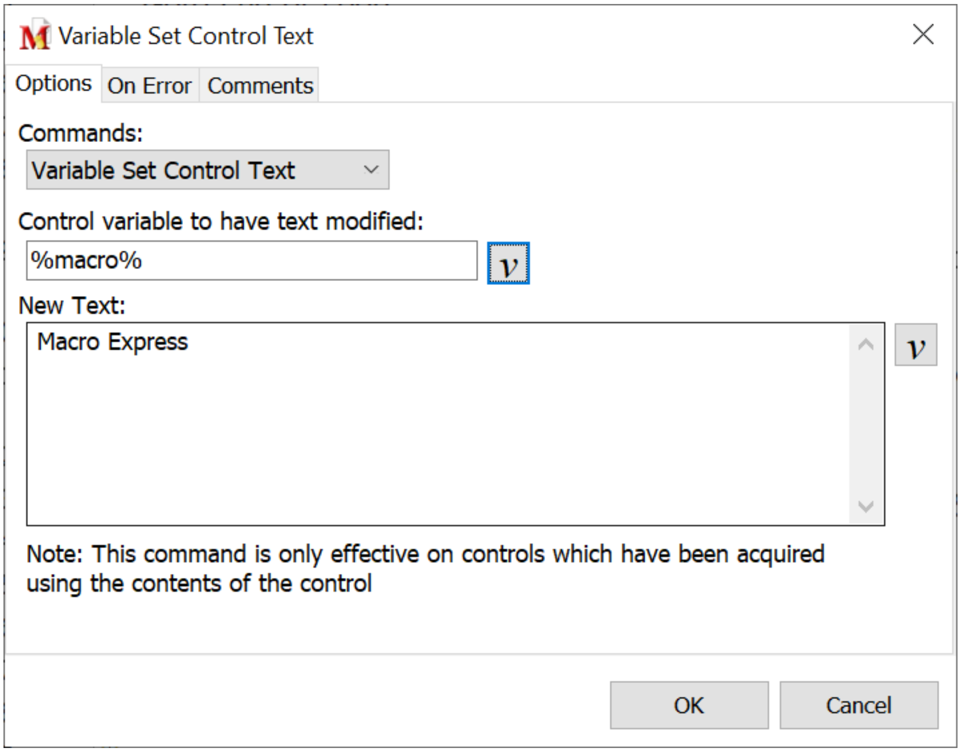
Macro Script

```
<Macro commands to save file>
Repeat Until %Bool% Equals True
  Variable Set Bool: %Bool%
End Repeat
<Macro commands after file is saved>
```

Variable Set Control Text

The Variable Set Control Text command changes the text in the control variable in order to look for the control using different text. For example, if the "Get Control" command was used to capture an edit box, Macro Express will not be able to find the edit box on playback if the text changes.

Below is an example of how this works. Capture an edit box containing the word "Testing". Subsequently the word "Testing" is changed to the word "Macro". In this case the macro would not find the edit box. But, knowing that the edit box could contain the word "Macro" then the Variable Set Control Text command will help. Set the New Text to "Macro" and the edit box would be found if it contained the word "Macro".



Commands: Variable Set Control Text

Control variable to have text modified: %macro%

New Text: Macro Express

Note: This command is only effective on controls which have been acquired using the contents of the control

Script Editor > Expand Variables Category > Variable Set Control Text

Control Variable to have Text Modified

Select the control that is to be modified.

New Text

Insert the text that will modify the control variable.

- ✓ **Note 1:** This command affects the Control Variable only. It does not affect the control in the application.

- ✓ **Note 2:** Macro Express uses two different methods to identify a control.

  Method one identifies the Class name of the control and the Text contained in the control. For example an OK button may have a class name of Button and the text of OK.

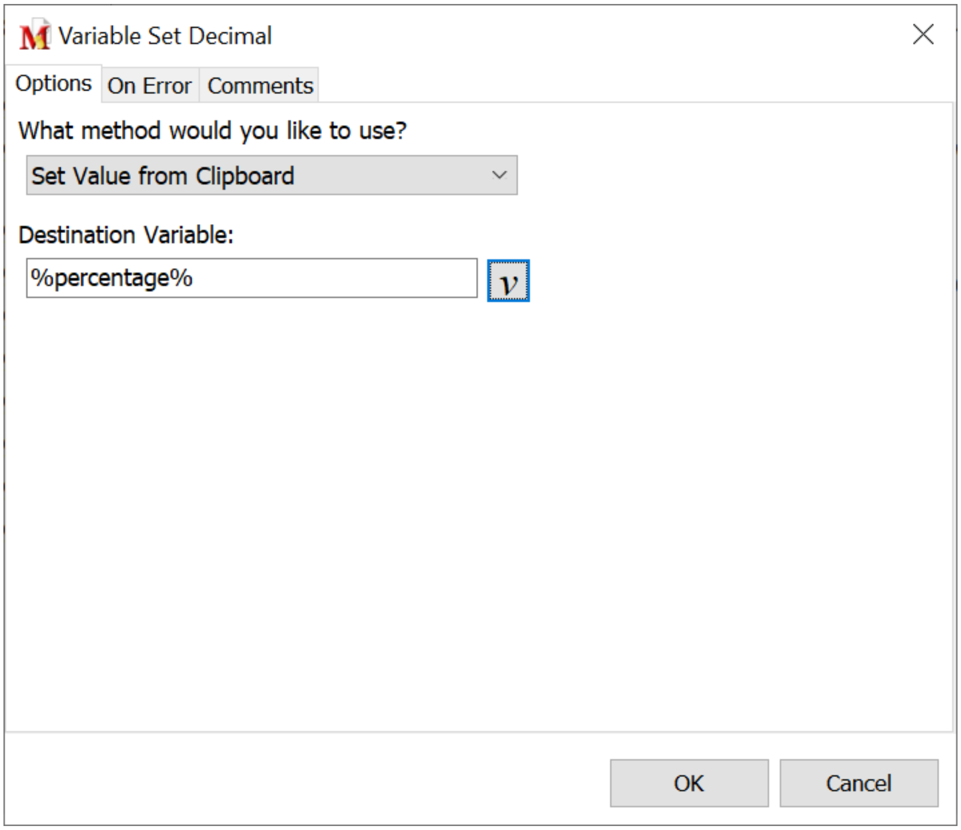
  Method two identifies the Class name and what is called the Z index. Basically the Z index is the order that the control appears on the form. For example you may have two buttons, "OK" and "Cancel". Macro Express will identify the OK button as class name Button and Z index of 12, but will identify the Cancel button as class name Button and Z index 13.

  The second method is the only one that can be used with the Variable Set Control Text command. Select which method to use when capturing a control. Leave the Capture Using Text option unchecked to capture using this second method.

Variable Set Decimal

The Variable Set Decimal command assigns a value to a decimal variable. This command provides four options for setting a decimal variable. Use the Modify Decimal Variables command to manipulate the original values assigned to a macro.

- ✓ **Note:** A decimal variable can range from 5.0 x 10^ -324 to 1.7 x 10^308 with 15 to 16 significant digits. This first example is a decimal expanded to include 324 leading zeros followed by the number 5, represented as .000(321 more zeros)5. The second range extends to 17 followed by 308 zeros which would look like 1700000000(300 more zeros).0. The numbers are rounded to 15 or 16 significant digits.



What method would you like to use?

Set Value from Clipboard

Destination Variable:

%percentage%

Script Editor > Expand Variables Category > Variable Set Decimal

What Method Would You Like to Use?

Select from the drop down list the method required for defining the variable. Each of the options are described below.

Set Value Now

The Set Value Now option assigns a value to the decimal variable now for use later in the macro.

Destination Variable

Select a variable to use or create a new decimal variable.

Initial Value

Insert the contents of the variable in this field.

Prompt for Value

The Prompt for Value function opens an edit box during macro playback. The user is prompted to insert a decimal value.

Destination Variable

Select a variable to use or create a new decimal variable.

Prompt

Optionally enter text in the Prompt edit field to explain the type of value the user should input. During macro playback, the instructions appear above the input field.

Mask the Input

The Mask the Input option masks the text entered into the input box and is viewed as *** instead of the actual digits.

The Dialog Should Always be on Top

This option makes sure that the prompt window is always on top of other open windows.

Positioning the Prompt Window

Use this feature to define the position on the screen where the input box will appear during the playback of the macro. Insert exact coordinates or use the Set Position Manually button to set where the window should appear. See the Window Positioning topic for additional information.

Hint: Setting a Default Value in the Prompt Window

First use the Set Value Now option. Insert the decimal value that you want set as the default. Then use the Prompt for Value option. Both options must reference the same variable, such as D[1]. When the macro is run, the prompt window displays the default value. The user can overwrite, amend, accept or append to this value.

Below is a sample of the commands as they would look in the Scripting Editor. The decimal .52 is the default value.

```
Variable Set Decimal %D[1]% to .52
Variable Set Decimal %D[1]% from
  Prompt
```

Set Value from Clipboard

Set Value from Clipboard examines the text contents of the clipboard. It takes this value and strips off any spaces, carriage returns or line feeds. It then attempts to convert the contents to a decimal value and place it in the variable. If a non-decimal value (i.e. a string of text) is in the clipboard or the text clipboard is empty, the variable value is set to zero.

Destination Variable

Select a variable to use or create a new decimal variable.

Display Size Multiplier

Windows allows users to adjust the size of text, apps, and other items on the screen by changing the Display Settings. Typical settings are the default of 100% or custom settings such as 125%, 150% or 200%. There is also an option to set a custom scaling level (not recommended). Windows changes the size of items on the screen by adjusting the Dots Per Inch (DPI).

Macros designed for one screen size setting will not be properly displayed on a computer with a different screen size. Windows automatically adjusts the font size but the size and positioning of commands such as Text Box Display, If Message and Complex Pause will need to be changed to allow it to fit in the box size selected.

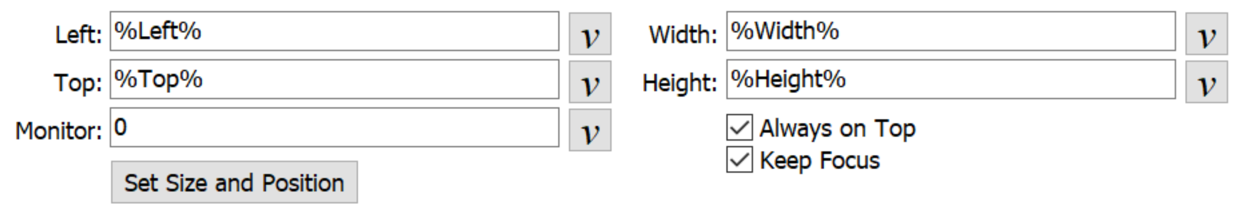
The Display Size Multiplier command returns a value that can be used to adjust the screen position and size to allow a macro to work correctly on computers with different screen size settings.

Example

To create a Text Box display with these parameters: Left: 50, Top: 10, Width: 300 and Height: 250, enter the values into variables, as shown below, rather than entering the values directly into the Text Box Display command.

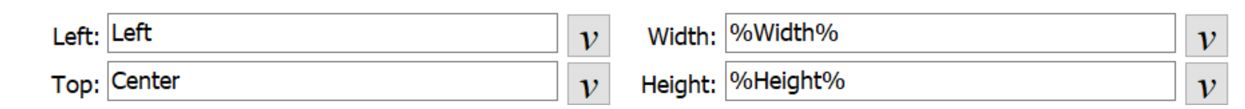
```
Variable Set Decimal %DisplaySizeMulitplier% to Display Size Multiplier
Variable Set Integer %Left% to 50
Variable Set Integer %Top% to 10
Variable Set Integer %Width% to 300
Variable Set Integer %Height% to 250
// Left Coordinate
Variable Modify Decimal: %lft% = %Left% * %DisplaySizeMulitplier%
Variable Modify Decimal: Round %lft% to 0 decimal places
Variable Modify Decimal %lft%: Truncate to Integer (%Left%)
// Top Coordinate
Variable Modify Decimal: %tp% = %Top% * %DisplaySizeMulitplier%
Variable Modify Decimal: Round %tp% to 0 decimal places
Variable Modify Decimal %tp%: Truncate to Integer (%Top%)
// Width
Variable Modify Decimal: %wid% = %Width% * %DisplaySizeMulitplier%
Variable Modify Decimal: Round %wid% to 0 decimal places
Variable Modify Decimal %wid%: Truncate to Integer (%Width%)
// Height
Variable Modify Decimal: %hgt% = %Height% * %DisplaySizeMulitplier%
Variable Modify Decimal: Round %hgt% to 0 decimal places
Variable Modify Decimal %hgt%: Truncate to Integer (%Height%)
```

Then insert the Integer variables in the Text Box Display command.



Left: %Left%
Top: %Top%
Monitor: 0
Width: %Width%
Height: %Height%
☑ Always on Top
☑ Keep Focus

Or mix variables along with the predefined positioning values of Top, Left, Right, Bottom and Center:



Left: Left
Top: Center
Width: %Width%
Height: %Height%

Variable Set from File

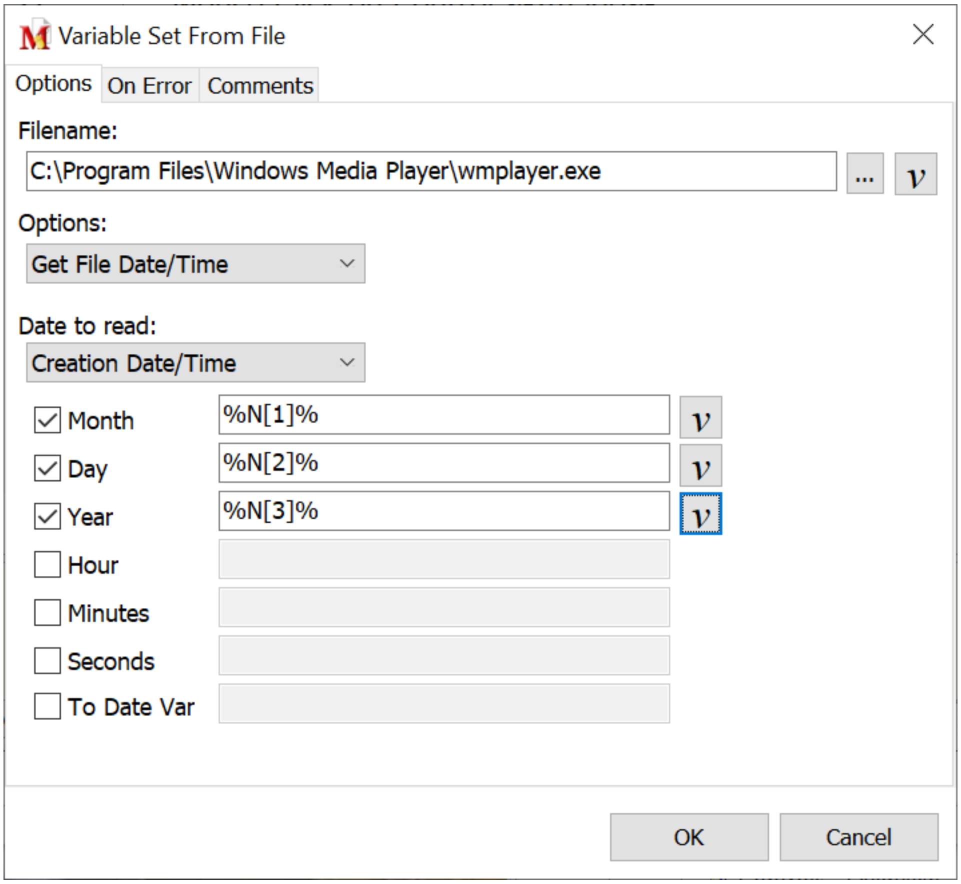
The Variable Set from File command obtains either the date and time of a specific file or the path of the file and saves the information to variables. Compare the variables for possible further action. For example, check to verify if the tested file should be replaced with one that has a more recent date and time.

Get File Date/Time

The Get File Date/Time option obtains the date and time of the file specified and saves the information into Integer variables. There are three different time frames to choose from.

- **Creation Date/Time** - The date and time that the file was originally created.
- **Modified Date/Time** - The date and time that the file was last modified or updated and saved.
- **Last Accessed Date/Time** - The date and time that the file was last accessed or opened.

✓ **Note:** Not every file system supports all three of these date/time formats. Macro Express issues a warning during playback if the format selected is not available.



Filename: C:\Program Files\Windows Media Player\wmplayer.exe

Options: Get File Date/Time

Date to read: Creation Date/Time

☑ Month %N[1]%
☑ Day %N[2]%
☑ Year %N[3]%
□ Hour
□ Minutes
□ Seconds
□ To Date Var

Script Editor > Expand Variables Category > Variable Set from File

The date and time components are broken down to six different variables as described below.

Month/Date/Year

In the example above, the month is saved to Integer variable N[1], the day to N[2] and year to N[3]. So if the file date was December 25, 2020, the value saved to variable N[1] would be 12. The value saved to N[2] would be 25. And the value saved to N[3] would be 2020.

- **Month** - The month is saved to the variable as 1 through 12.
- **Day** - The day is saved to the variable as 1 through 31.
- **Year** - The year is always saved to the variable as a four digit number.

Hour/Minute/Seconds

- **Hour** -The hour is saved in a 24 hour format. If the file time were between midnight and 1:00am, the hour would be saved to the variable as 0. If the file time was between 11:00 am and noon, the hour would be saved to the variable as 11. If the file time was between 11:00 pm and midnight, the hour would be saved to the variable as 23.

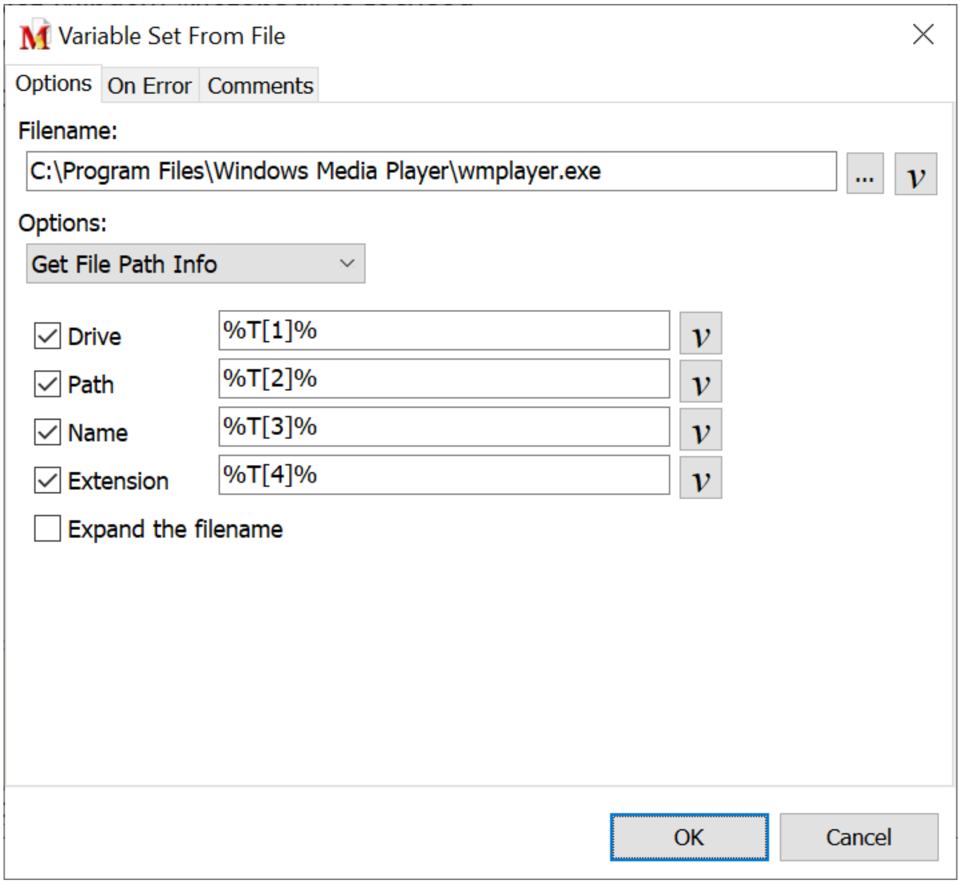
- **Minute** - Minutes are saved to variables as 0 to 59.
- **Seconds** - Seconds are also saved to variables as 0 to 59.

To Date Variable

This option saves all of the above information (month, date, year, hour, minutes, seconds) to a Date Variable. Select the date variable to store the information.

Get File Path Info

The Get File Path option obtains the file path of a particular file and saves the information into text string variables. The file path is split into four distinct units, as explained below.



Filename: C:\Program Files\Windows Media Player\wmplayer.exe

Options: Get File Path Info

☑ Drive %T[1]%

☑ Path %T[2]%

☑ Name %T[3]%

☑ Extension %T[4]%

☐ Expand the filename

Script Editor > Expand Variables Category > Variable Set from File

- Drive - This is the letter drive on the computer, including the colon.

- Path - This variable contains the folder or folders where the file is located.
- Filename - This variable contains just the name of the file without the file extension.
- Extension - This variable contains the file extension, such as .txt, .bmp or .exe.

Example
For the file displayed in the image above, here is what would be saved to each variable.

Drive: **C:**
Path: **\Program Files\Windows Media Player**
Filename: **wmplayer**
Extension: **.exe**

Expand the File Name

The Expand Filename converts a relative file name into a fully qualified path name. For example, if the current directory is c:\Temp and the Filename/Path is "..\work.txt" and Expand Filename is unchecked, then resulting values would be:

Drive:
Path: ..\
Filename: work
Extension: .txt

But if Expand Filename is checked, then the resulting values would be:

Drive: c:
Path: \
Filename: work
Extension: .txt

Get File Size

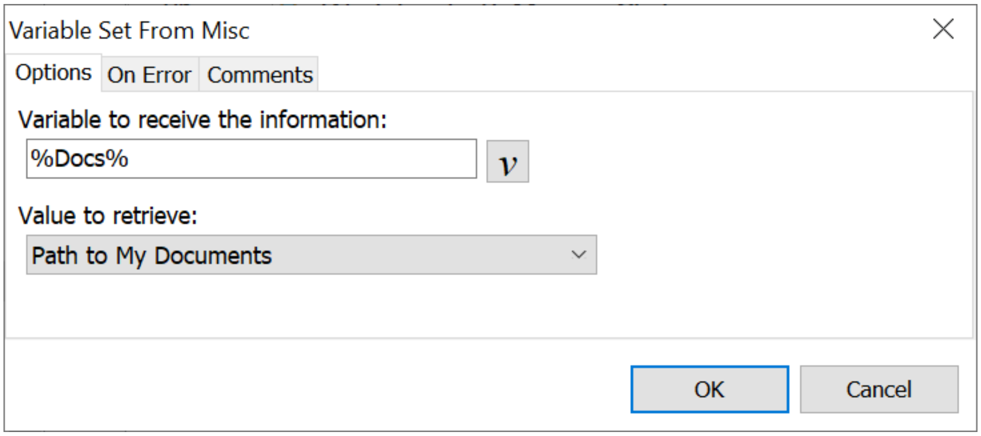
The Get File Size option obtains the size of the file selected and saves it to a variable. The value is saved in bytes.

Store Size In

Enter the variable name to receive the file size.

Variable Set from Miscellaneous

The Variable Set from Miscellaneous command is used to retrieve a variety of information about the computer, the macro file, or Macro Express.



Variable to receive the information: %Docs%

Value to retrieve: Path to My Documents

Script Editor > Expand Variables Category > Variable Set from Miscellaneous

Variable to receive the information

Select the text string variable to store the information retrieved. The variable can be used later in the macro.

Value to retrieve

Select the information needed from the drop down list. This information is stored in the variable chosen.

1. Obtain the path information to various folders on the computer such as My Documents shown in the example above.
2. Obtain various Macro Express information such as the macro file in use, the installation path, where the Macro Express preference settings are stored, etc.
3. Retrieve machine specific information such as owner name, machine name, etc.

List of Values Available

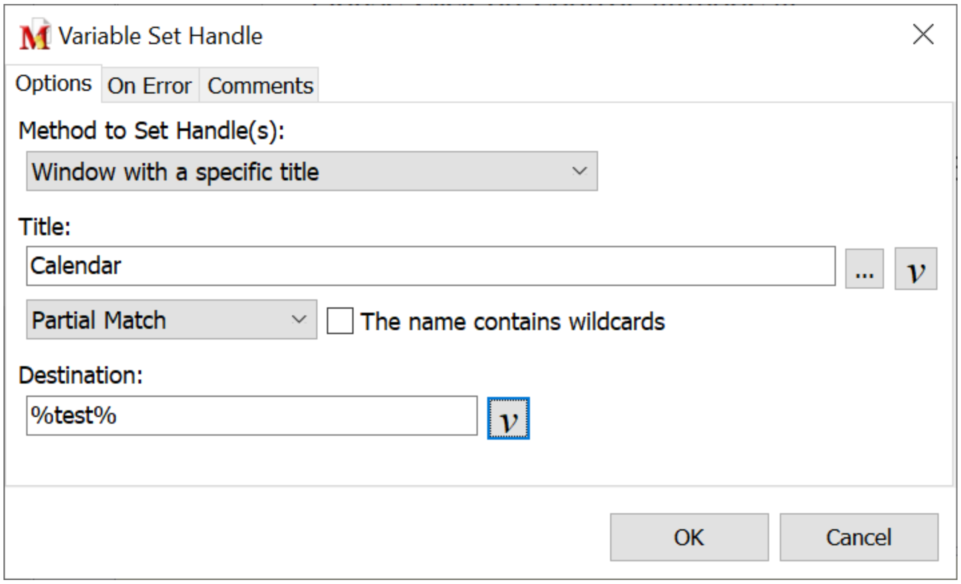
- Path to My Documents
- Path to Start Menu

- Path to Desktop
- Path to Common Documents
- Path to Common Start Menu
- Path to Common Desktop
- Installation Path
- Preferences Registry Key
- Name of Current Macro
- Name of Machine
- User Name
- Registered Owner
- Registered Company

Variable Set Handle

This command is a reliable way to uniquely identify windows by their titles. Even if the window title later changes, this command still identifies the window - as long as the window has not closed.

In the operating system each window receives a handle for unique identification purposes. Use this variable later in other Window commands such as Resize, Reposition, etc.



Method to Set Handle(s): Window with a specific title

Title: Calendar

Partial Match

□ The name contains wildcards

Destination: %test%

Script Editor > Expand Variables Category > Variable Set Handle

Method to Set Handle(s)

Enumerate Windows with a Specific Title

Enumerate Windows with a Specific Title saves any windows that would include for example Microsoft, Macro Express, Word, etc. in the title. Create a variable array to save the Windows to individual variables.

Window with a Specific Title

Window with a Specific Title saves the name of a single window to a handle variable. If more than one running window contains the same Title, the first window found by Macro Express is saved to the variable.

Get from a Control

First define a Control variable using the Get Control or Capture Control commands. Insert the name of the control variable in this field.

Title

Insert the window title or control name in the title field. Or click on the Browse button to select a currently running window.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

✓ **Note:** The Exact/Partial Match and Wildcards are only available with the window options. They are disabled when setting the handle of a control.

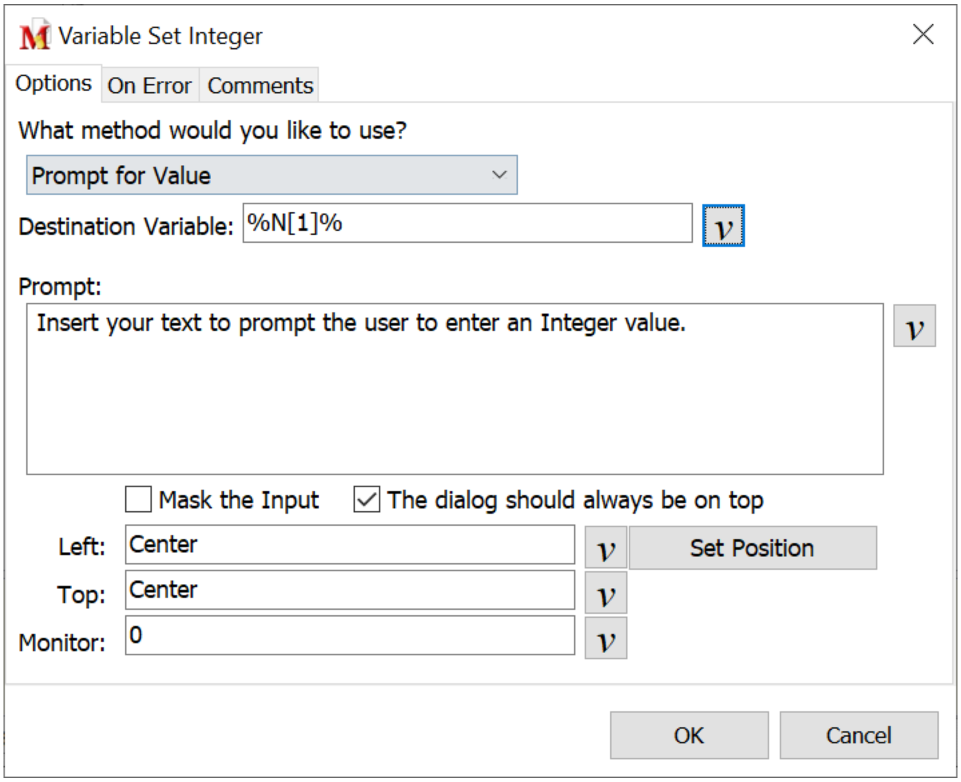
Destination

Insert the variable name to receive the handle information. Or click on the Variable button to select the Handle variable previously defined.

Variable Set Integer

This command assigns a value to an integer variable. Once saved to an integer variable, use the Modify Integer Variable commands or others to manipulate the original values.

- ✓ **Note:** An integer variable (N[1], etc.) can range from −2147483648 to 2147483647. When the value exceeds 2147483647, it ‘wraps’. If an integer variable contains 2147483647 and 1 is added to it, the integer variable will contain −2147483648.



What method would you like to use? Prompt for Value

Destination Variable: %N[1]%

Prompt: Insert your text to prompt the user to enter an Integer value.

☐ Mask the Input ☑ The dialog should always be on top

Left: Center

Top: Center

Monitor: 0

Script Editor > Expand Variables Category > Variable Set Integer

What Method Would You Like to Use?

Select from the drop down list the method required for defining the variable. Each of the options are described below.

Set Value Now

Set Value Now assigns a value to the variable now. Enter the variable value and save. This value will be used when the macro is run.

Destination Variable

Select a variable to use or create a new integer variable.

Initial Value

Insert the contents of the variable in this field.

Prompt for Value

Prompt for Value opens an input box when the macro runs and prompts the user to input an integer value.

Destination Variable

Select a variable to use or create a new integer variable.

Prompt

Optionally enter text in the Prompt edit field to explain the type of value the user should input. The instructions appear above the input field during macro playback.

Mask the Input

The Mask the Input option masks the text entered into the input box and is viewed as *** instead of the actual digits.

The Dialog Should Always be on Top

This options makes sure that the prompt window is always on top of other open windows.

Positioning the Prompt Window

Use this feature to define the position on the screen where the input box will appear during the playback of the macro. Insert exact coordinates or use the Set Position Manually button to set where the window should appear. See the Window Positioning topic for additional information.

Hint: Setting a Default Value in the Prompt Window

First use the Set Value Now option. Insert the integer value to be set as the default. Then use the Prompt for Value option. Both options must reference the same variable, such as N[1]. When the macro is run, the prompt window displays the default value. The user can overwrite, amend, accept or append to this value.

Below is a sample of the commands as they would look in the Scripting Editor. The number 10 is the default value.

```
Variable Set Integer %N[1]% to 10
Variable Set Integer %N[1]% from Prompt
```

Set to the Mouse X Coordinate

Set to the Mouse X Coordinate examines the current mouse position on the screen at macro runtime and places the X pixel coordinate of the mouse position in the variable.

Destination Variable

Select a variable to use or create a new integer variable.

Set to the Mouse Y Coordinate

Set to the Mouse Y Coordinate examines the current mouse position on the screen at macro runtime and places the Y pixel coordinate of the mouse position in the variable.

Destination Variable

Select a variable to use or create a new integer variable.

Set to the Size of a File

Set to the Size of a File examines the size of the file selected. The file size (in bytes) is saved to the variable.

Destination Variable

Select a variable to use or create a new integer variable.

Filename

Insert the name of the file and path to be examined or click on the Browse button to select the file.

Set to a Random Value

Set to a Random Value saves a random number to an Integer variable. Select the range of Integer values. If 5 is selected as the minimum value and 99 as the maximum value, then the possible random number may be any integer from 5 to 99.

Destination Variable

Select a variable to use or create a new integer variable.

Minimum Value

Insert the lowest number that may be included as a possible random number.

Maximum Value

Insert the highest number that may be included as a possible random number.

Set to the Screen

These two options save the width or height of the screen to an integer variable. Select a variable to use or create a new integer variable.

Set to the Screen Width

The integer value is set to the width of the screen. The value assigned is the number of pixels in the width of the screen.

Set to the Screen Height

The integer value will be set to the height of the screen. The value assigned is the number of pixels in the height of the screen.

Set to the Current Window

These four commands examine the current window that has focus and saves values to the variable. Select a variable to use or create a new integer variable.

Set to the Current Window's Top

Set to the Current Window's Top examines the window that is currently on top of all others, specifically its location on the screen. This function obtains the top position (in X,Y screen coordinates) of the window and places the value in the variable. Note: Obtain the bottom coordinate of the window by adding the

Window Top to the Window Height.

Set to the Current Window's Left

Set to the Current Window's Left examines the window that is currently on top of all others, specifically its location on the screen. This function obtains the left position (in X,Y screen coordinates) of the window and places the value in the variable. Note: Obtain the right coordinate of the window by adding the Window Left to the Window Width.

Set to the Current Window's Width

Set to the Current Window's Width examines the size of the window that is currently on top of all other open windows. This function obtains the width (in pixels) of the window and places the value in the variable.

Set to the Current Window's Height

Set to the Current Window's Height examines the size of the window that is currently on top of all other open windows. This function obtains the height (in pixels) of the window and places the value in the variable.

Set to the Clipboard Value

Set to the Clipboard Value examines the text contents of the clipboard. It reads this value and strips off any spaces, carriage returns or line feeds. This function then attempts to convert the value to an integer and place it in the variable. If a non integer value (i.e. a string of text) is in the clipboard or the clipboard is empty, the variable value is set to zero.

Destination Variable

Select a variable to use or create a new integer variable.

Set to the Length of a Text Variable

Set to the Length of a Text Variable counts the number of characters in a text string variable and saves that number to an integer variable. For example, if the length of text string variable T[1] is 100 characters, then the value saved to the Integer variable is 100.

Example

This example shows how to obtain the last 14 characters of a text string when the string is of varying lengths.

For this example, the text string is saved to variable T[1]. The text string could have been copied to the clipboard and then saved to a variable, extracted from a Text File, input by a user, or any number of ways.

1. Variable Set Integer %N[1]% to the length of variable %T[1]%
2. Variable Modify Integer: %N[1]% = %N[1]% - 13
3. Variable Modify String %T[2]% to a substring in %T[1]% starting at %N[1]% and 14 characters long

- Step one calculates the length of the text string variable and saves the value to N[1].
- Step two subtracts 13 characters from the length of the string and saves the new value to N[1].
-The last step copies just the last 14 characters of the string and saves the text to a new value of T[2].

Destination Variable

Select a variable to use or create a new integer variable.

Text Variable

Enter the variable name or select the text variable that is to be examined.

Set to the Position of Text in a Text Variable

Set to the Position of Text in a Text Variable locates specific text in a text string variable and saves the location (position) to an integer variable.

For example, if variable T[1] contained "myemailaddress@myisp.com", this command could find the location of the @ symbol in the string. The integer variable would be set to 15 as @ is the 15th character. This information could then be used to delete or copy all of the text found before the @ symbol, or after.

✓ **Note:** If the text string cannot be found during playback of the macro, the position is set to (0) zero.

Also look at the Split String command for another text string manipulation option.

Example

This example shows how to save all characters before the @ symbol to a new text string variable when the string is of varying lengths. If the text string was always the same length, you would just use the Copy Part of Text option from the Variable Modify String command.

For this example, the text string is saved to variable T[1]. The text string could have been copied to the clipboard and then saved to a variable, extracted from a Text File, input by a user, or any number of ways.

1. Variable Set Integer %N[1]% to the position of "@" in %T[1]%
2. Variable Modify Integer: %N[1]% = %N[1]% - 1
3. Variable Modify String %T[2]% to a substring in %T[1]% starting at 1 and %N[1]% characters long

-Step one calculates the position of the @ symbol in the text string variable and saves the value to N[1].
-Step two subtracts one character from the length and saves the new value to N[1].
-The last step copies all of the characters before the @ symbol and saves the text to a new value of T[2].

Destination Variable

Select a variable to use or create a new integer variable.

Text Variable

Enter the variable name or select the text variable that is to be examined.

Text to Search For

Enter the text that you want to find in the variable you have selected.

Ignore Case Sensitivity

Select this option if you want to ignore upper or lower case letters in the string.

Set Value to Date or Time

These seven Date or Time functions set a value to the variable based on the date or time of the computer's system clock. If the system clock is incorrect, the integer values will also be incorrect.

An example of using one of these options would be the Set Value From Current Hour. The macro could play one set of instructions if the system clock reads before 5:00pm or play back another set of instructions if the system clock reads after 5:00pm.

Select a variable to use or create a new integer variable.

Set to the Current Day

Set to the Current Day sets the integer value based on the current date. A value of 1 to 31 is saved to the variable.

Set to the Current Month

Set to the Current Month sets the integer value based on the current month. A value 1 to 12 is saved to the variable.

Set to the Current Year

Set to the Current Year sets the integer value based on the current year. The value is saved as a four digit number.

Set to the Current Day of Week

Set to the Current Day of Week sets the integer value based on the current day of the week. The day of the week is an integer between 1 and 7. Sunday is the first day (i.e. 1) of the week and Saturday is the seventh (i.e. 7).

Set to the Current Hour

Set to the Current Hour sets the integer value based on the current hour. The hours are set in military time. For example, 5:00pm would be set as 17:00, or 8:00am would be set as 8:00. The integer variable is saved as 17 for 5:00pm or 8 for 8:00am.

Set to the Current Minute

Set to the Current Minute sets the integer value based on the current minute. A value of 0 to 59 is saved to the variable.

Set to the Current Second

Set to the Current Second sets the integer value based on the current second. A value of 0 to 59 is saved to the variable.

Set to Control

These four commands are similar to the Set Value to the Current Window commands. But rather than look at the current window, these commands set the value from a specific control.

Controls refer to the identifiable parts of a specific application window. Controls may include such things as a button, an edit box, a toolbar, a window title bar or other items. The control may be part of the main application window or be part of a child window (a window within a window). Depending on how the application was written, the entire window may be a single control.

Set to a Control's Left

Set to a Control's Left examines the location on the screen of the selected Control. This function obtains the left position (in X,Y screen coordinates) of the Control and places the value in the variable.

- ✓ **Note:** Find the right position of the Control by adding the Control Left to the Control Width.

Set to a Control's Top

Set to a Control's Top examines the location on the screen of the selected Control. This function obtains the top position (in X,Y screen coordinates) of the Control and places the value in the variable.

- ✓ **Note:** Find the bottom position of the Control by adding the Control Top to the Control Height.

Set to a Control's Width

Set to a Control's Width examines the size on the screen of the selected Control. This function obtains the width (in pixels) of the Control and places the value in the variable.

Set to a Control's Height

Set to a Control's Height examines the size on the screen of the selected Control. The function obtains the height (in pixels) of the Control and places the value in the variable.

Computer Monitor

These monitor options obtain the width or height of a monitor in pixels, as well as determine the active monitor or the number of monitors in use. Then use the If Variable command to take specific actions based on the variable results.

Set to a Monitor's Width

Save the width of the monitor in pixels to an Integer variable. If using a single monitor, then insert 0 (zero) as the monitor number. If multiple monitors are connected to the system, then insert 0 as the first monitor, 1 (one) as the second monitor, 2 (two) as the third monitor, etc.

Set to a Monitor's Height

Save the height of the monitor in pixels to an Integer variable. If using a single monitor, then insert 0 (zero) as the monitor number. If multiple monitors are connected to the system, then insert 0 (zero) as the first monitor, 1 (one) as the second monitor, 2 (two) as the third monitor, etc.

Set to Number of Monitors

Determine the number of monitors attached to the system and save the value to an Integer variable. So, for example, if there are three monitors attached to the computer, then the Integer variable value would be 3.

Set to Active Monitor

This option saves the value of the active monitor (whichever monitor has focus) to an Integer variable. The values are similar to the monitor numbers mentioned above. The first monitor receives a value of 0 (zero). A second monitor receives a value of 1 (one), the third monitor 2 (two), and so on.

Get the Idle Time

Get the Idle Time calculates the computer idle time and saves the value to a variable. The idle time is specified as no mouse or keyboard activity occurring. The amount of time since the last mouse event or keyboard activity is saved to the Integer variable. The value is saved in time increments of seconds.

Desktop

These two desktop options obtain the width or height in pixels of the entire desktop, which may span more than one monitor. Macro Express retrieves the desktop size from what Windows reports as the desktop.

Set to Desktop Width

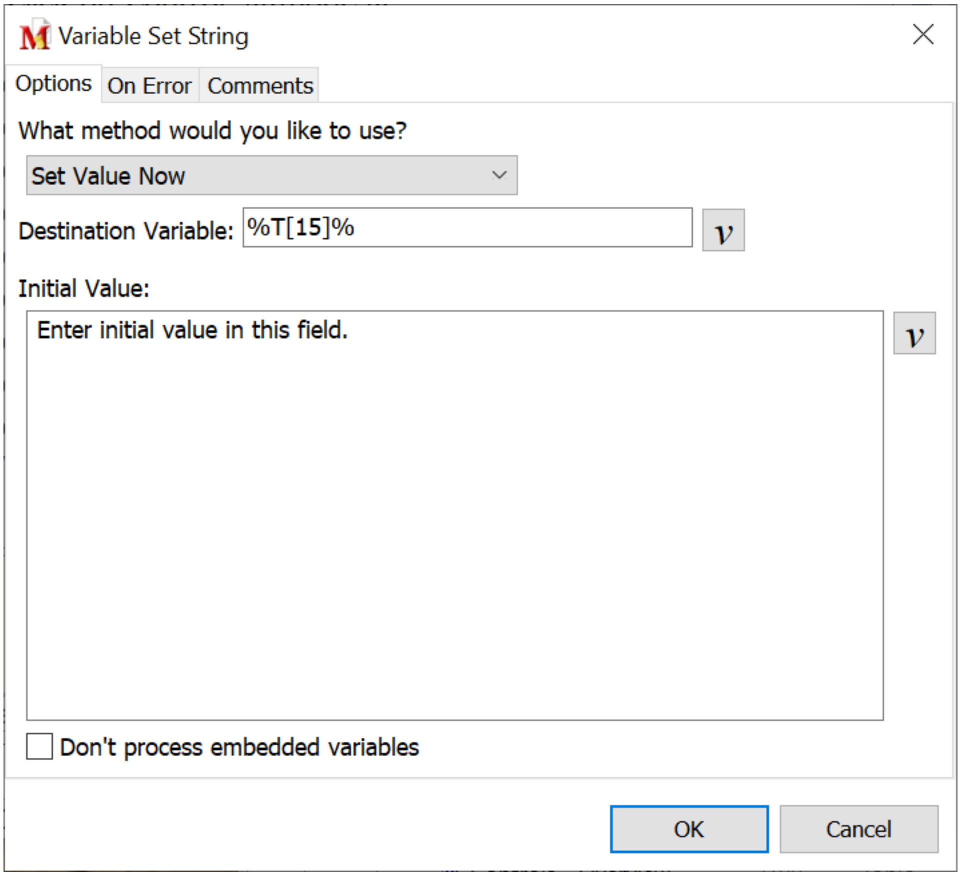
Save the width of the desktop in pixels to an Integer variable.

Set to Desktop Height

Save the height of the desktop in pixels to an Integer variable.

Variable Set String

The Variable Set String command assigns a value to a text string variable. Once saved to a string variable value, use the Modify String Variables command to manipulate the original values assigned in the macro.



What method would you like to use?

Set Value Now

Destination Variable: %T[15]%

Initial Value:

Enter initial value in this field.

□ Don't process embedded variables

Script Editor > Expand Variables Category > Variable Set String

Don't process embedded variables

This option is for use if the initial value text contains variables or a combination of % symbols. Select this option to have the macro ignore any variables embedded in the Initial Value field.

For example, select %T[15]% as the destination variable and insert %sample% in the Initial Value field. When the "Don't process embedded variables" option is selected %sample% will be stored in variable %T[15]% as %sample% rather than the expanded value of the variable.

This feature is available for the following Variable Set String options: Set Value Now, Set Value from Clipboard, Set Value from File, Set Value from INI File, Set from Environment Variable and Set Value from List.

Set Value Now

Set Value Now assigns a value to the variable now, which will be used later in the macro.

Destination Variable

Select a variable to use or create a new string variable.

Initial Value

Insert the contents of the variable in this field.

Prompt for Value

Prompt for Value displays an input box during macro playback prompting the user to input a text string.

Destination Variable

Select a variable to use or create a new string variable.

Prompt

Enter text in the Prompt edit field to explain the type of value the user should input. During macro playback, the instructions appear above the input field.

Mask the Input

The Mask the Input option masks the text entered into the input box and is viewed as *** instead of the actual digits.

The Dialog Should Always be on Top

This options makes sure that the prompt window is always on top of other open windows.

Positioning the Prompt Window

Use this feature to define the position on the screen where the input box will appear during the playback of the macro. Insert exact coordinates or use the Set Position Manually button to set where the window should appear. See the Window Positioning topic for additional information.

Hint: Setting a Default Value in the Prompt Window

First use the Set Value Now option. Insert the text to be set as the default. Then use the Prompt for Value option. Both options must reference the same variable, such as T[1]. When the macro is run, the prompt window displays the default value. The user can overwrite, amend, accept or append to this value.

Below is a sample of the commands as they would look in the Scripting Editor. "This is a test" is the default value.

```
Variable Set String %T[1]% "This is a test"
Variable Set String %T[1]% from Prompt
```

Set Value from Clipboard

Set Value from Clipboard extracts the text currently saved to the clipboard and places it into the variable.

Destination Variable

Select a variable to use or create a new string variable.

Set Value from File

Set Value from File loads the contents of a text file into a variable.

Destination Variable

Select a variable to use or create a new string variable into which the file contents will be loaded.

Filename

Insert the file name and path into this field or click on the Browse button to select the file to be used.

Strip the Trailing CR/LF Characters

Use this option to save the text file contents to a variable with all trailing carriage returns or line feeds removed. This does not remove all CR/LF's from the variable, just any that trail the end of the text. Use the Variable Modify String - Strip CR/LF option to remove all carriage returns or line feeds from a text variable.

Set Value to Top Most Program Name

Set Value to Top Most Program Name examines the window that is currently on top of all others and determines what program it is from. It places the program name (i.e. the EXE file name) into the variable.

Destination Variable

Select a variable to use or create a new string variable.

Set Value to Top Most Window Title

Set Value to Top Most Window Title examines the window that is currently on top of all others, extracts the name that is in the caption bar (i.e. the Window Title) and saves it to the variable.

Destination Variable

Select a variable to use or create a new string variable.

Set Value to Current Folder Name

Set Value to Current Folder Name obtains the folder name that Windows is currently pointing to and saves the name to the variable.

Destination Variable

Select a variable to use or create a new string variable.

Set Value From INI File

Set Value From INI File obtains a value from an INI file and places it into a variable. Specify the file name of the INI file, the section name and the key name.

Destination Variable

Select a variable to use or create a new string variable.

Filename

Insert the name and path of the INI file in this field or browse to the file location.

Key - Insert the Key name in this field.
Value - Insert the Value name in this field.

Prompt for Filename

Prompt for Filename prompts the user to select a file during macro runtime by displaying a standard "Find File" dialog box. The file name and path chosen by the user are saved to the variable.

Destination Variable

Select a variable to use or create a new string variable.

Titlebar

Name the window title of the dialog box that opens. This may include instructions in the title bar for users to follow.

> *Hint: Starting the Prompt in a Predetermined Folder*
>
> When prompting for a file you may want to start the search from a specific folder, rather than starting at Computer, in order to save time. To do this, use the Variable Set String - Set Value Now command and save the starting folder path and a file in that folder to a variable. Use this variable in the Destination Variable field of the Prompt for Filename command.
>
> ```
> Variable Set String %T[1]% to "C:\Users\Fred\Documents\Insight Software\Testing\Sample.txt"
> Variable Set String %T[1]%: Prompt for a filename
> ```
>
> In the example above, the search prompt will start in the Testing folder to save time clicking through several other folders to arrive at the starting point you want.

Prompt for Foldername

Prompt for Foldername prompts the user with a standard "Browse for Folder" dialog box during macro playback. The folder name and path selected by the user are saved to the variable. If that variable is predefined, Macro Express looks to make sure that there are no invalid values in the variable string. The value will only be saved up to the invalid character in the string.

Destination Variable

Select a variable to use or create a new string variable.

Titlebar

Name the window title of the dialog box that opens. This may include instructions in the title bar for users to follow.

Hint: Starting the Prompt in a Predetermined Folder

When prompting for a folder you may want to start the search from a specific folder, rather than starting at Computer, in order to save time. To do this, use the Variable Set String - Set Value Now command and save the starting folder path to a variable. Use this variable in the Destination Variable field of the Prompt for Foldername command.

```
Variable Set String %Folder[1]% to
"C:\Users\Howard\Documents\Insight Software Solutions\Macro Express"
Variable Set String %Folder[1]%: Prompt for a foldername
```

In the example above, the search prompt will start in the Macro Express folder to save time clicking through several other folders to arrive at the starting point you want.

Set to a Random Letter

Set to a Random Letter randomly selects a letter of the alphabet and saves the result to a variable.

Destination Variable

Select a variable to use or create a new string variable.

Set from Environment Variable

Set from Environment Variable obtains a value from an environmental variable and places it into a Text String variable. The complementary command to write a text value to an environmental command may be found in the Variable Modify String window.

Destination Variable

Select a variable to use or create a new string variable.

Environment Variable

Select the Environmental Variable by clicking on the Variables button or type the variable name in the edit field.

Additional information about Environmental Variables from Microsoft

To determine which Environment variables exist on your computer, run the Command Prompt. Type set in the Command Prompt window and press Enter to view the Environment variables.

--- Definition #1 ---

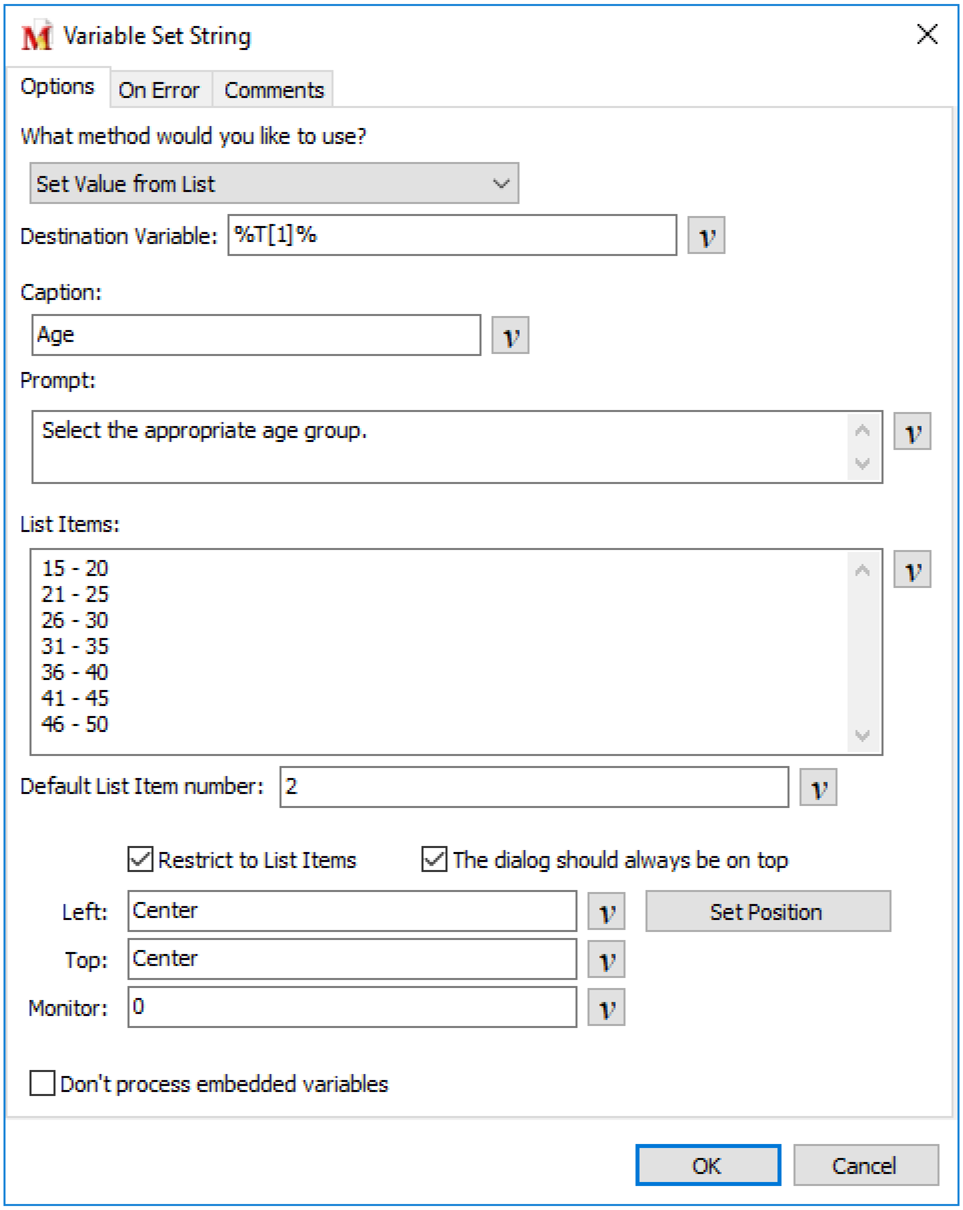
Environment variable
An element of the operating system environment, such as a path, a directory name, or a configuration string. Environment variables are typically set within batch files.

--- Definition #2 ---

Environment variables specify search paths for files, directories for temporary files, application-specific options, and other similar information. The system maintains an environment block for each user and one for the computer. The system environment block represents environment variables for all users of the particular computer. A user's environment block represents the environment variables the system maintains for that particular user, including the set of system environment variables.

Set Value from List

This command is similar to the Variable Set from Prompt command, but with the added feature of providing a list of options for the user to choose from. The choice selected is then saved to a text string variable.



Variable Set String

What method would you like to use?

Set Value from List

Destination Variable: %T[1]%

Caption:

Age

Prompt:

Select the appropriate age group.

List Items:

15 - 20
21 - 25
26 - 30
31 - 35
36 - 40
41 - 45
46 - 50

Default List Item number: 2

☑ Restrict to List Items ☑ The dialog should always be on top

Left: Center

Top: Center

Monitor: 0

☐ Don't process embedded variables

Destination Variable

Select a variable to use or create a new string variable into which to save the selection made.

Caption

The text that is displayed in the header of the dialog window that opens during macro playback.

Prompt

Provide instructions as to how to select the appropriate choice from the list.

List Items

Create the list of choices that the user may choose from. The choice selected is saved to the text string variable defined.

Default List Item number

Optionally set one of the choices as the default selection in the list. Just enter the number of the line to be the default. In the image above, the second choice of 21 - 25 is set as the default. Leave the field blank to not set a default choice.

Restrict to List Items

Select this option if only one of the list items may be used in the macro. If this option is not selected, then the user has the option to type in a response, rather than just picking one of the drop down choices.

The dialog should always be on top

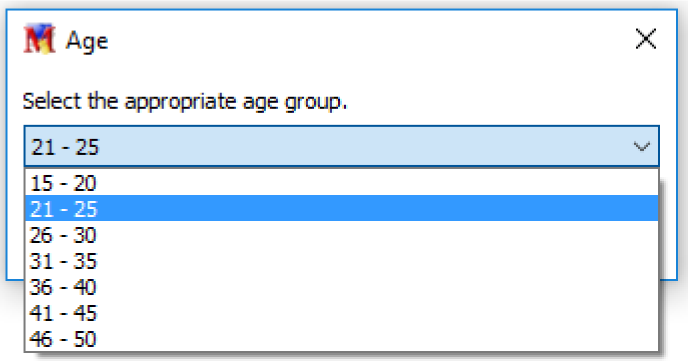
Select this option so that the prompt window displays on top of other windows.

Set Position

Use this feature to define the position on the screen where the input box will appear during the playback of the macro. Insert exact coordinates or use the Set Position button to set where the window should display. See the Window Positioning topic for additional information.

Sample image

Below is an example of how the Set Value from List dialog is displayed when the macro runs.



Age

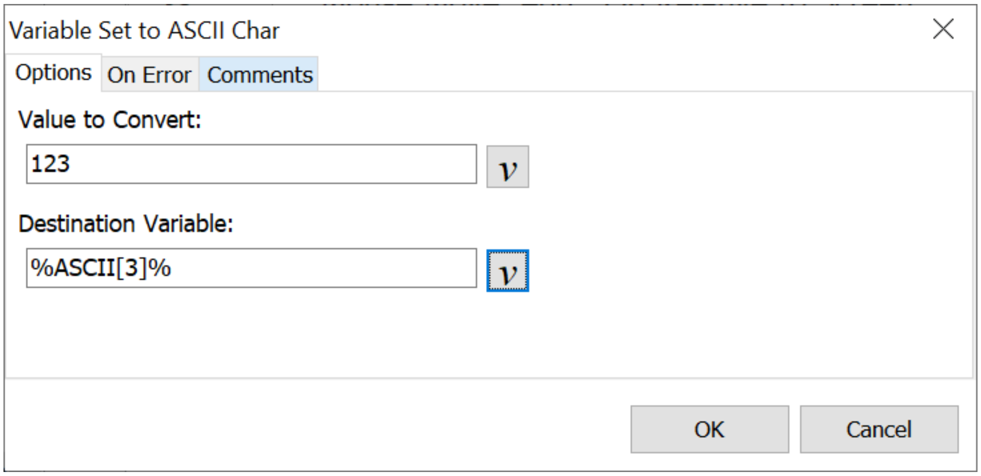
Select the appropriate age group.

21 - 25

15 - 20
21 - 25
26 - 30
31 - 35
36 - 40
41 - 45
46 - 50

Variable Set to ASCII Char

The Variable Set to ASCII Char command converts a numeric value into its ASCII character equivalent. For example the number 65 represents the letter A. Or in the example below, the number 123 represents the character {. The range of values allowed is from 0 to 255.



Value to Convert: 123

Destination Variable: %ASCII[3]%

Script Editor > Expand Variables Category > Variable Set to ASCII Char

Value to Convert

Insert the numeric value to be converted to an ASCII character. Or insert a variable that contains the numeric value to be converted.

Destination Variable

The ASCII character that matches the numeric value is saved to a text string variable. Enter the variable name or click on the Variables button to select a variable for use.

Chart of ASCII Characters 0 - 127

For convenience, the list below displays the numeric value and the ASCII character represented by that number. Values 0 - 31 are non-printable characters, such as line feeds, tabs, etc. Extended ASCII codes 128 - 255 may be found with a quick online search.

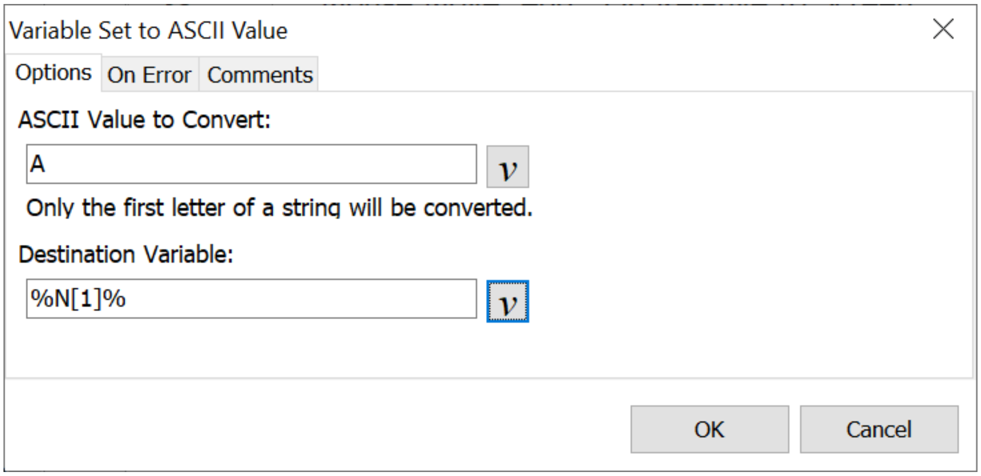
#	Description	#	Char	#	Char	#	Char

0	Null char	32	Space	64	@	96	`
1	Start of Heading	33	!	65	A	97	a
2	Start of Text	34	"	66	B	98	b
3	End of Text	35	#	67	C	99	c
4	End of Transmission	36	$	68	D	100	d
5	Enquiry	37	%	69	E	101	e
6	Acknowledgment	38	&	70	F	102	f
7	Bell	39	'	71	G	103	g
8	Back Space	40	(	72	H	104	h
9	Horizontal Tab	41	)	73	I	105	i
10	Line Feed	42	*	74	J	106	j
11	Vertical Tab	43	+	75	K	107	k
12	Form Feed	44	,	76	L	108	l
13	Carriage Return	45	-	77	M	109	m
14	Shift Out	46	.	78	N	110	n
15	Shift In	47	/	79	O	111	o
16	Data Line Escape	48	0	80	P	112	p
17	Device Control 1	49	1	81	Q	113	q
18	Device Control 2	50	2	82	R	114	r
19	Device Control 3	51	3	83	S	115	s
20	Device Control 4	52	4	84	T	116	t
21	Negative Acknowledgement	53	5	85	U	117	u
22	Synchronous Idle	54	6	86	V	118	v
23	End of Transmission Block	55	7	87	W	119	w
24	Cancel	56	8	88	X	120	x
25	End of Medium	57	9	89	Y	121	y
26	Substitute	58	:	90	Z	122	z
27	Escape	59	;	91	[	123	{
28	File Separator	60	<	92	\	124	|
29	Group Separator	61	=	93	]	125	}

30	Record Separator	62	>	94	^	126	~
31	Unit Separator	63	?	95	_	127	Delete

Variable Set to ASCII Value

The Variable Set to ASCII Value command is essentially the inverse of the Variable Set to ASCII Character. It converts an ASCII character to its ASCII numeric equivalent. For example, the letter "a" converts to a numeric value of 97. The letter "A" converts to a numeric value of 65. The numeric value is saved to an Integer variable that can be used later in the macro.



ASCII Value to Convert: A

Only the first letter of a string will be converted.

Destination Variable: %N[1]%

Script Editor > Expand Variables Category > Variable Set to ASCII Value

ASCII Value to Convert

Enter the ASCII character to be converted to its ASCII numeric equivalent. Or insert the variable that contains the ASCII character to be converted.

Destination Variable

The ASCII character selected is saved as a numeric value and placed in an integer variable. In the example above the ASCII character "A" is saved to the variable selected, which in this case is N[1].

Chart of ASCII Characters 0 - 127

#	Description	#	Char	#	Char	#	Char
0	Null char	32	Space	64	@	96	`
1	Start of Heading	33	!	65	A	97	a
2	Start of Text	34	"	66	B	98	b

3	End of Text	35	#	67	C	99	c
4	End of Transmission	36	$	68	D	100	d
5	Enquiry	37	%	69	E	101	e
6	Acknowledgment	38	&	70	F	102	f
7	Bell	39	'	71	G	103	g
8	Back Space	40	(	72	H	104	h
9	Horizontal Tab	41	)	73	I	105	i
10	Line Feed	42	*	74	J	106	j
11	Vertical Tab	43	+	75	K	107	k
12	Form Feed	44	,	76	L	108	l
13	Carriage Return	45	-	77	M	109	m
14	Shift Out	46	.	78	N	110	n
15	Shift In	47	/	79	O	111	o
16	Data Line Escape	48	0	80	P	112	p
17	Device Control 1	49	1	81	Q	113	q
18	Device Control 2	50	2	82	R	114	r
19	Device Control 3	51	3	83	S	115	s
20	Device Control 4	52	4	84	T	116	t
21	Negative Acknowledgement	53	5	85	U	117	u
22	Synchronous Idle	54	6	86	V	118	v
23	End of Transmission Block	55	7	87	W	119	w
24	Cancel	56	8	88	X	120	x
25	End of Medium	57	9	89	Y	121	y
26	Substitute	58	:	90	Z	122	z
27	Escape	59	;	91	[	123	{
28	File Separator	60	<	92	\	124	|
29	Group Separator	61	=	93	]	125	}
30	Record Separator	62	>	94	^	126	~
31	Unit Separator	63	?	95	_	127	Delete

Controls - Overview

The use of the Control commands can be a very powerful addition to macro building. We will attempt to explain some of the benefits of using these commands in this page and subsequent topics. But first, here is a definition of controls from Microsoft:

> "A control is a child window an application uses in conjunction with another window to perform simple input and output (I/O) tasks. Controls are most often used within dialog boxes, but they can also be used in other windows. Controls within dialog boxes provide the user with the means to type text, choose options, and direct a dialog box to complete its action. Controls in other windows provide a variety of services, such as letting the user choose commands, view status, and view and edit text."

For our purposes, controls refer to the identifiable parts of a specific application window. Controls may include such things as a button, an edit box, a toolbar, a window title bar or other items. The control may be part of the main application window or be part of a child window (a window within a window).

Assigning Controls to Variables

A Control type variable contains the control information. There are two commands that can be used to obtain the control information and assign it to a Control variable.

> The first choice would be to use the Get Control command. This launches a small utility and a pointer that is dragged over top of a control. The control information is captured and assigned to a variable. In this case the control information is assigned while building the macro.
>
> The other option is to use the Capture Control command. The control information is captured during the playback of a macro. The information is taken from the control sitting underneath the mouse cursor at the time the command is run.

Manipulating the Variable

After the information has been saved to a Control type variable, there are a number of options available for using this information.

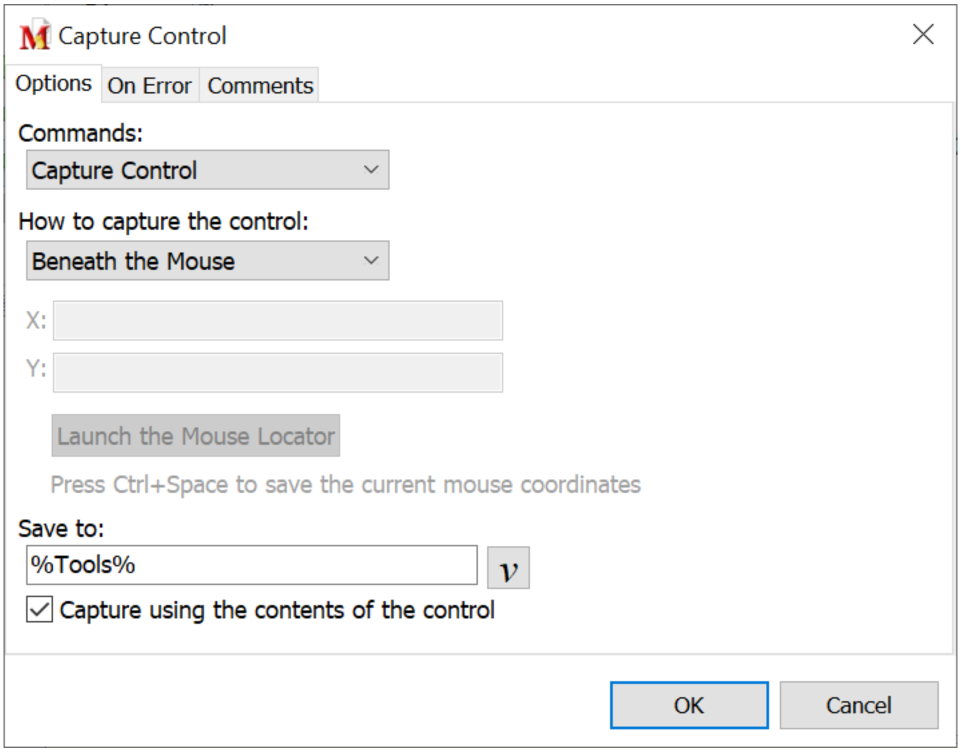
If Statements	Set up conditional If Control statements, such as if the Control is visible, has focus or is enabled. Or base the statement on if the control is not visible, doesn't have focus or is disabled.
Mouse Click	Have the mouse click on the control at some point during the macro. This is a good way to make sure the correct button is pressed, the right edit field has focus, etc.
Set Focus	Set focus to the control during playback of the macro.
Set Control Text	This Set Control Text option defines the text for the control variable. This is helpful when the captured control is an edit box containing text.
Modify the Controls	The Variable Modify Controls options allow for copying the variable to another variable or to modify the top level window for more macro flexibility.
Post Message	The Post Message command allows the user to send a Windows message to a specific control.
Send Text Directly to a Control	The Text Type command includes a Send Text Directly to Control option. The text is inserted directly to the control (if the control selected accepts text input). The control must be running on the computer, but focus to the control is not required for the text to be inserted.
Wait for Control	Instruct a macro to wait for the control to gain focus, become visible or become enabled. Or cause the macro to wait for their opposites - wait for control to lose focus, become invisible or become disabled.

In summary, the controls open up many more possibilities for creating macros. Use them to more reliably work with Child Windows (windows within windows). Directly click on a button or go to a specific edit box without having to Tab through fields or position the mouse just right.

✓ **Note:** Not every button, menu item or component on a program is actually a Control. Sometimes the only control in a given program is the window or dialog that pops up when the program starts. How the controls work depends on how the application was written.

Capture Control

The Capture Control command saves the Window Control information to a variable. The control information is captured during playback of the macro. Use the Get Control command to obtain the control information during the creation of the macro.



Commands: Capture Control

How to capture the control: Beneath the Mouse

X:

Y:

Press Ctrl+Space to save the current mouse coordinates

Save to: %Tools%

☑ Capture using the contents of the control

Script Editor > Expand Window Controls Category > Capture Control

How to Capture the Control

Select to capture the control that currently has focus, is underneath the mouse cursor or at a specific location on the screen.

Beneath Mouse

During playback of the macro, this command determines the location of the mouse cursor. The control located beneath the mouse cursor is captured and saved to the variable in the "Save to" field.

Focused Control

During playback of the macro, this command determines which control has focus. The control information is captured and saved to the variable you specify.

Specific Coordinates

During playback of the macro, the mouse moves to the X and Y coordinates of the screen. The X coordinate represents the number of pixels from the left edge of the screen. The Y coordinate represents the number of pixels the mouse is positioned from the top of the screen.

If necessary, click on the Launch the Mouse Locator button to find the screen coordinates. The control that exists beneath these coordinates is captured. This control information is saved to the variable.

Save To

Save the control to a variable. Select the control variable to store the information.

Capture Using the Contents of the Control

This option makes the capture of a control more exact by also capturing the text found in the control. So for example, assume capturing an edit box with the word Test in it. During playback the macro looks for the edit box and verifies that it contains the word Test. If the edit box text is changed to something else, such as Reset or left blank, then the control that was captured, such as C[1], will not match the actual control in the window - edit box.

✓ **Note:** Not every button, menu item or component on a program is actually a Control. Sometimes the only control in a given program is the window or dialog that pops up when the program starts. We have also noticed that many of Microsoft's programs seem to have fewer controls than programs from other manufacturers. How the controls work depends on how the program was written.

Clear Control Cache

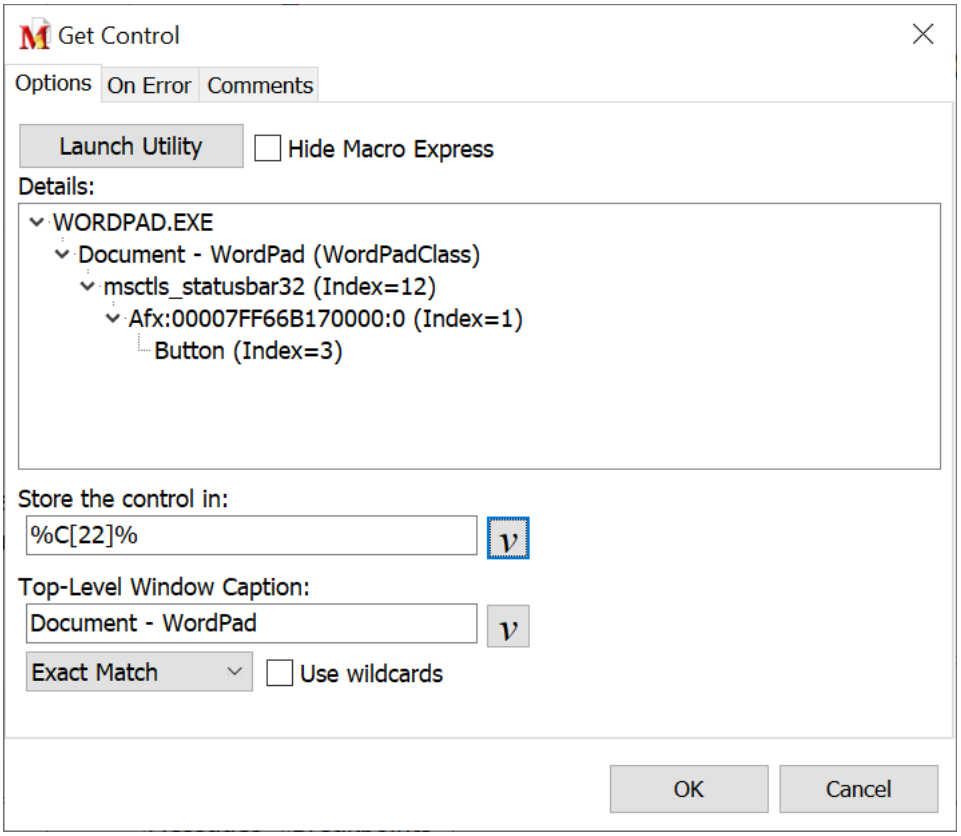
The Clear Control Cache command is inserted directly into the script as there is not an associated edit window. From the Script Editor click on the Window Controls category of commands to expand the category. Select Clear Cache Control to insert into the script.

When a control is used in Macro Express, the control data is cached. The information cached is used to help the Window Controls commands function more reliably.

The Clear Control Cache command clears the information that is being cached.

Get Control

The Get Control command saves the control information to a variable. A capture utility is launched to select buttons, edit boxes, etc. that can be used later in the macro. Use the Capture Control command to obtain the control information during playback of the macro.



□ Hide Macro Express

Details:

```
WORDPAD.EXE
  Document - WordPad (WordPadClass)
    msctls_statusbar32 (Index=12)
      Afx:00007FF66B170000:0 (Index=1)
        Button (Index=3)
```

Store the control in: %C[22]%

Top-Level Window Caption: Document - WordPad

Exact Match

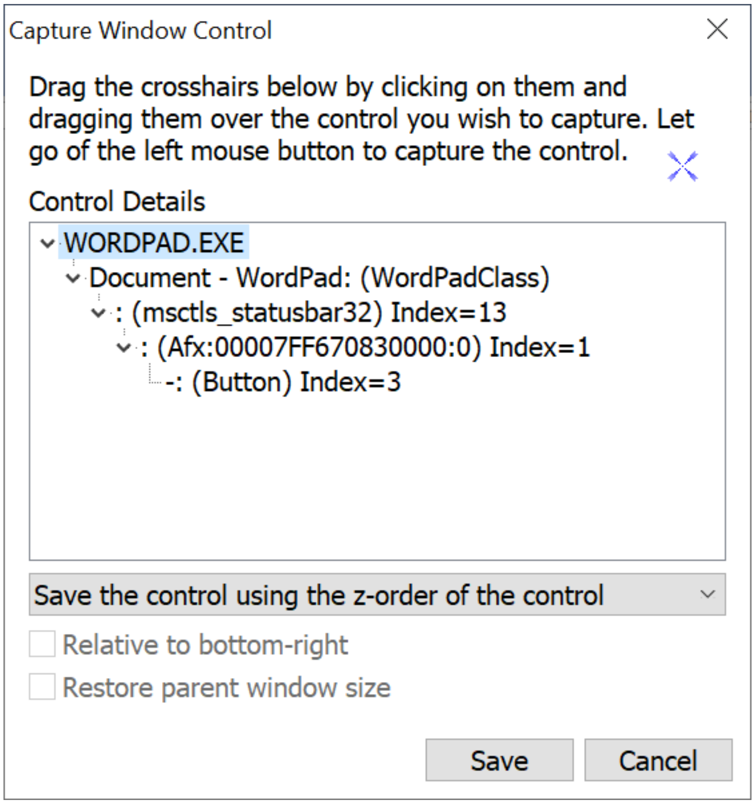
□ Use wildcards

Script Editor > Expand Window Controls Category > Get Control

Launch Utility

Below is an image of the window that appears when launching the control utility. Place the mouse cursor on the crosshair image, hold down the left mouse button and drag the mouse over the control needed. Drag it over title bars, buttons, edit boxes, forms, toolbars, etc.

After selecting the control, release the left mouse button. The Control Details edit field is populated with the control information. Click on the Save button to return to the Get Control Window. The control information is saved in the Get Control Window.


Capture Window Control

Drag the crosshairs below by clicking on them and dragging them over the control you wish to capture. Let go of the left mouse button to capture the control.

Control Details

- WORDPAD.EXE
  - Document - WordPad: (WordPadClass)
    - : (msctls_statusbar32) Index=13
      - : (Afx:00007FF670830000:0) Index=1
        - -: (Button) Index=3

Save the control using the z-order of the control

□ Relative to bottom-right

□ Restore parent window size

Expand Window Controls Category > Get Control > Launch Utility Button

Save the Control Using the Z-Order of the Control

This method saves the control information based on the order in which the controls are created or manipulated by the parent program. In some applications, the z-order may change without notice or the user being able to determine that it has changed.

Save the Control Using the Contents of the Control

This option makes the capture of a control more exact by also capturing the text found in the control. So for example, assume capturing an edit box with the word Test in it. During playback the macro looks for the edit box and verifies that it contains the word Test. If the edit box text is changed to something else, such as Reset or left blank, then the control that was captured, such as C[1], will not match the actual control in the window - edit box.

Save the Control Using its Coordinates

There are some controls that change information each time the window with the control is opened or the program with the control is restarted as mentioned above. This renders the information saved to the control obsolete and the control will not be found the next time the macro is run. This Coordinates option saves the control information based on the location of the control in the window.

Relative to the Bottom Right

Typically the coordinates of the control are measured from the top, left corner of the window that contains the control. Use this alternative method to obtain the position of the control from the bottom, right corner of the window.

Restore Parent Window Size

This option restores the size of the parent window (the window that contains the control) to the size of the window when the original capture of the control was made.

Hide Macro Express

Hide Macro Express does what it implies. The program hides or gets out of the way after pressing the Launch Get Control Utility. This makes it easier to locate the application containing the control to capture.

Store the Control In

Select the variable to store the control name and information.

Top Level Window Caption

The Top Level Window Caption contains the text that appears in the Window Title of the application where the control was captured.

Partial Match

A partial match allows the Control variable to find a Window Title that partially matches the name defined in the control.

Exact Match

An exact match requires that the Control variable find the Window Title that matches exactly - letter for letter.

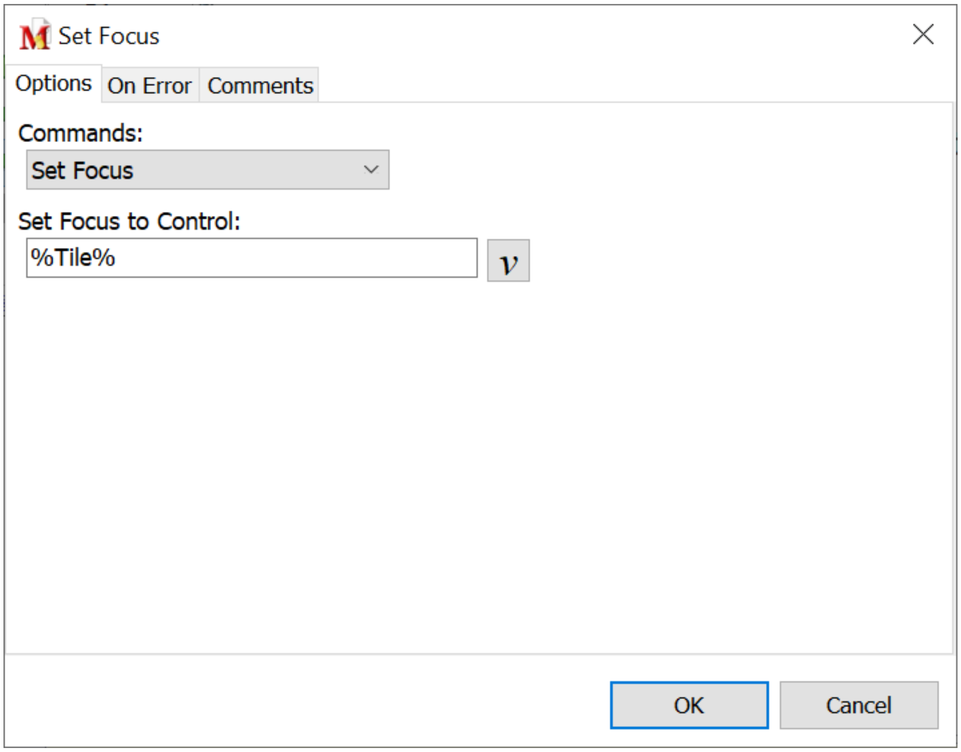
Use Wildcards

Select this option if using a wildcard in the window title. A wildcard would include a * or ??? characters in the title to expand the possibilities of a match.

✓ **Note:** Not every button, menu item or component on a program is actually a Control. Sometimes the only control in a given program is the window or dialog that pops up when the program starts. We have also noticed that many of Microsoft's programs seem to have fewer controls than programs from other manufacturers. How the controls work depends on how the program was written.

Set Focus

Set Focus is similar to the Window Activate command, but uses a control instead. It brings the window that contains the selected control to the front and then sets focus to the control. This is most useful for edit box controls. Set focus to the edit box and then use the Text Type command to insert text, etc.



Commands: Set Focus

Set Focus to Control: %Tile%

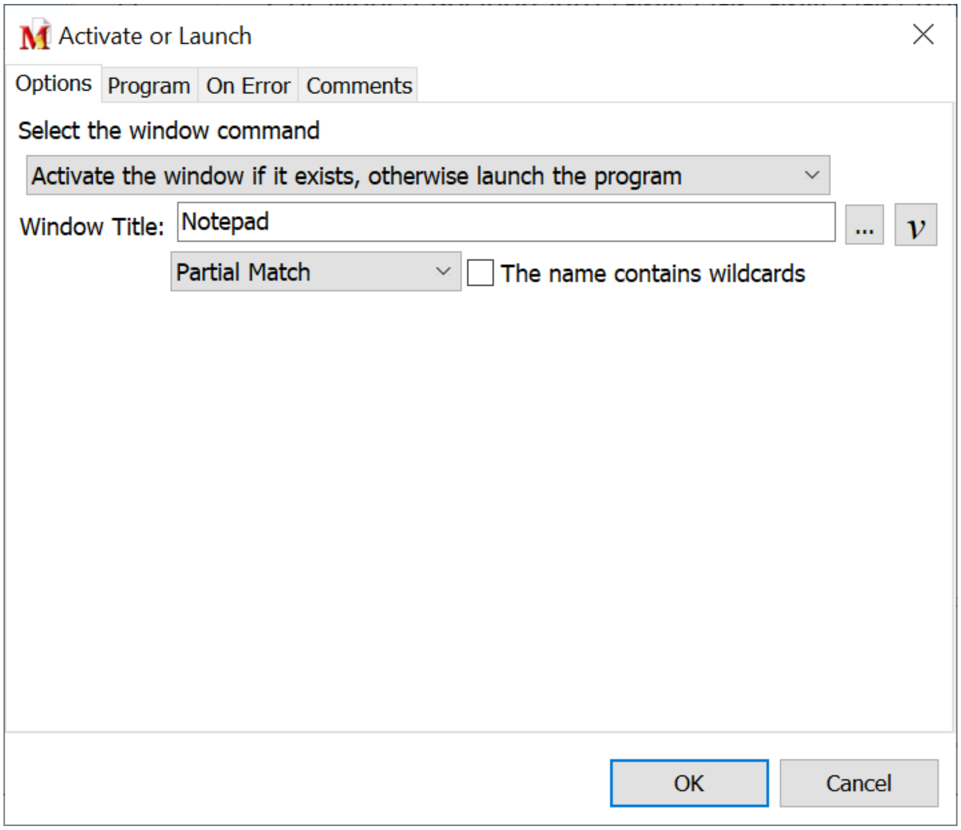
Script Editor > Expand Window Controls Category > Set Focus

Set Focus to Control

First use the Get Control or Capture Control command to save the control to a variable. Then select the variable to receive focus.

Activate or Launch

The Activate or Launch command first looks to see if the program window selected is running. If so, the macro activates the window or brings focus to that window. If the window is not running, the macro launches the program and then activates the window.



Select the window command

Activate the window if it exists, otherwise launch the program

Window Title: Notepad

Partial Match

□ The name contains wildcards

Script Editor > Expand Windows/Programs Category > Activate or Launch

Options

Window Title

Enter the title of the window to open. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

> *Exact Match*
>
> This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Program

Program/Path Name

Enter the complete file path and program name (typically a .exe file). Or click on the Browse button to search for the program desired. Select programs listed in the Start menu or from a list of processes running on the computer. If the program is not found, then click on the Browse button of the Select a Program window to access all programs on the computer.

Run Mode

Choose to have the program launch and either maximize, minimize, hide or display the window as it would normally at program launch. These options may not be available if the application being launched does not support them.

Program Parameters

Some programs allow values following the name of the program to make it behave in a specific manner. These are called Program Parameters and can be set in this field, if desired.

Advanced

Wait for ___ Seconds After Executing the Command

This Wait option instructs the macro to wait after executing the program and attempting to switch to the window specified in the Window Title field. For many programs, this will simply be zero as no wait is required. For others, it may require a second or two delay. A little experimenting may be necessary.

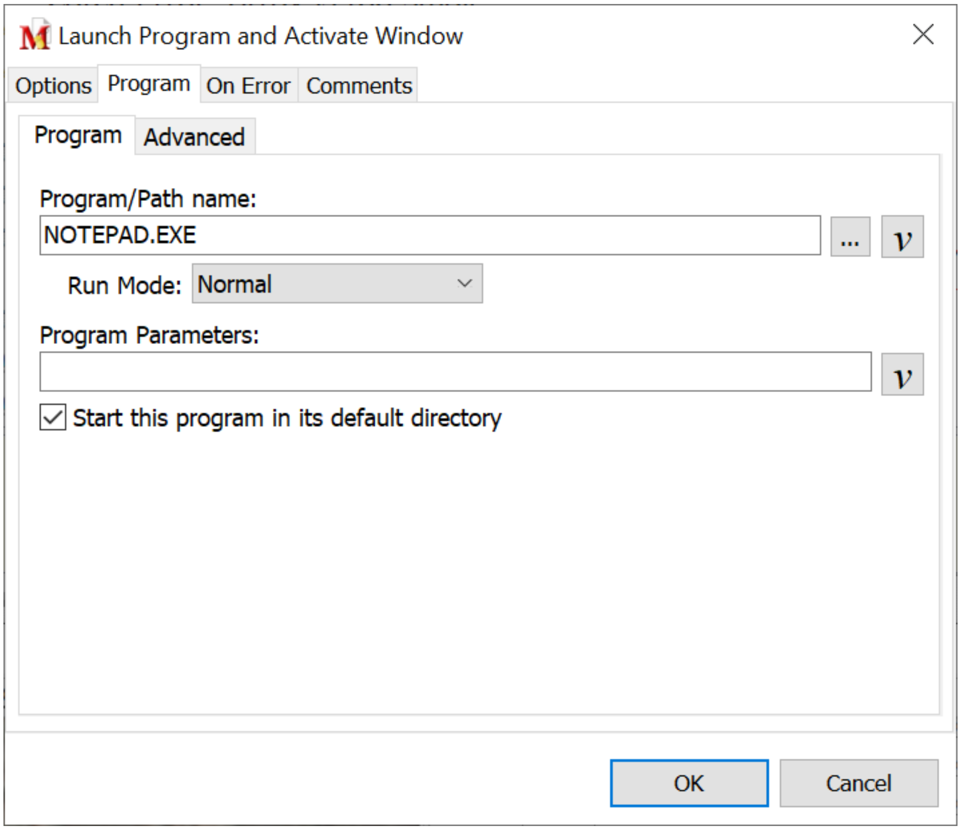
Wait for ___ Seconds After Launching the Program

This specifies the maximum time the macro should wait for the window title of the launched program to appear. Some programs may take a little while to load before the final window appears that has the correct window title. The default value for this field is 12 seconds.

Some larger programs may take longer than 12 seconds to load, especially on a fresh reboot of Windows where many common DLL's have not yet been loaded. Some experimenting with this value may be necessary when launching larger applications. But for most programs 12 seconds will be sufficient.

Launch Program and Activate Window

This command instructs the macro to launch a program and activate the window (or give focus to the window) that opens when the program is started. Define the window and program to launch.



Program/Path name: NOTEPAD.EXE

Run Mode: Normal

Program Parameters:

☑ Start this program in its default directory

Script Editor > Expand Windows/Programs Category > Launch Program and Activate Window

Options

Window Title

Enter the title of the window to open. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Program

Program/Path Name

Enter the complete file path and program name (typically a .exe file). Or click on the Browse button to search for the program desired. Select programs listed in the Start menu or from a list of processes running on the computer. If the program is not found, then click on the Browse button of the Select a Program window to access all programs on the computer.

- ✓ **Note:** Any type of file extension that has been defined on the computer may be inserted in the Program Path/Name field. As long as there is a file type association set up in Windows for that particular file type, the corresponding program is loaded and the specific file is opened. For example, if inserting C:\My Documents in the file path, and .doc files are associated with Microsoft Word, then Microsoft Word starts and loads the test file.

Run Mode

Choose to have the program launch and either maximize, minimize, hide or display the window as it would normally at program launch. These options may not be available if the application being launched does not support them.

Program Parameters

Some programs allow values following the name of the program to make it behave in a specific manner. These are called Program Parameters and can be set in this field, if desired.

Start this Program in its Default Directory

Some programs require a specific default folder to be set before they are run. The default folder may contain the program shortcut or the program .exe file. This option allows the program to start from the default directory.

Advanced

Wait for ___ Seconds After Executing the Command

This option instructs the macro to wait after executing the program and attempting to switch to the window specified in the Window Title field. For many programs, this will simply be zero as no wait is required. For others, it may require a second or two delay. A little experimenting may be required.

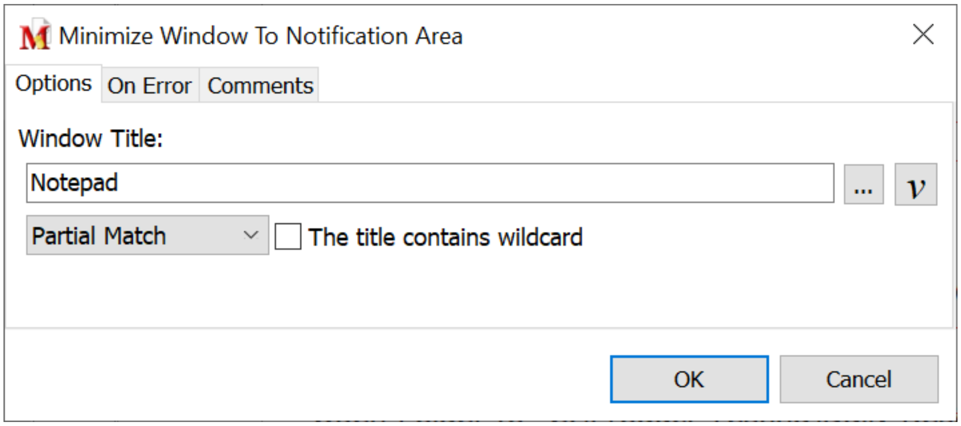
Wait for ___ Seconds After Launching the Program

This specifies the maximum time the macro should wait for the window title of the launched program to appear. Some programs may take a little while to load before the final window appears that has the correct window title. The default value for this field is 12 seconds.

Some larger programs may take longer than 12 seconds to load, especially on a fresh reboot of Windows where many common DLL's have not yet been loaded. Some experimenting with this value may be necessary when launching larger applications. But for most programs, 12 seconds will be sufficient.

Minimize Window to Notification Area

The Minimize Window to Notification Area command minimizes the window selected and places the program's icon in the notification area. Clicking on the icon in the tray restores the window to its original location and size.



Window Title: Notepad

Partial Match

□ The title contains wildcard

Script Editor > Expand Windows/Programs Category > Minimize Window to Notification Area

Window Title

Enter the title of the window to minimize to the notification area. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

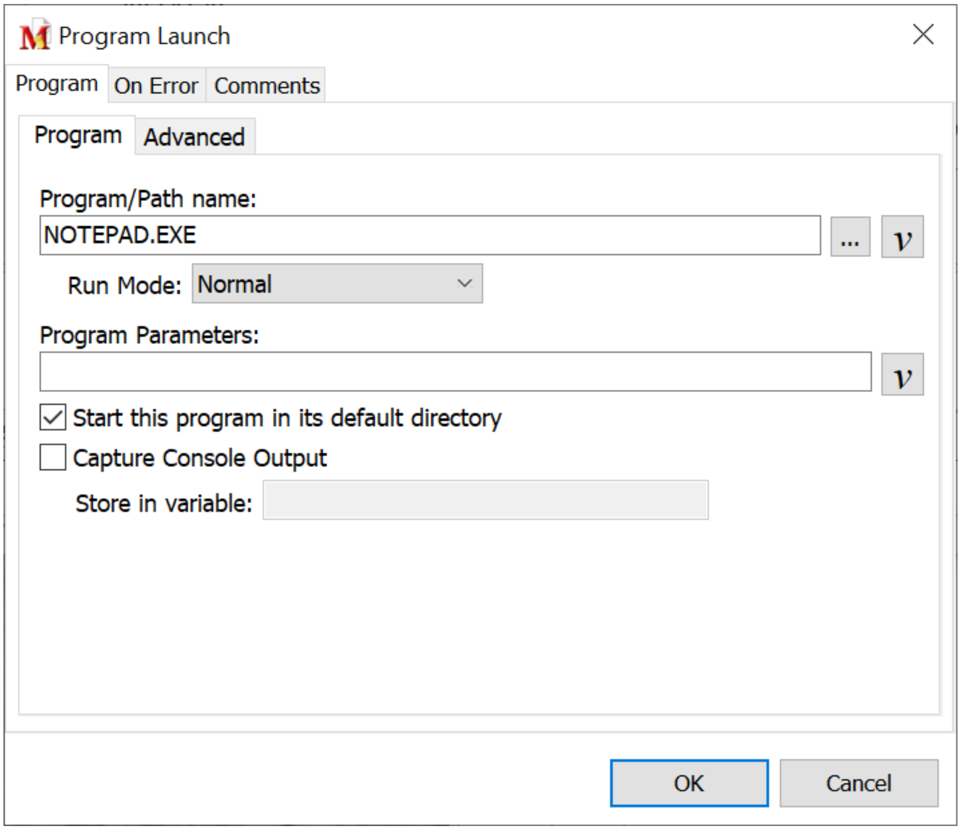
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Program Launch

The Program Launch command instructs the macro to start a specific program. If the program cannot be found, the macro terminates. This option may not display the launched program on top of other programs currently running. To ensure that the program window opens at program launch, use the Launch Program and Activate Window command.



Program/Path name: NOTEPAD.EXE

Run Mode: Normal

Program Parameters:

☑ Start this program in its default directory

☐ Capture Console Output

Store in variable:

Script Editor > Expand Windows/Programs Category > Program Launch

Program

Program/Path Name

Enter the complete file path and program name (typically a .exe file). Or click on the Browse button to search for the program desired. Select programs listed in the Start menu or from a list of processes running on the computer. If the program is not found, then click on the Browse button of the Select a Program window to access all programs on the computer.

- ✓ **Note:** Any type of file extension that has been defined on the computer may be inserted in the Program Path/Name field. As long as there is a file type association set up in Windows for that particular file type, the corresponding program will be loaded and the document specified will be opened. For example, if you insert C:\My Documents in the file path, and .doc files are associated with Microsoft Word, then Microsoft Word will start and load the test file.

Run Mode

Choose to have the program launch and either maximize, minimize, hide or display the window as it would normally at program launch. These options may not be available if the application being launched does not support them.

Program Parameters

Some programs can have values following the name of the program that can make the program behave in a specified manner. These are called Program Parameters and can be set here if desired.

Start this Program in its Default Directory

Some programs require a specific default folder to be set before they are run. The default folder may contain the program shortcut or the program .exe file. This option allows the program to start from the default directory.

Capture Console Output

This option captures the standard output (stdout) from the launched application and places output in a variable. When using this option, the Macro Express script does not continue until the application being launched closes.

Store in Variable

Select or input the variable to store the information output from the Console Application.

Advanced

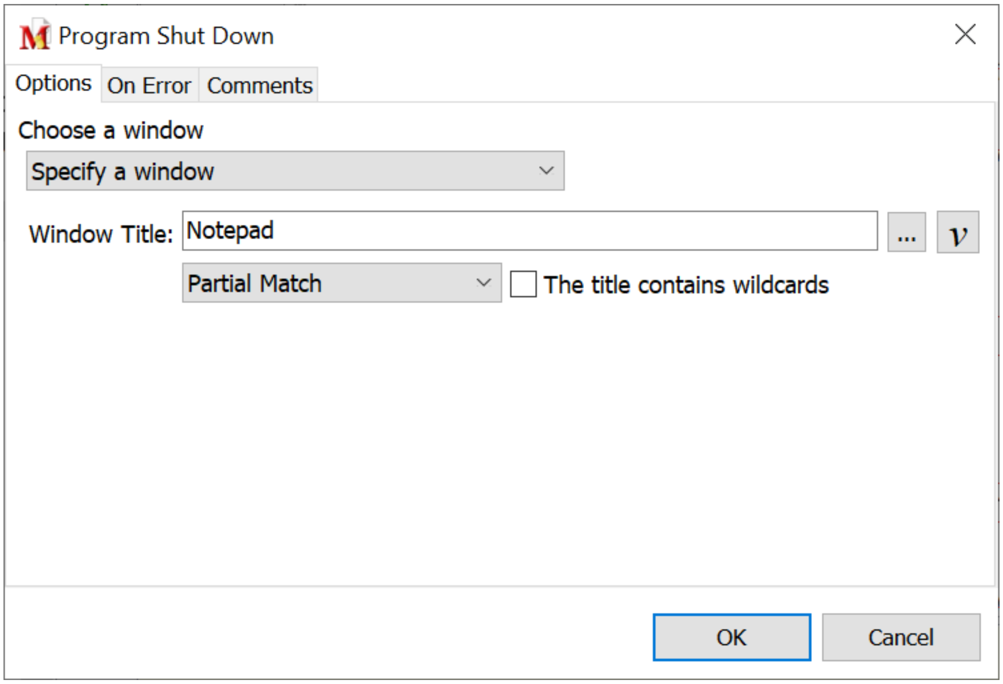
Wait for _____ Seconds after Executing the Command

This option instructs the macro to wait after executing the program. For many programs this value will simply be zero as no wait is required. For others, it may require a second or two delay. A little experimenting may be required.

Program Shut Down

Program Shut Down closes a program based on the window title name. While the Window Close command can be used to shut down a program if the window specified is the main window, the Program Shut Down option closes the program regardless of which program window is selected.

- ✓ **Note:** Another option for closing programs is the Terminate Process command. This option also closes applications where the window title is not seen, such as notification area applications that run in the background.



Choose a window: Specify a window

Window Title: Notepad

Partial Match

□ The title contains wildcards

Script Editor > Expand Windows/Programs Category >Program Shut Down

Choose a Window

First choose the window of the program to shut down. Select from the current window or specific window options.

Current Window

Current Window refers to the window that has focus when the Program Shut Down command is run in the macro.

Specific Window

Enter the title of the specific window to close. Or click on the Browse button to view a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

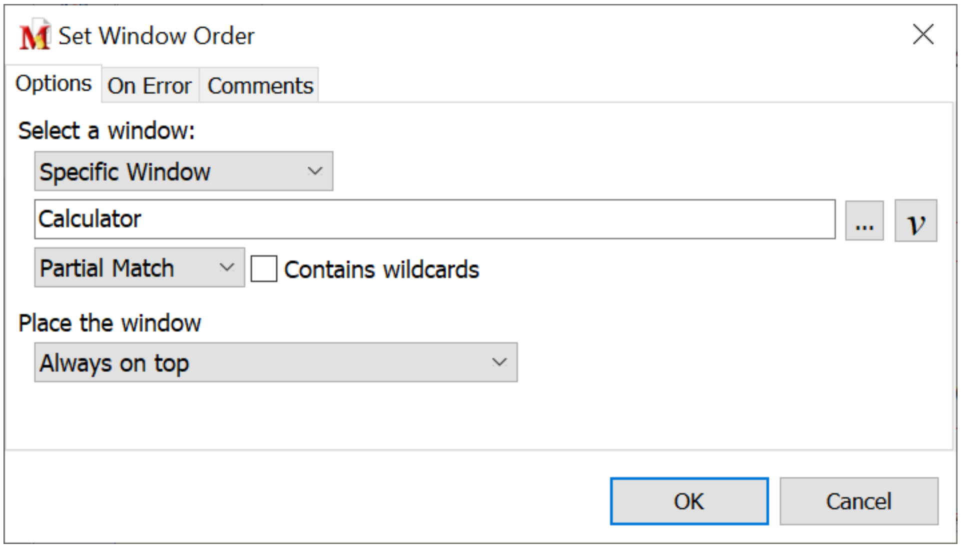
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Set Window Order

Set Window Order places a window as the top most window or sends it to the bottom of the list of windows open on the screen.



Select a window:
Specific Window
Calculator
Partial Match
□ Contains wildcards

Place the window
Always on top

Script Editor > Expand Windows/Programs Category > Set Window Order

Select a Window

First choose the window to order. Select from the current window or specific window options.

Current Window

Current Window refers to the window that has focus when the Set Window Order command is run in the macro.

Specific Window

Enter the title of the specific window to close. Or click on the Browse button to view a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

> *Exact Match*
>
> This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.
>
> *Partial Match*
>
> The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the

window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

Contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Place the Window

After selecting the window, now choose the window order.

Always on Top

This option sets the window so that it is always on top. If other windows are opened, the window selected still displays on top of all others.

Remove Topmost Status and Place on Top

If the window was previously set to always be on top, this option removes the always on top status and leaves the window on top of all others.

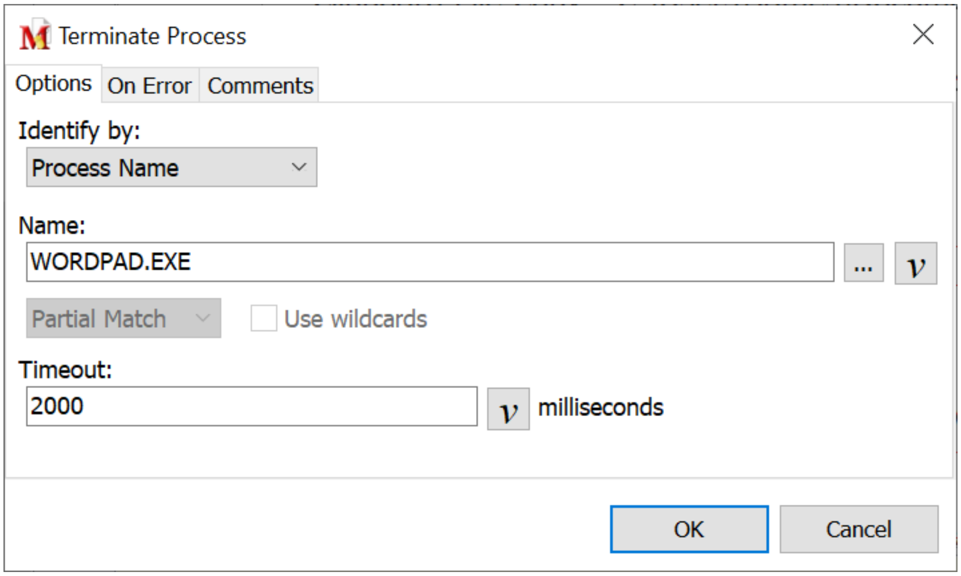
Otherwise this option places the window on top of all others, but does not set the always on top status.

Place on Bottom

Place on Bottom sends the window to the bottom of the list of windows open on the computer.

Terminate Process

The Terminate Process command goes through several routines to make sure that the window or program selected is completely closed.



Identify by: Process Name

Name: WORDPAD.EXE

□ Use wildcards

Timeout: 2000 milliseconds

Script Editor > Expand Windows/Programs Category > Terminate Process

Identify By

First select whether to terminate the process by process name (typically a .exe file name) or by the window title.

Name

Based on the selection above, insert the name of the window or process to close. Or click on the Browse button to choose from a list of open windows or running processes.

If using the Close Window option, select an exact or partial match of the window title.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

Use Wildcards

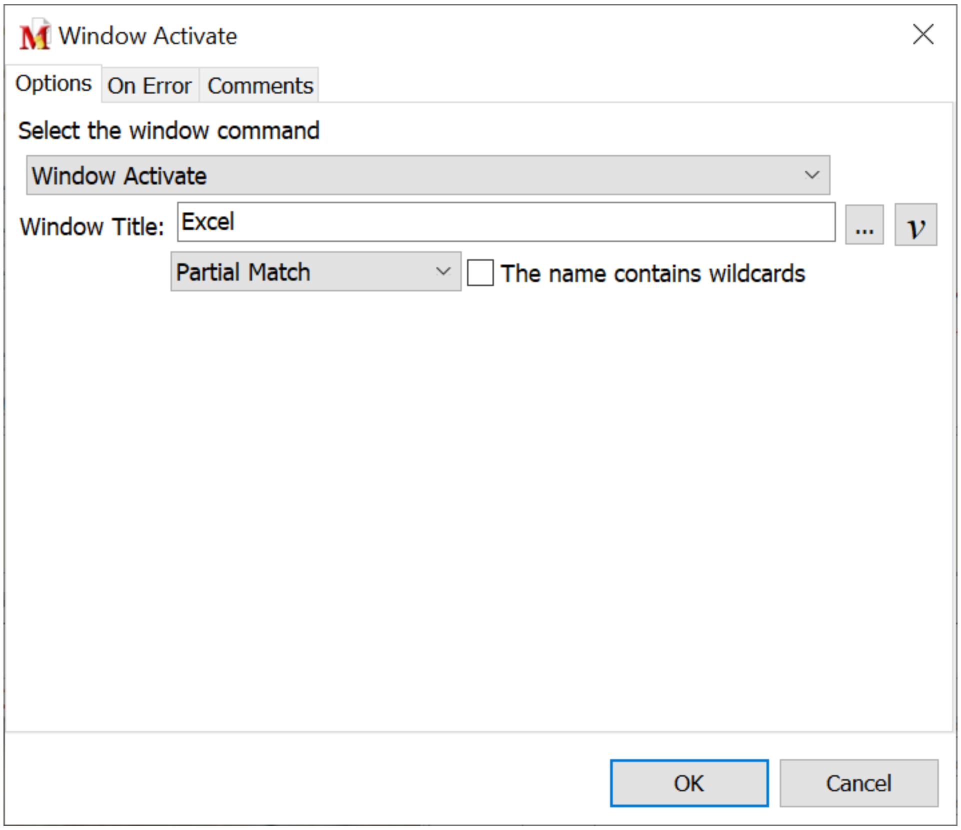
Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Timeout:

The Terminate Process command uses several methods to attempt to close the window or program. The Timeout setting determines how long Macro Express waits before trying the next method.

Window Activate

The Window Activate command activates the selected window or in other words brings focus to the window so that it can accept keystroke or mouse input.



Select the window command: Window Activate

Window Title: Excel

Partial Match

□ The name contains wildcards

Script Editor > Expand Windows/Programs Category > Window Activate

Window Title

Enter the title of the window to activate. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window

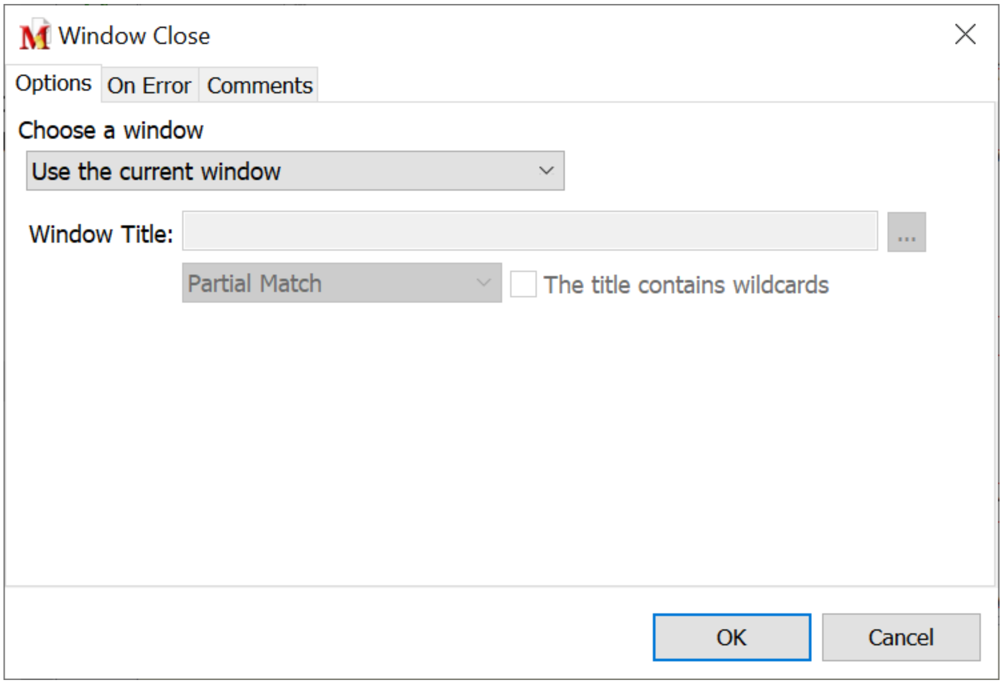
Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Name contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Window Close

This command instructs a window to close. If the window selected is the window that opens when the program loads, the application as well as the window closes.



Choose a window

Use the current window

Window Title:

Partial Match

□ The title contains wildcards

Script Editor > Expand Windows/Programs Category > Window Close

Choose a window

Choose from the drop down list whether to close the current window or a specific window.

Current Window

Current window refers to the window that has focus when the Window Close command plays in the macro.

Specific Window

Enter the title of the specific window to close. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

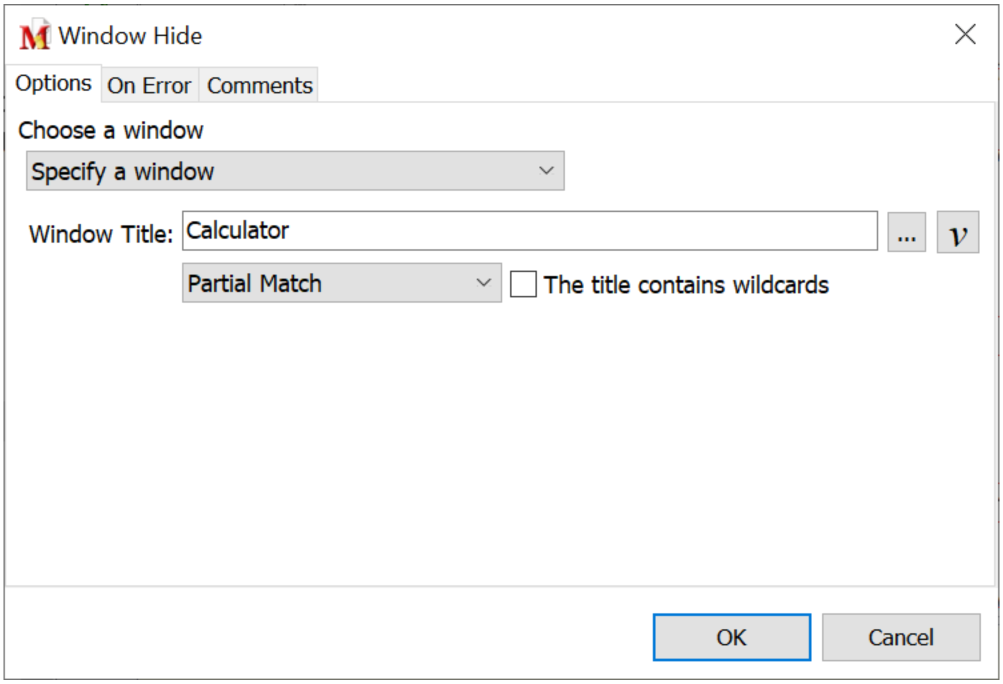
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Window Hide

This instructs a window to be hidden from the screen. The window is still running, but is hidden from view. Use the Window Show command to display the window so that it returns to view.



Choose a window

Specify a window

Window Title: Calculator

Partial Match

□ The title contains wildcards

Script Editor > Expand Windows/Programs Category > Window Hide

Choose a window

Choose from the drop down list whether to hide the current window or a specific window.

Use the Current Window

Current Window refers to the window that has focus when the Window Hide command runs during macro playback. Whichever window is on top and has focus when the Window Hide command runs is hidden.

Specific Window

Enter the title of the specific window to hide. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro

would not wait.

Partial Match

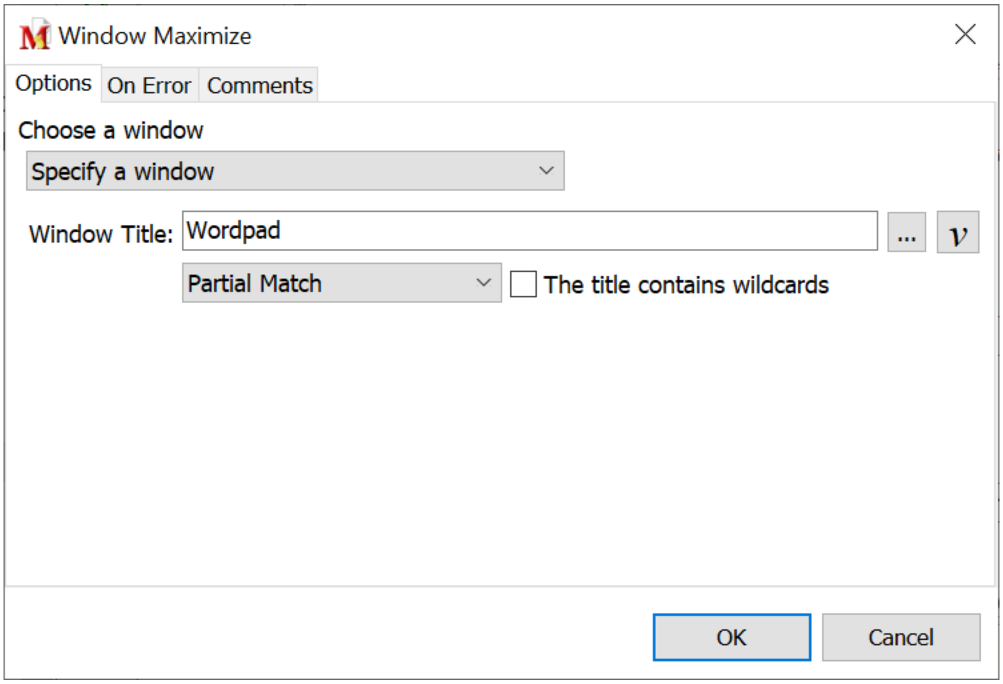
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Window Maximize

The Windows Maximize command resizes a window to fill the entire screen.



Choose a window: Specify a window

Window Title: Wordpad

Partial Match

□ The title contains wildcards

Script Editor > Expand Windows/Programs Category > Window Maximize

Choose a Window

Choose from the drop down list whether to maximize the current window or a specific window.

Use the Current Window

This refers to the window that has focus when the Window Maximize command runs during macro playback. Whichever window is on top and has focus when the Window Maximize command runs is maximized.

Specify a Window

Enter the title of the specific window that you want to maximize. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

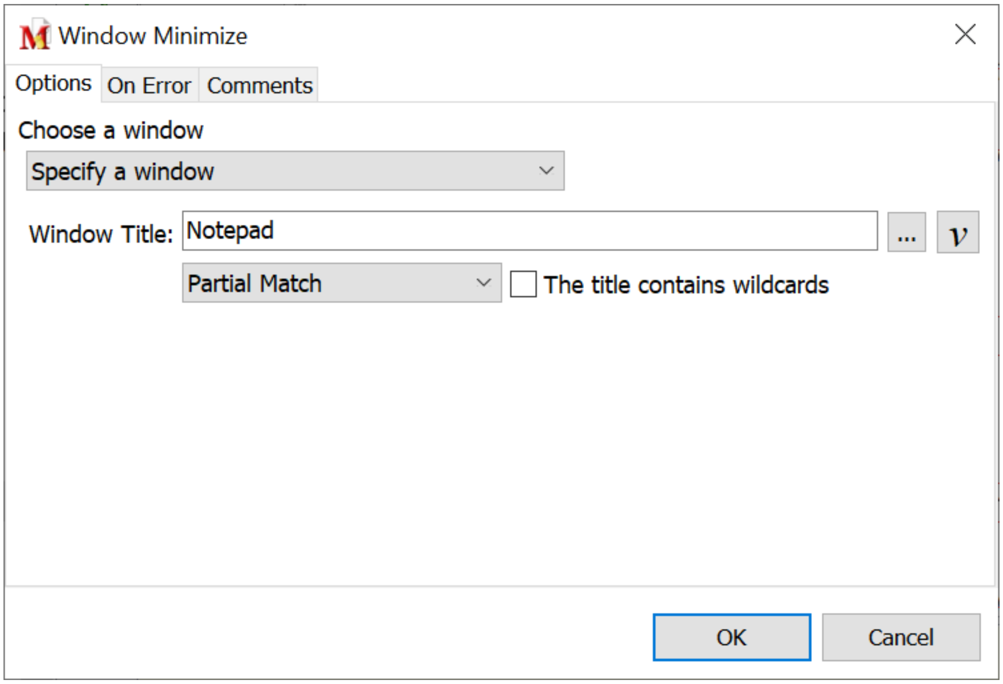
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Window Minimize

The Window Minimize command reduces the selected window to a button on the Task Bar.



Choose a window: Specify a window

Window Title: Notepad

Partial Match

□ The title contains wildcards

Script Editor > Expand Windows/Programs Category > Window Minimize

Choose a window

Choose from the drop down list whether to minimize the current window or a specific window.

Use the Current Window

Current Window refers to the window that has focus when the Window Minimize command is run during macro playback.

Specific Window

Enter the title of the specific window to minimize. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

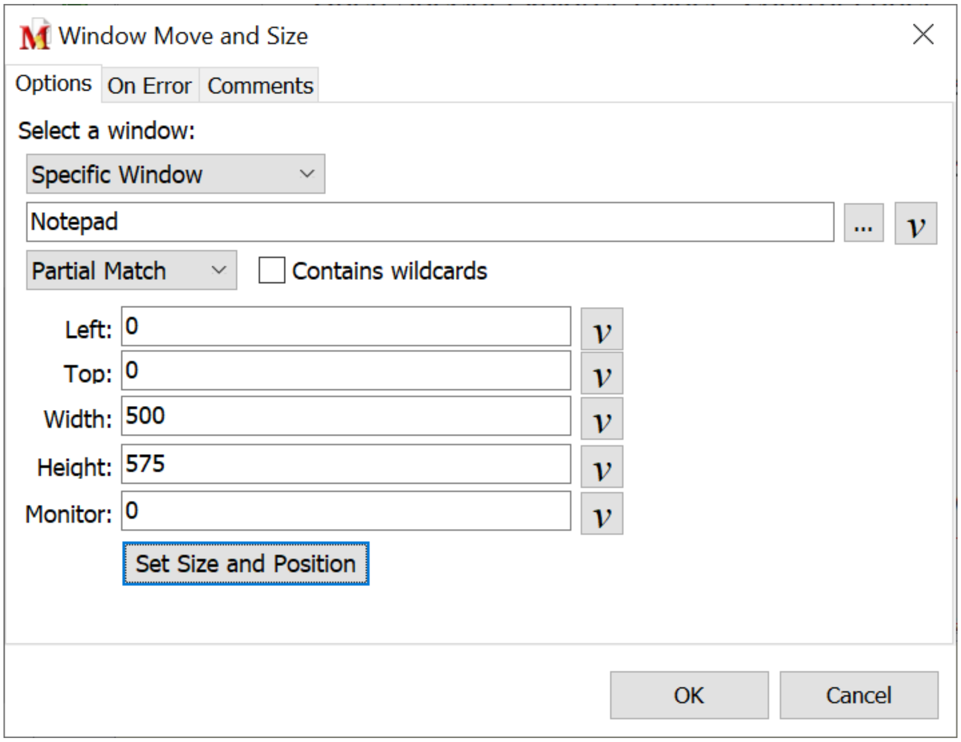
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Window Move and Size

The Window Move and Size command provides an easy method for sizing and positioning a window on the screen. Set up automated sequences for rearranging windows on the desktop. Or standardize the window position and size for each macro run to ensure that mouse moves and clicks work correctly.



Select a window: Specific Window

Notepad

Partial Match

□ Contains wildcards

Left: 0

Top: 0

Width: 500

Height: 575

Monitor: 0

Script Editor > Expand Windows/Programs Category > Window Move and Size

Select a Window

Choose from the drop down list whether to reposition the current window or a specific window.

Current Window

This refers to the window that has focus when the Window Move and Size command runs.

Specific Window

Enter the title of the specific window to be moved and resized. Or click on the Select Window button to see a list of all windows currently running. Select the window needed and the title is inserted in the edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window to be moved and resized. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the command would fail.

Partial Match

The Partial Match option does not require an exact window title match. For example, if resizing or moving a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Coordinates

Enter the coordinates manually or use the Size and Position Tool to simplify the process.

Specify Exact Size and Location

Enter the coordinates directly into the edit box. The Left and Top options determine the location of the window. The Height and Width options determine the size of the window. The values are expressed in pixels.

If using multiple monitors, select which monitor to display the window. The first monitor receives a value of 0 (zero). A second monitor receives a value of 1 (one), the third monitor 2 (two), and so on. If using a single monitor, leave the setting as 0.

Below is an example of setting the exact size and location of a window.

Setting	Value	Description
Left	100	The left edge of the window is positioned 100 pixels to the right of the left side of the screen.
Top	200	The top edge of the window is positioned 200 pixels down from the top of the screen.
Width	600	The window is sized to 600 pixels in width.

Height	750	The window height is set to 750 pixels.
Monitor	0	Zero represents the main monitor.

Set Size and Position

Use the Set Size and Position tool for a simple way to select the size and position of the window. Clicking on this button opens the Window Move and Size window and the Dialog Positioner. Stretch this window to the size needed and drag to the location desired. Or after sizing the window, click on one of the position buttons in the Dialog Positioner to have the window move to the location indicated.

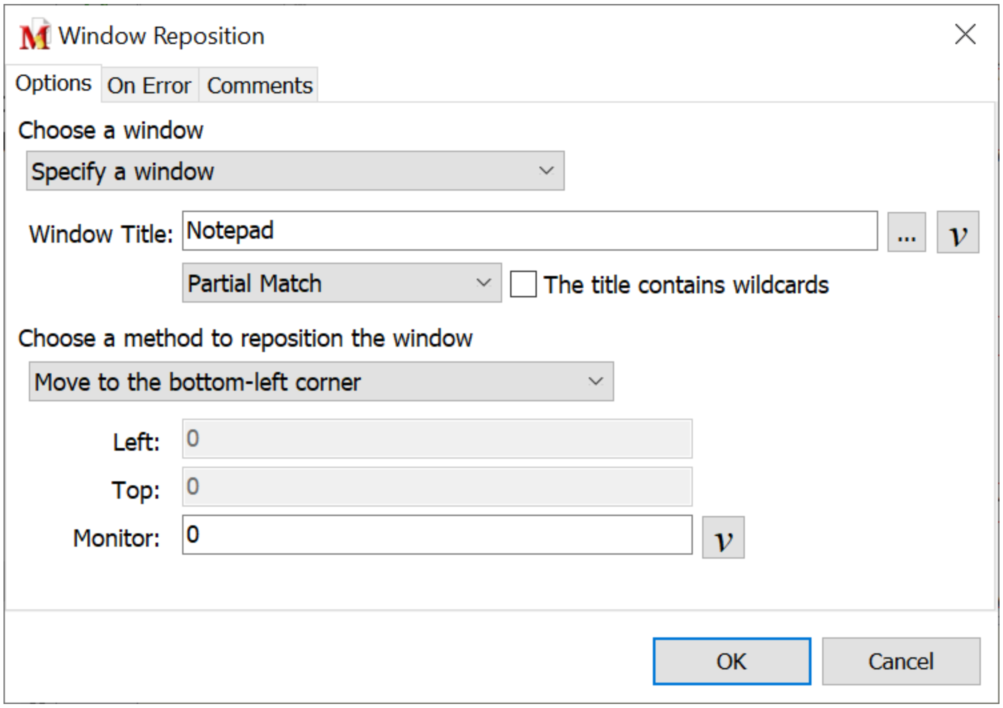
Click ok to save. The pixel values are automatically filled out.

When the macro is run, the window selected in the Window Title field is moved and sized to the same location and dimensions as drawn in the Window Tool window.

The Window Position and Size topic explains these options in detail.

Window Reposition

Use this command to position program windows anywhere on the screen.



Choose a window
Specify a window

Window Title: Notepad

Partial Match
□ The title contains wildcards

Choose a method to reposition the window
Move to the bottom-left corner

Left: 0
Top: 0
Monitor: 0

Script Editor > Expand Windows/Programs Category > Window Reposition

Choose a window

Choose from the drop down list whether to reposition the current window or a specific window.

Use the Current Window

Use the Current Window refers to the window that has focus when the Window Reposition command runs during macro playback. Whichever window is on top and has focus when the command runs will be repositioned.

Specific Window

Enter the title of the specific window to reposition. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Choose a Method to Reposition the Window

Twelve reposition options are available. Nine of these options are essentially self-explanatory. Clicking on the Center of Screen option places the window in the center of the screen. The Move to Top Left command moves the window to the top left corner of the screen, etc. We'll discuss the other three choices below.

Specify Exact Location

This option places the window anywhere on the screen based on the coordinates that you provide. Enter the coordinates for positioning the window in the Left and Top edit fields.

Specify Exact Location Example

Left - 100
Top - 200

With the above coordinates, the left edge of the window is positioned 100 pixels to the right of the left side of the screen. The top edge of the window is positioned 200 pixels down from the top of the screen.

Move in Pixels Relative to the Current Location

This option moves the window a fixed number of pixels away from the current location of the window. Insert the number of pixels in the left and top boxes when using the Value option. Or insert integer variables in these fields. Negative numbers (place a " -" sign before the number) move the window to the left or up. Positive numbers move the window to the right or down.

Move In Pixels Example

Left - -150
Top - 250

For this example, we'll assume the current window is positioned at left 400 and top 100.

Moving the window the number of pixels we have listed above, positions the window at left 250 and top 350. The window moved to the left 150 pixels and down 250 pixels on the screen.

The left edge of the window is now positioned 250 pixels to the right of the left side of the screen. The top edge of the window is positioned 350 pixels down from the top of the screen.

Move in Percent Relative to the Current Location

This option moves the window proportionally, on a percentage basis. Insert the percent change in the left and top boxes when using the Value option. Or insert integer variables in these fields. Negative numbers (place a " -" sign before the number) move the window up or to the left. Positive numbers move the window down or to the right.

Move In Percent Example

Left - -10
Top - 25

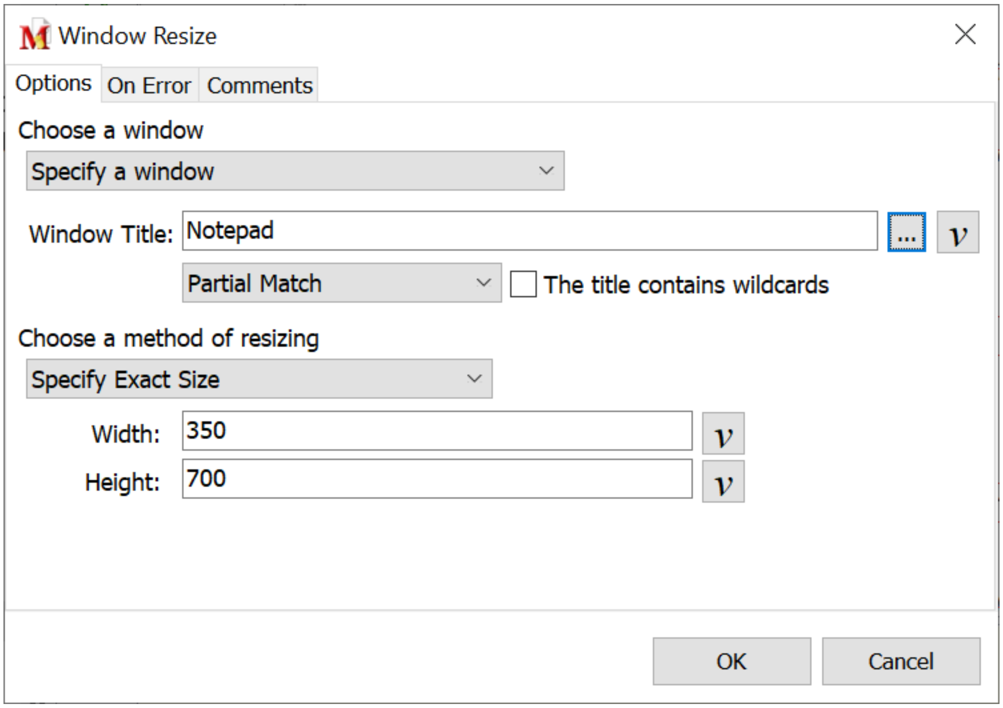
For this example, we'll assume the current window is positioned at left 400 and top 500.

Using the values we input above, the negative 10 value moves the window to the left 10% of the available screen distance.

The positive 25 value moves the window down 25% of the available distance towards the bottom of the screen.

Window Resize

Three options are available for changing the size of a window - select a specific size in pixels, resize by a percentage or change the size by a set number of pixels. These options are described below.



Choose a window: Specify a window

Window Title: Notepad

Partial Match □ The title contains wildcards

Choose a method of resizing: Specify Exact Size

Width: 350

Height: 700

Script Editor > Expand Windows/Programs Category > Window Resize

Choose a window

Choose from the drop down list whether to resize the current window or a specific window.

Current Window

Current window refers to the window that has focus when the resize command is run in the macro.

Specify a Window

Enter the title of the specific window that you want to resize. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro

would not wait.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Choose a Method of Resizing

Resize the selected window by using one of the three options listed in the drop down menu. The resizing methods are described below.

Specify Exact Size

The Specify Exact Size option sizes the window to specific dimensions. Insert the pixel values for the width and height of the window in the appropriate fields. If 500 in entered in the width box and 400 in the height box, the window is resized to a width of 500 pixels and a height of 400 pixels during macro playback. Integer variables may also be used in the width and height fields.

Enlarge/Decrease in Pixels

This option enlarges or decreases the size of a window by a fixed number of pixels. Insert the number of pixels in the width and height boxes by which to increase or decrease the window size. Integer variables may also be used in the width and height fields.

Negative numbers (place a " -" sign before the number) decrease the size of the window. Positive numbers increase the window size. In the example above, a window size is set to width 350 and height 700 pixels. Inserting values of width -100 and a height of 50 change the window size to width 250 and height 750 when the macro is run.

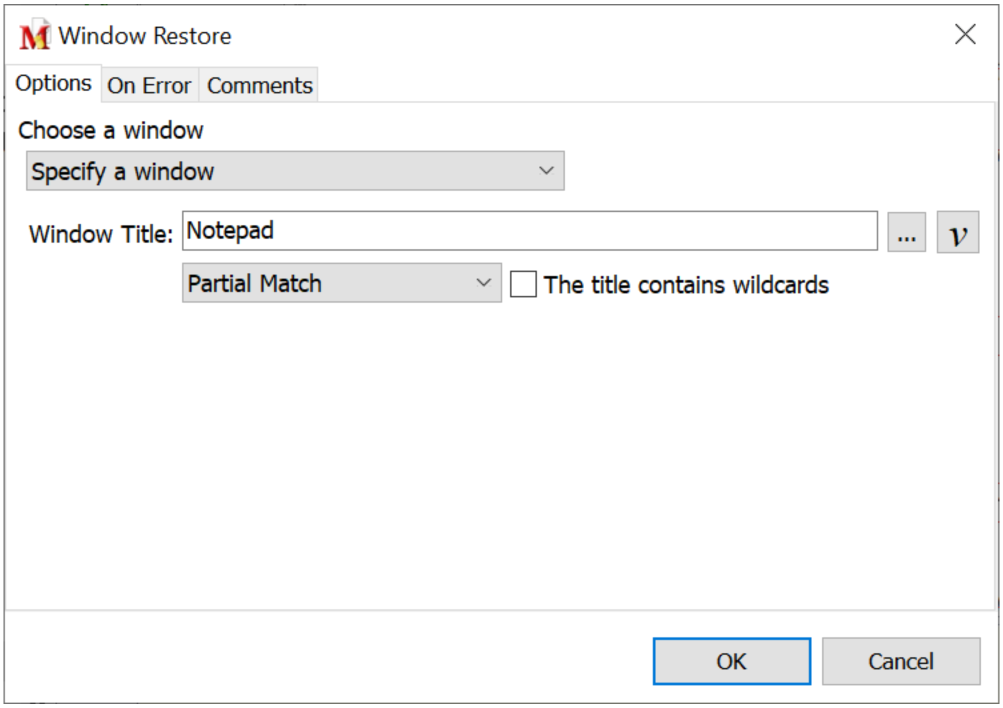
Enlarge/Decrease by Percent

This option enlarges or decreases the size of a window proportionally - on a percentage basis. Insert the percentage values by which to increase or decrease the window size. Integer variables may also be used in the width and height fields.

Negative numbers (place a " -" sign before the number) decrease the size of the window. Positive numbers increase the window size. In the first example, a window size is set to a size of width 500 and height 400 pixels. Inserting values of width -10 and a height of 25 will decrease the window width by 10% and increase the height by 25% when the macro is run.

Window Restore

The Window Restore command instructs a window to be restored (resized) to its original dimensions before a maximize or minimize command was issued.



Choose a window: Specify a window

Window Title: Notepad

Partial Match

□ The title contains wildcards

Script Editor > Expand Windows/Programs Category > Window Restore

Choose a window

Choose from the drop down list whether to restore the current window or a specific window.

Current Window

Current window refers to the window that has focus when the Window restore command is run in the macro.

Specific Window

Enter the title of the specific window to restore. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro would not wait.

Partial Match

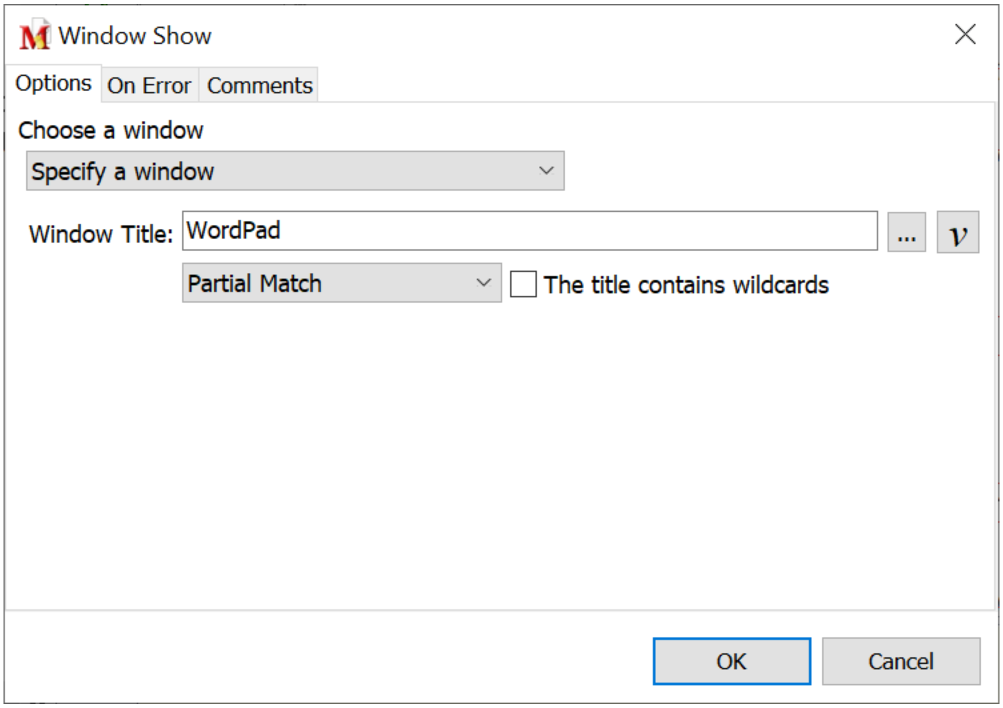
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Window Show

The Window Show command displays a window that has been hidden. It typically brings the window to the front of the screen. Use the Window Activate command to bring a window to the top of the screen.



Choose a window

Specify a window

Window Title: WordPad

Partial Match

□ The title contains wildcards

Script Editor > Expand Windows/Programs Category > Window Show

Choose a window

Choose from the drop down list whether to show the current window or a specific window.

Current Window

Current window refers to the window that has focus when the Window Show command is run in the macro.

Specific Window

Enter the title of the specific window to show. Or click on the Browse button to see a list of all windows currently running. Select the window from the list to insert the name into the Window Title edit box.

Exact Match

This requires that the window name entered matches exactly with the name of the window waiting to receive focus. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the macro

would not wait.

Partial Match

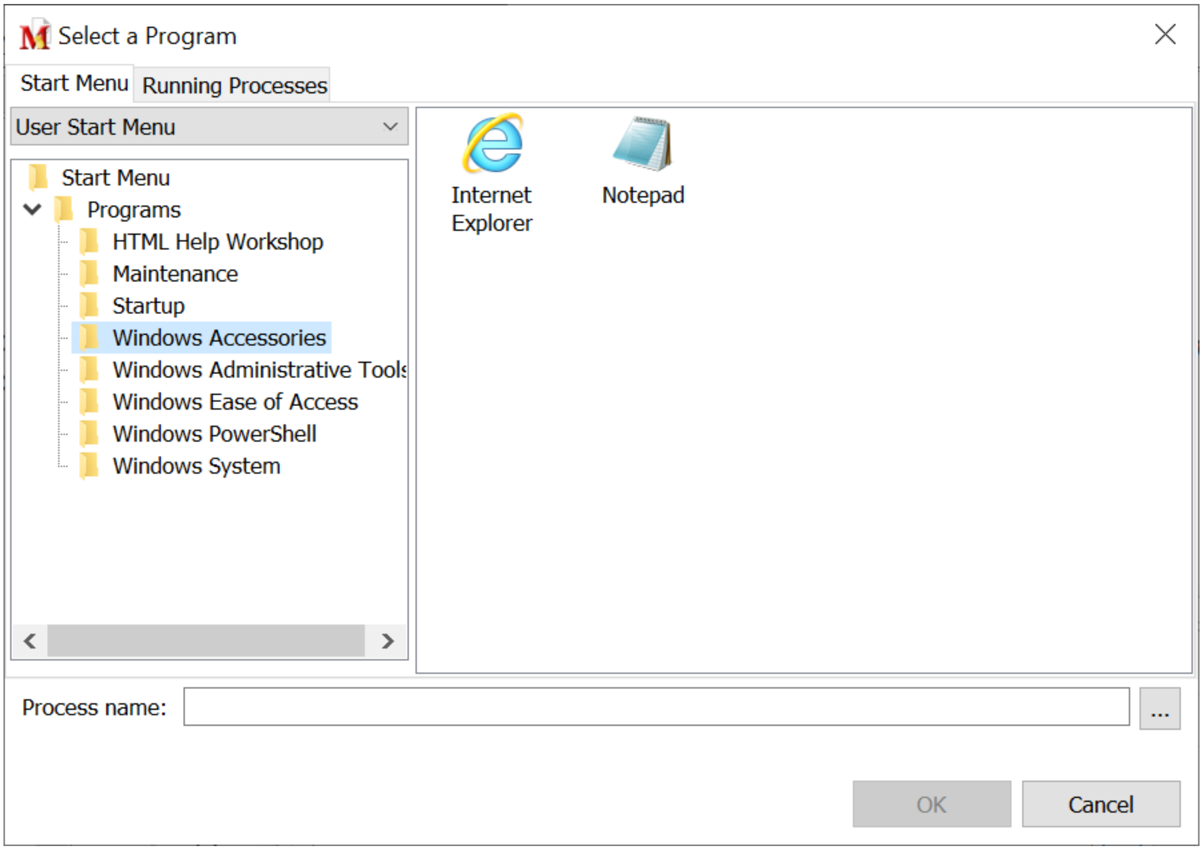
The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The Title contains Wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

Add Program

This window or similar is used in several places throughout the program, such as selecting a program specific scope, using the Terminate Process command, the Program Launch command and others that require selecting a program.



Process name:

There are three methods for selecting a program. Each are described below.

Start Menu

Choose from programs that are listed in the Windows Start Menu. Choose from the drop down menu to find a program in the User Start Menu or the Shared Start Menu. Highlight the program to insert in the Executable Path field.

Running Processes

Click on the Running Processes tab to view a list of all programs currently running on the computer. Highlight the program to insert in the Executable Path field.

Browse

Click on the Browse button to open a window similar to My Computer. Navigate to find the program on the computer and select.

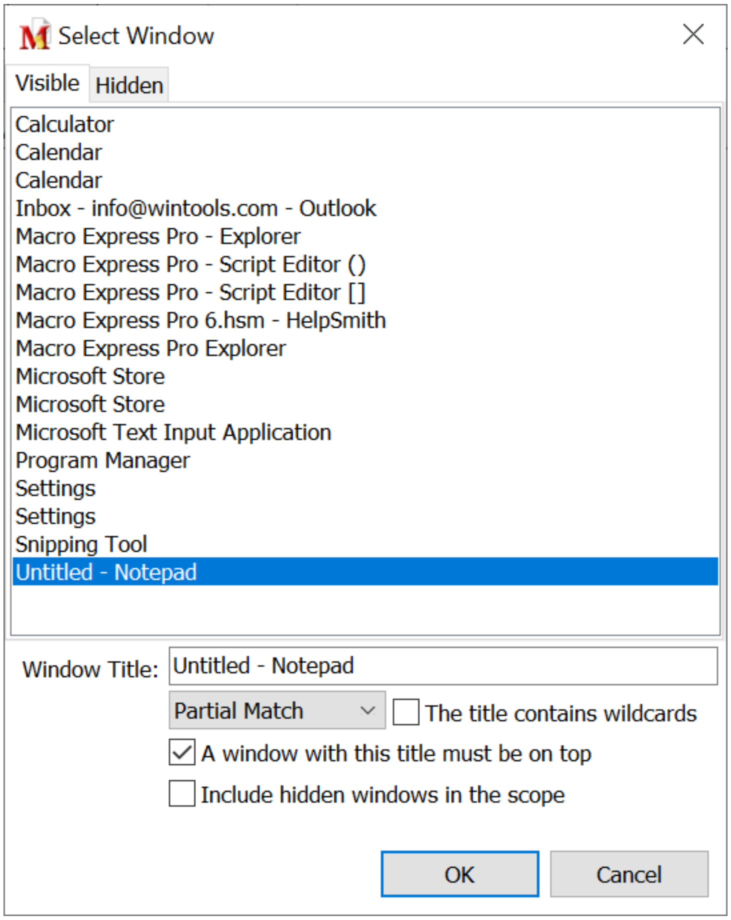
This program must have a window on top

This option requires that a window of the program selected be on top in order for the macro to continue. Otherwise, the macro will complete it's task whether the program window is on top or not.

This option is available when selecting a Scope.

Add Window

This window or similar is used in several places throughout the program, such as selecting a window specific scope, using the Wait for Window Title command, the Window Activate command and others that require selecting a window title.



Calculator
Calendar
Calendar
Inbox - info@wintools.com - Outlook
Macro Express Pro - Explorer
Macro Express Pro - Script Editor ()
Macro Express Pro - Script Editor []
Macro Express Pro 6.hsm - HelpSmith
Macro Express Pro Explorer
Microsoft Store
Microsoft Store
Microsoft Text Input Application
Program Manager
Settings
Settings
Snipping Tool
Untitled - Notepad

Window Title: Untitled - Notepad

Partial Match
□ The title contains wildcards
☑ A window with this title must be on top
□ Include hidden windows in the scope

Visible

The Visible tab displays a list of windows currently running on the computer that are viewable.

Hidden

The Hidden tab displays a list of windows currently running on the computer that are not open (minimized to the notification area) or run in the background and are never seen.

Window Title

Enter the name of the window title or select the title from the hidden or visible list of windows.

Exact Match

This requires that the window name entered matches exactly with the name of the window running. For example, if listing Notepad as the window name, but the actual window title is Notepad - Untitled, then the match is ignored.

Partial Match

The Partial Match option does not require an exact window title match. For example, if waiting for a notepad window, inserting "notepad" or "notep" as the window title will find the running window. This is particularly useful as some Window Titles change based on whether a given file is open. Case sensitivity is not required. For example, "NOTepad" will work as well as "notepad".

The title contains wildcards

Select this option if a wildcard is included in the window title name. A wildcard uses the " * " or " ? " characters in the title to expand the possibilities of a match. These are similar to the wildcards used in Windows for searches.

A window with this title must be on top

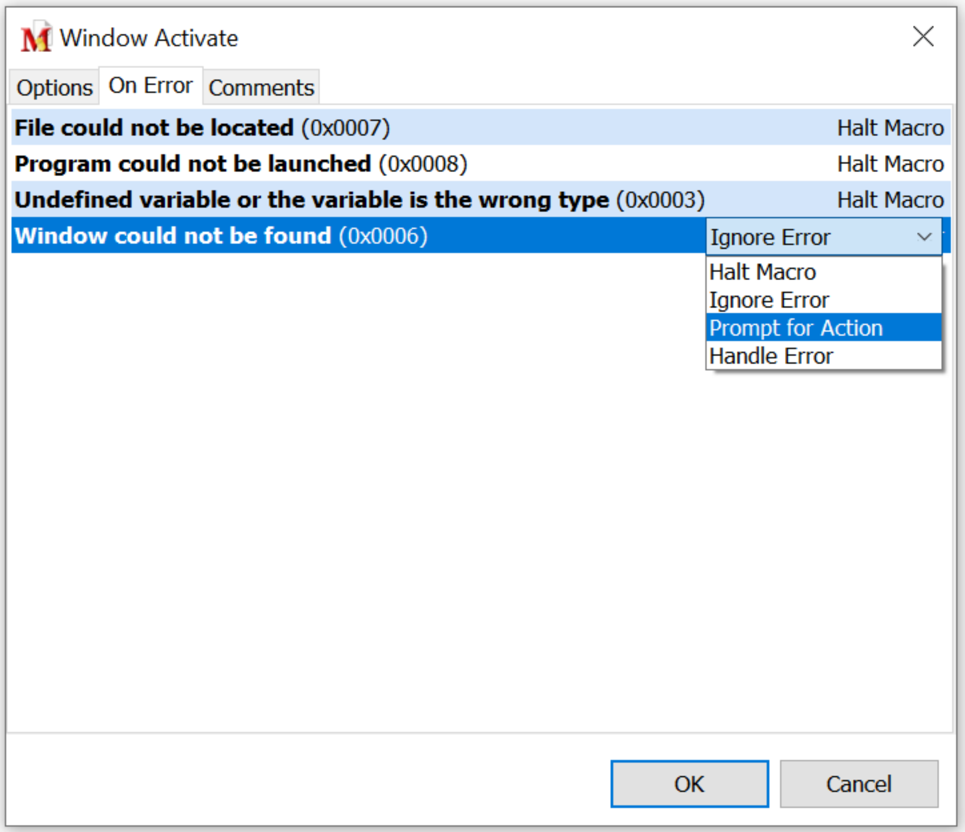
This option requires that the window be on top in order for the macro to continue. Otherwise, the macro will complete it's task whether the window is on top or hidden.

Include hidden windows in the scope

By default the macro will only look for windows that are visible, not those that run hidden and are not seen by the user. Select this option to have the scope apply to the windows listed in the Hidden tab, as well as those that are visible.

On Error Tab

Most macro command windows have a corresponding On Error tab for error handling during playback of the macro. For example if a specific condition isn't met during macro playback, such as a file or a window could not be found, the On Error tab can instruct the macro to stop, ignore the error and proceed, prompt for a course of action, or handle the error.



File could not be located (0x0007)	Halt Macro
Program could not be launched (0x0008)	Halt Macro
Undefined variable or the variable is the wrong type (0x0003)	Halt Macro
Window could not be found (0x0006)	Ignore Error

Script Editor > Open Any Macro Command Window > On Error Tab

In the example above the Window Activate command has four possible types of errors that may occur during macro playback. Each command lists a different set of errors that are pertinent to that specific macro command. In most cases it is appropriate to just leave the default action for each of the errors. But there may be cases for debugging or other situations where one of the other actions is more appropriate. A description of the actions is listed below.

Halt Macro

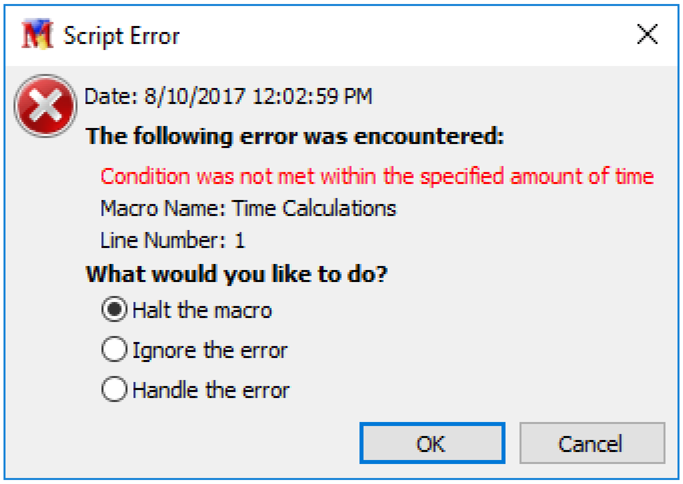
Halt Macro stops the macro if the error occurs during playback. This is the default error action and the one most commonly required.

Ignore Error

Ignoring the error instructs the macro to continue even though the error has occurred.

Prompt for Action

If an error occurs, a message similar to the one below appears on the screen prompting the user to decide how to proceed with the macro. Choose to stop the macro, ignore the error and continue or handle the error if additional error handling code is in the macro script.



Date: 8/10/2017 12:02:59 PM

The following error was encountered:

Condition was not met within the specified amount of time
Macro Name: Time Calculations
Line Number: 1

What would you like to do?

● Halt the macro
○ Ignore the error
○ Handle the error

Handle the Error

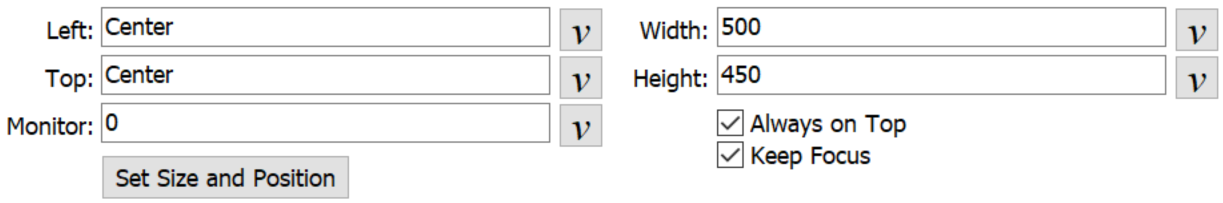
Use this option if subsequent error handling commands are included in the script, such as the On Error command. These subsequent error commands handle the error during macro playback.

Position and Size Macro Express Dialogs

During the playback of a macro, Macro Express may open a dialog box, such as a prompt, multiple choice, text box, etc. Many of these macro commands allow for sizing and/or positioning the dialog box on the screen.

This topic discusses the options to size or position dialog boxes, if the macro command supports the function.

The macro commands that allow for the positioning and sizing of dialog boxes contain edit boxes similar to these displayed below.



Left: Center
Top: Center
Monitor: 0
Width: 500
Height: 450
☑ Always on Top
☑ Keep Focus

Left

The Left edit box positions the dialog a specific number of pixels from the left edge of the screen. In this example the left side of the dialog box is positioned one pixel to the right of the left screen edge.

Top

The Top edit box positions the dialog a specific number of pixels from the top edge of the screen. In this example the top side of the dialog box is positioned one pixel below the top screen edge.

Width

Insert the pixel values for the Width of the dialog to be displayed. In the example above, the text box displays with a width of 500 pixels.

Height

Insert the pixel values for the Height of the dialog to be displayed. In the example above, the text box displays with a height of 450 pixels.

Monitor

This option applies to systems that utilize more than one monitor. If a system has a single monitor, then leave the monitor setting to zero (0). Zero represents the single monitor or the left monitor if there is more than one monitor. One (1) represents a second monitor or the monitor to the right of the left monitor. Similarly, two (2) represents the third monitor or two monitors to the right of the left monitor, and so on.

Always on Top

Always on Top keeps the text box visible on the screen (on top of other running applications) until the box is closed.

Keep Focus

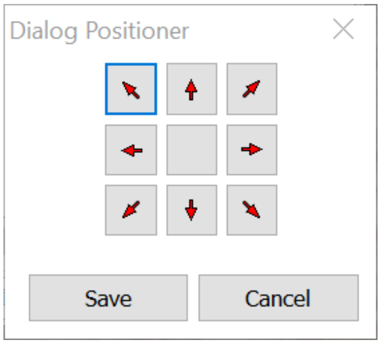
Keep Focus ensures that the text box is the window that has focus. This allows users to immediately input information into the dialog without first giving focus to the text box. Unchecked, this option leaves focus with the application that had focus before the dialog opened.

Set Size and Position

The Set Size and Position button is used to visually set and view the position of the dialog box. Or view and set the size of the dialog by dragging the edges.

Dialog Positioner

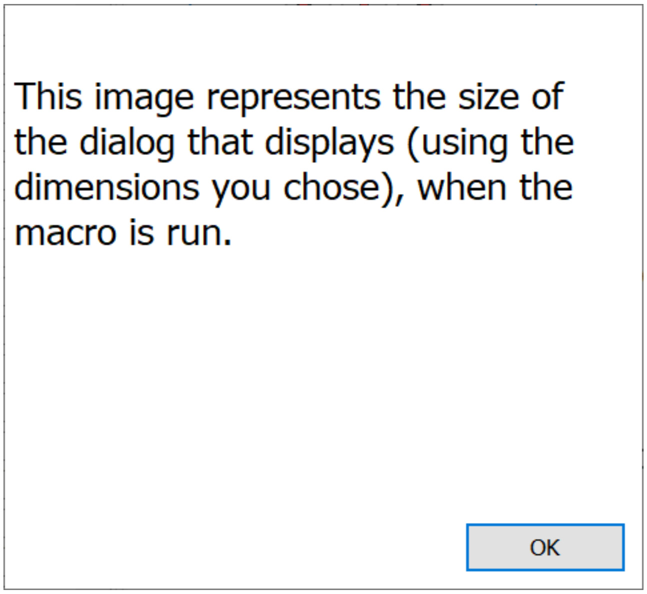
Instead of using pixel coordinates, the window may appear in the center, corners or midpoints of the screen by using the Dialog Positioner. Pressing the Set Size and Position button activates the Dialog Positioner and Dialog Sizer windows.



Click on one of the arrows or the center button. This moves the Dialog Sizer window relative to the button selected. If the arrow button in the upper left corner is selected, the Dialog Sizer window moves to the upper left corner of the screen. Click Save and this is where the dialog displays during macro playback.

Dialog Sizer

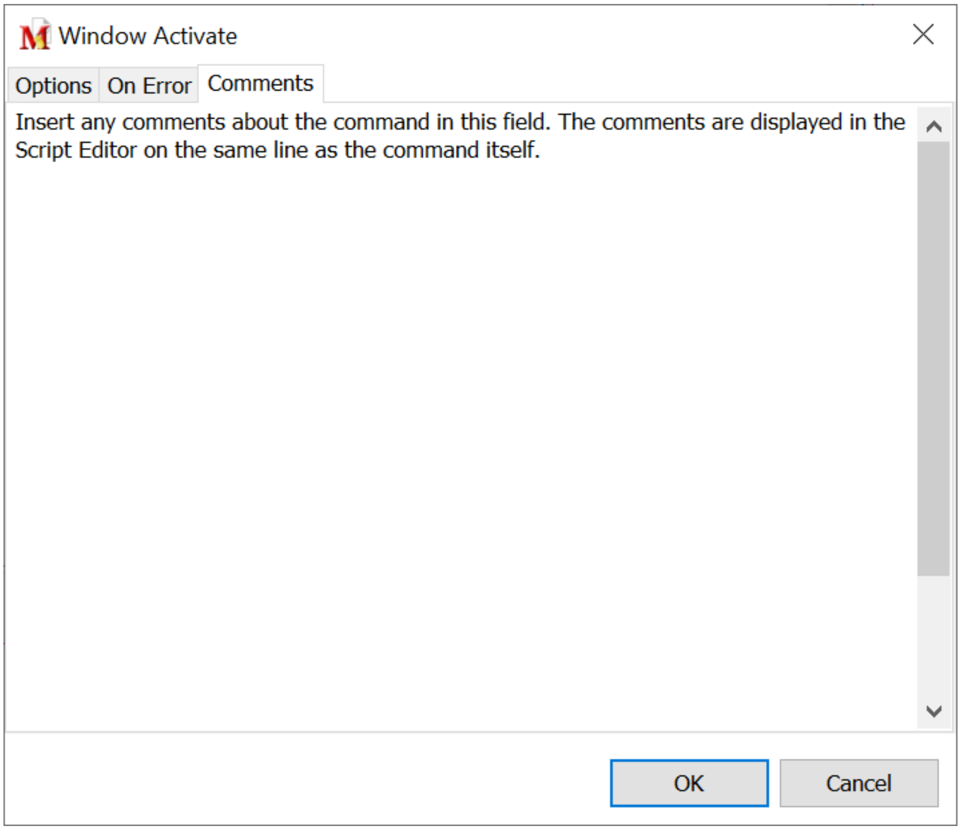
This window displays the text of the dialog box. It is also displayed in the same dimensions as it will appear during macro playback.


This image represents the size of the dialog that displays (using the dimensions you chose), when the macro is run.

Drag the edges of this window to resize it to preferred dimensions. Click OK to save the size. The new size is displayed in the Width and Height edit boxes.

Comments Tab

The Comments tab is found in any of the macro command windows. Insert comments about the command into the edit field - what it is supposed to do, why, etc. The comments appear in the Script Editor on the same line in the script as the command. The comments text displays as blue text to be distinguishable from the command text.



Insert any comments about the command in this field. The comments are displayed in the Script Editor on the same line as the command itself.

Script Editor > Open Any Macro Command Window > Comments Tab

Insert Comments

Or with the Script Editor open, highlight a line in the macro script and click on the Insert Comments button. This option opens a Comments edit box. Type any comments about the line in the script in the this field and click OK to insert the comments next to the command listed in the script.

Network and other topics

Networking Questions

How can I deploy Macro Express across a network?

There is a silent install option for installing Macro Express from a network server. Read the Advanced Installation topic for more information about this process and the command line options available during the install.

How do I run the program over a network?

Place the macro (.mex) file on an accessible network drive. Have each copy of Macro Express point to the file on the network. Use File > Open Macro File to load in the file or set this up during the silent install process.

Click on Options > Preferences > Network from the Macro Express Pro - Explorer and choose the "File Locking of Macro Files" option. This enables the networking capabilities of the program.

File locking prevents overwriting of data by 2 or more users. If a user attempts to change any data in the Macro Express file and the data currently displayed has been altered by another user, the user is notified that the data has been changed and the new data is loaded and displayed.

The **Automatic Synchronization Updating** option is available to allow automatic synchronization of data files between users on a network. This updating can be set to occur every minute, every two minutes on up to every 999 minutes.

For example, if the interval is set to 5 minutes and another user makes a change to the same macro file currently open, then within 5 minutes, the macro data is automatically synchronized between the networked computers.

How do I prevent my network users from changing the macro data?

Use the command line option /B which places Macro Express in a browse only mode. All macros still function, but they cannot be changed by the user.

- ✓ **Note:** Please note that multi-user networking of this product requires multi-user licensing.

Frequently Asked Questions

- ✓ Please visit our Macro Express Knowledge Base for additional topics.

Why are so many of the icons and menu items grayed out in the Explorer window?

Most likely focus is on the System Macros category, with all of the system macros displayed. Select any other macro category and the icons/menu items will be accessible.

How do I place a macro in an endless loop?

The best way to place a macro in an endless loop is to use the Repeat Until Command. Basically, assign a value to a variable such as N[1] = 0. Use the Repeat Until condition to test the variable against a different value such as Repeat Until N[1] = 1. As long as variable N[1] never is changed to 1 during the macro playback, the macro repeats indefinitely. Below is a brief example of what the code would look like in the Script Editor:

```
Set Integer Variable N[1] to 0
Repeat Until N[1] = 1
.... (your macro code here)
Repeat End
```

- ✓ **Note:** Please do not have one macro call another macro (via the "Macro Run" command) and in turn have the second macro call the first macro. Putting two macros in an endless loop in this fashion will eventually end up in a system crash. The system resources used by Macro Express for each macro never get freed in this instance because the macro never actually finishes executing before calling the other macro.

How do I stop a running macro?

There are several ways to do this. If the Running Man icon is displayed in the notification area when a macro runs, right mouse click on the icon to stop the macro. Or press the default keys of Scroll Lock and Pause keys to stop the macro. Click on Options > Preferences > Playback - Miscellaneous Tab to change the default to Win + ` or the Pause key.

Any of these options will cancel a macro that is playing. Please note that macros that consist of many keystrokes may send the keystrokes rather quickly and Windows will buffer up the keystrokes. Once such commands have been buffered up in Windows, aborting a macro does not cancel them. In this cases, it may appear that the abort command did not work when in reality the macro has already finished issuing its commands and Windows is now processing them.

How do I click on a button on a web page?

Once the cursor is positioned over a button (via the tab key), press the Space bar. Use the Text Type command to issue a Space bar press or insert using the Direct Editor.

For example, when inserting name, street address, city, state and zip code followed by a submit button on a web page, the macro may look as follows: "Fred Smith<TAB>123 Apple Lane<TAB>Small Town<TAB>UT<TAB>80000<TAB><SPACE>". When invoked, this macro tabs through the fields, inserting the data and then presses the Space bar to trigger the Submit button. The Space bar key works equally well with checkbox and radio type buttons on web page forms.

How do I ensure the CTRL, ALT or SHIFT keys are held down in a macro?

There are two methods that may be employed. They are listed below.

1. The first method is to use the CTRL, ALT and SHIFT Keys that can be accessed in the Text Type command or through the Direct Editor. By default, anything that follows <CTRL>, <ALT> or <SHIFT> is acted upon as if it were being held down.

For example, if the macro contains <CTRL>a, then this is the same as holding down the CTRL key and pressing the "a" key. If the macro contained <CTRL><ALT>a, then this would be the same as holding down the CTRL key and the ALT key at the same time and then pressing the "a" key. The same applies to the SHIFT key.

✓ Two notes about these keys:

A. Use a lower case letter with these keys, such as <CTRL>a. Macro Express interprets <CTRL>A as <CTRL><SHIFT>a, which renders a different result.

B. There are certain system keystroke commands that are built into the operating system at a low level such as CTRL+ALT+DELETE. These keystroke sequences in macros are not recognized by Windows at this low level and will not work as intended.

2. The second method uses the specific functions for "Alt Key Down", "Alt Key Up", "Ctrl Key Down", etc. These are found in the Text Type command. Choose "Alt Key Down" followed by the letters to be associated with the ALT key. Then follow with an "Alt Key Up" command. Again, be sure to use lower case letters. An example would look like <ALTD>fo<ALTU>.

Make sure to include the corresponding Up key or the key will remain held down and cause unpredictable results. The "WIN" key is also available for down and up usage.

How do I dynamically change the contents of a macro?

There are a couple of ways to accomplish this.

1. One method is the "Text Insertion" option. The macro can load in a text file at any point during playback and execute the text of the file as part of the macro. The dynamic portion of the macro is stored in the text file. Change the contents of the text file to alter the playback of the macro..

2. A second method is to include "Variables" in the macro script. Populate the variable in any number of ways including being prompted, loading from a text file, loading from the clipboard and many others.

How do I get Macro Express to not display when I first start the program? I just want the icon in the notification area.

From the Macro Explorer, click on Options > Preferences > Startup. Make sure the Show the Editor option is not checked.

If the Macro Express icon does not display in the notification area, click on Options > Preferences > General, Notification Area tab and select the Display Icon in Notification Area option.

How can I not have any delays between keystrokes when capturing a macro?

Before capturing a macro click on Options > Preferences > General - Capture tab. Remove the check next to the "Capture at a Specific Speed" option. This captures the macro without any delays or hesitations. Be careful with this option as it may cause the macro to play back too fast and throw off the timing of the macro.

Can I use macros in DOS programs?

Macro Express currently only works with Windows programs. It does not paste keystrokes into a DOS program. The global Windows "keyboard hooks" that we tap into are not relayed to DOS programs even if they are running under Windows.

Can I auto repeat keystrokes?

This question references the "Why can't I hold down the CTRL key and continue to press and release the "a" key for multiple CTRL+A macros to be played?". The real answer deals with the "global" nature of our program -- that is our macros will work in any program at any time. For this to occur, we actually send the keystrokes through the Windows internal keystroke processing system just as if the key were pressed from the keyboard. Because of this, all keys must be released before the macro plays back. This is to ensure that no adverse keyboard effects will affect the playback of the macro.

Let's take an example. Suppose a company name of "My Company, Inc." is assigned to the hot key "Alt-A". Hold down the ALT key and press the "A" key to activate the macro. If the macro were to start playing immediately upon doing this, the ALT key would still be held down and the macro would play back as ALT-M, ALT-y, etc until the ALT key is released. To prevent this from occurring, we require all keys to be released before the macro will start.

My macros do not play back.

There are several possibilities.

1. The simplest and sometimes overlooked by new users is that the Macro Express Player needs to be running for the macros to work.

2. It is possible that another program loaded after Macro Express installs itself improperly into the Windows Keyboard Hooks. When this happens, our program never sees the keystrokes and therefore does not work. (The last program run gets first look at the keystrokes). Your recourse here is to try one of the following:

A. Unload the program that is causing the problem

B. Make sure that Macro Express is run after starting the offending program

C. Use the Tools > Restore Keyboard & Mouse Hooks menu option.

D. Choose Options > Preferences > Activation - General Tab and select Auto Restore Keyboard and Mouse Hooks.

In the latter three cases both Macro Express and the offending program will work as Macro Express properly handles the situation. Ultimately, the vendor of the offending program needs to fix the problem.

3. Make sure that the Macro Scope is set correctly.

4. If you are using the shortkey activation, click on Options > Preferences >

Activations - Shortkeys tab. Make sure that you have the Prefix or Suffix Key option set to the way you expect the program to work.

5. If you have not rebooted since installing, reboot the computer.

6. Check your security software to make sure it allows Macro Express to function properly.

7. Close Macro Express by clicking File > Terminate Macro Express Editor & Player. Locate the shortcut to start the program and right click on it. Select Run as Administrator and follow any UAC prompts to finish loading the program.

My macro does not seem to invoke menu commands.

Some programs appear to interpret ALT keystroke sequences as something other than menu commands. This exhibits itself when certain menu commands work from Macro Express and others do not. There are two methods for working around this problem.

1. Use the ALT Key Down and ALT Key Up keystrokes in the Text Type command or Direct Editor. This option holds down the ALT key while the letter is pressed and then the releases the key. The syntax would look like the following if attempting to execute an ALT f - <ALTD>f<ALTU>.

2. The other option is to use a double ALT (<ALT><ALT>). Two ALTs in a row is the equivalent of pressing the ALT Key and releasing it. Doing so activates the menu and the next letter activates the appropriate menu option. The syntax would look like the following if attempting to execute an ALT f - <ALT><ALT>f.

My macro seems to go bonkers or play out of sequence.

The problem is a timing issue. It is sometimes possible for the keystrokes themselves to be sent too quickly for the desired actions. The program or window may not be ready to accept the keystrokes from the macro. There are several ways to work around this problem.

1. Use one of the Wait commands to instruct the macro to wait until ready to proceed.
2. Insert a Delay at the necessary spot in the script.
3. Use the Keystroke or Macro Speed command to slow down the keystrokes in the macro.

4. Apply a system wide delay to all keystrokes played back via macros.

Command Line Parameters

Command Line Parameters are commands that may be passed to Macro Express. A command line parameter follows the name of the program .exe and is separated by a single space from the name and other command line parameters.

The /A and /MXE command line parameters listed below are used for launching macros.

Use this whether or not Macro Express Pro is already running:
    "C:\Program Files\Macro Express Pro 6 (x86)\MeProc.exe" <command line parameters>

You may also use this if Macro Express Pro is not currently running:
    "C:\Program Files\Macro Express Pro 6 (x86)\MacExp.exe" <command line parameters>

Not all command line parameters may be combined on a single command line. For example, the /IC option imports the program configuration, including the macro file. This cannot be combined with the /F option used to load a macro file.

/A<Nickname> (Activate a Macro)

The Activate Macro command specifies a given macro to run when invoking Macro Express. <Nickname> is the nickname of the macro to run.

For example, if a macro is nicknamed "My Address", use the command line parameter of "/AMy Address" (without quotes). This command can be particularly useful when running macros from batch files. If Macro Express is already running, it will not launch a second copy of the program yet the macro will still be executed. Spaces are allowed in the nickname.

The /A command does not load the macro file that contains the macro to be run. If that macro file is already loaded or will load when Macro Express is started, the selected macro will run. If not loaded, the macro will not execute.

- ✓ **Note:** If more than one macro has the specified <Nickname> only the first macro encountered with that name will run.

/A::<FileName>?<Nickname> (Activate a Macro in a specific macro file)

This Activate Macro command allows you to specify both the macro file and the specific macro to run when invoking Macro Express. <FileName> is the path to the macro file. <Nickname> is the nickname of the macro to run. Spaces are allowed in the pathname of the macro file and in the nickname. For example:

```
    "C:\Program Files\Macro Express Pro 6 (x86)\MeProc.exe" /A::c:\My
Macros\Test Macros.mex?Test macro
```

This command can be particularly useful when running macros from batch files or other programming languages. (See Launching macros from a command line parameter.) If Macro Express is already running, it will not launch a second copy of the program yet the macro will still be executed.

- ✓ **Note:** If more than one macro has the specified <Nickname> only the first macro encountered with that name will run.

- ✓ **Note:** The macro file containing the macro to activate must be loaded into Macro Express in order for the macro to run.

/B (Browse Mode)

This sets Macro Express to a browse mode where the macro data cannot be modified. The Macro Express editor cannot be invoked when this option is used. This option can be especially helpful in administrating uniformity of macros over a network.

/EN (Do Not Run Editor)

This option instructs the Macro Express editor to not run when the Macro Express player first runs.

/EY (Run Editor)

This option instructs the Macro Express editor to run when the Macro Express player first runs.

/F<FileName> (Load Macro Express File)

Use this option to select a specific Macro Express file for loading on startup of Macro Express. <Filename> is replaced with the Macro Express file name. A full path along with the ".mex" extension on the file name must be used. For example:

```
   /Fc:\documents\macro express pro\mymacros.mex
```

- ✓ **Note:** This command can have spaces in the file name and the path.

The macro file specified by the /F command will be added to the list of macro files for Macro Express Pro to open.

If the Close open macro files before loading another macro file preference option is enabled, only the macro specified by the /F command will be opened.

/H (Hide Program)

This completely hides the Macro Express player program. Invoke the Macro Express editor by using the system "Activate Macro Express Editor" hot key.

/IC<FileName> (Import Program Configuration)

This option loads in the program configuration settings on startup of Macro Express. If more than one user shares a computer, each user could have their own configuration settings loaded when they start Macro Express. Each individual user would need to create their own startup shortcut in order to do this.

/MXE (Run .MXE file)

Use this option to run a macro that is contained in a playable macro file (.MXE file). Example: "/MXEc:\program files\macro express pro\mymacro.mxe" (without the quotes).

/N (Network)

This option forces the program to use the Network Mode.

/NQS (No Quick Start)

This option prevents the Quick Start program from running for the current session.

/S (Synchronize Network)

This option is only useful if the Network Mode is active. This option forces the network synchronization option to be used.

/SMxx (Synchronization Minutes)

This option is only useful if the Network Mode is active. "xx" should be substituted with the number of minutes that macro file synchronization should take place. Valid values are from 1 minute to 99 minutes. /SM10 would synchronize the file every ten minutes.

/T (Notification Area)

This option places an icon in the notification area.

/V (Variables)

This option is used for passing variable information into Macro Express and is best used in conjunction with one of the other commands that launches a macro. For example, launch a macro and the variable information is already predefined. Text, Integer or Decimal variable values may be used.

Follow the /V with the Text, Integer or Decimal type variable. Place a colon after the variable name and follow with the data to populate the variable value. Some examples are shown below:

/VMyString:Now is the time - Places "Now is the time" (without quotes) into variable "MyString"

/VN34:150 - Places the value 150 into variable "N34"

/VADecimal:3.1459 - Places the value 3.1459 into variable "ADecimal"

This function places the values into the "Saved Variable Locations". The macro must use the "Variable Restore All", "Variable Restore Text", etc. commands first before it can access the data.

- ✓ **Note 1:** Most of these options allow for a quick and easy setup for network considerations. For example, users can be restricted from changing Macro Express data using the browse mode, allowing manageable uniformity across the network. Users can all access the same Macro Express file using the "Load Macro Express File" option so no configuration is required by first running the program. Network usage can be enforced using the "Network" option, etc.

- ✓ **Note 2:** Insert quote marks around the file path and file name when there are spaces in the path or name. An example of how this would look is shown below.

  "C:\Program Files (x86)\Macro Express Pro 6\MeProc.exe" /AMyMacro

Launching macros from a command line parameter

Below are examples of launching macros from a command line parameter with Delphi, C/C++/Visual C or VBA. Replace Filename?Macroname with the actual path, file name and macro when using this function. An example would be c:\Macro Express Pro\mymacrofile.mex\Play System Sounds.

Example: Delphi

```
       ShellExecute (0, nil, ‘c:\Program Files (x86)\Macro Express Pro
6\MeProc.exe’, ‘/A::Filename?Macroname’, nil, SW_SHOW);
```

Example: C/C++/Visual C

```
        ShellExecute (NULL, NULL, "c:\Program Files (x86)\Macro Express
Pro 6\MeProc.exe", "/A::Filename?Macroname", NULL, SW_SHOW);
```

Example: VBA

```
   Sub CallMEMacro()
       Shell ("C:\Program Files (x86)\Macro Express Pro 6\MeProc.exe
/A::Filename?Macroname")
   End Sub
```

Backup and Restore Macros

Macro Express allows for the creation of multiple files of macros. The macex.mex file is the default file installed. Or create a new macro by clicking on File > New Macro File from the Macro Express Pro - Explorer window.

Backup

There are several ways to back up the macro file that is currently open.

1) Use the automatic backup routine found in Options > Preferences > General > Backup. Backups can be created at scheduled intervals and saved to specific locations.

2) Click on File > Backup Macro File(s). The file is backed up to the default location defined in Options > Preferences > General > Backup.

3) Click on File > Export > Export Macros to activate the Export Macros window. All macros in the file are displayed. Select all or some of the macros and save to a new macro file. This option does not preserve any category information.

4) Use the Copy and Paste options of Windows Explorer or My Computer. Navigate to the .mex file, copy and save to a different location.

Restore

1) From the Macro Express - Explorer window, click on File > Restore. A Windows dialog will open prompting to select the backup file. The dialog opens to the folder designated in the Backup preferences panel for file backups. If the backup options were recently changed, then it may be necessary to browse to the location where previous backups were stored.

Performing the restore closes the file currently open and replaces it with the backup file selected.

2) Click on File > Open Macro File to load in the backup macro file or a copy of the file. Browse to the location where the backup or copy resides and open the file.

3) Click on File > Import > Import Macros to activate the Import Macros window. All macros in the .mex file selected are displayed. Highlight all or some of the macros in the file and import them into the macro file currently open.

Macro Express Pro End-User License Agreement (EULA)

Agreement

The use of this Software indicates your understanding and acceptance of the following terms and conditions. If you do not understand or accept these terms you must cease using this product immediately.

Copyright

This product is Copyright © by Insight Software Solutions, Inc. All Rights Reserved. This Software is protected by United States copyright law and various international treaties.

License

You may use an unlicensed copy of this Software for up to 30 days for evaluation purposes only. After the 30 day evaluation period you must purchase a license to continue to use the Software.

Once you have purchased a license you will receive a license code that will enable you to continue using the Software on a non-exclusive basis. This license will not expire.

The Software is licensed, not sold. Ownership of the Software, Documentation and all intellectual property rights therein shall remain at all times with Insight Software Solutions, Inc. In licensing the Software you do not acquire any rights beyond the right to continue to use the Software on the system for which you purchased the license.

Personal License: A single-user ("Personal") License may either be used by a single user who uses the Software personally on one or more computers, or installed on a single computer used by multiple people, but not both.

Multi-User License: A multiple-user ("Multi-User") License may be purchased based upon either (a) the number of users who have access to the Software on any number of computers, or (b) based upon the number of computers on which the Software will be installed for use by any number of users. You may not use the Software in excess of either (a) the number of purchased users or (b) the number of purchased computers, whichever is applicable. A Multi-User License may not be used by your subsidiary companies, customers, or any other third parties.

You shall be responsible for maintaining the License in a safe location and are specifically prohibited from distributing the License, intentionally or unintentionally, to any third party. Upon loss or distribution of the License, Customer shall be required to pay a reinstatement fee at Insight Software Solution's discretion.

License Transfer

You may not sell, rent, lease, or transfer the Software to any third party and may not lend the Software other than as authorized by this EULA.

You may transfer the product to another party only if the other party agrees to all the terms and conditions of this agreement and if you provide Insight Software Solutions notice of the transfer including your name, company (if applicable), and address and the name, company (if applicable), and address of the person to whom you are transferring the rights granted herein. If you transfer the program you must at the same time transfer or destroy any back-up copies. You may not retain any portion of the program, in any form, under any circumstance. If the Product is an upgrade, any transfer must include all prior versions of the Product. You must secure written approval and a personalized license code from Insight Software Solutions, Inc. The license code issued to the initial user will be invalidated.

If the Software was purchased as part of a Multiple User license, you may only sell or transfer the entire multiple user license; you may not sell or transfer a portion of a Multiple User license. You may not sell or transfer any Trial Version or Not For Resale Version of the Software. The sale or transfer of the Software does not transfer the 30 day money back guarantee offered to the original purchaser.

Restrictions

You may not copy, modify, translate, loan or create derivative works based upon the Software or any part thereof. You may not reverse engineer, decompile, adapt, or disassemble the Software, nor shall you attempt to create the source code from the object code of the Software. You may not attempt to unlock or bypass any licensing algorithm utilized by the Software. You may not distribute the license code you received when purchasing the Software or post this information to discussion forums, web sites, email messages or servers. You may not remove or modify any copyright notices. Violations will result in termination of this license.

Limited Warranty

Insight Software Solutions, Inc. warrants that the Software will perform substantially in accordance with the user documentation for a period of ninety (90) days from the date of receipt. This limited warranty is void if failure of the Software has resulted from accident, abuse, or misapplication. Any replacement Software will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

Disclaimer of Warranty

THERE ARE NO OTHER WARRANTIES RESPECTING THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY REPRESENTATIONS OR ENDORSEMENTS REGARDING THE USE OF, THE RESULTS OF, OR PERFORMANCE OF THE PRODUCT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY, RELIABILITY, OR THAT THE SOFTWARE IS ERROR FREE. THE ENTIRE RISK AS TO THE USE OF THIS SOFTWARE IS ASSUMED BY THE USER AND/OR LICENSEE. INSIGHT SOFTWARE SOLUTIONS, INC. DOES NOT ASSUME LIABILITY FOR THE USE OF THIS SOFTWARE BEYOND THE ORIGINAL PURCHASE PRICE OF THE SOFTWARE. IN NO EVENT WILL INSIGHT SOFTWARE SOLUTIONS, INC. BE LIABLE FOR ADDITIONAL DIRECT OR INDIRECT DAMAGES INCLUDING ANY LOSS OF PROFITS OR OTHER INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES ARISING FROM ANY DEFECTS, OR THE USE OR INABILITY TO USE THE SOFTWARE, EVEN IF INSIGHT SOFTWARE SOLUTIONS, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO LIMITATIONS MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION.

Customer Remedies

The entire liability of Insight Software Solutions, Inc. and your exclusive remedy shall be, at the option of Insight Software Solutions, Inc., either (a) return of the original purchase price paid for the Software, or (b) repair or replacement of the Software that does not meet the Limited Warranty and that is returned to Insight Software Solutions, Inc. with a copy of your receipt.

Export Controls

None of the Software, Documentation or underlying information or technology may be downloaded or otherwise exported or re-exported (i) into (or to a national or resident of) any country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Orders. By downloading or using the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list.

High Risk Activities

The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). Insight Software Solutions, Inc. specifically disclaims any express or implied warranty of fitness for High Risk Activities.

Terms

This license is effective until terminated. You may terminate it by destroying the Software, the documentation and copies thereof. This license will also terminate if you fail to comply with any terms or conditions of this agreement. You agree upon such termination to destroy all copies of the Software and of the documentation, or return them to Insight Software Solutions, Inc.

Other Rights

No amendments or modifications may be made to this Agreement except in writing signed by both parties. If one or more provisions of this Agreement are found to be invalid or unenforceable, this Agreement shall not be rendered inoperative but the remaining provisions shall continue in full force and effect.

No Insight Software Solutions, Inc. dealer, distributor, or agent is authorized to make any modification or addition to this Agreement.

All other rights not specifically granted in this license are reserved by Insight Software Solutions, Inc.

Governing Law

This agreement shall be governed by the laws of the State of Utah. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded.

Support

Important: Before requesting support, please ensure that the answer to your question is not already contained in the Help File, the Frequently Asked Questions or the web based Knowledge Base for Macro Express. For questions or comments regarding Macro Express please visit our support page at www.macros.com/support.htm.

If you are experiencing an issue with Macro Express, please provide the following:

- The product name and version number. (Click on Help > About for version information.)
- The version of Windows you are running (7, 8.1, 10, etc.) including service packs.
- The sequence of actions that produces the problem.
- Any other relevant details.

Before contacting us for technical support, please make sure that you are using the latest version of Macro Express, as the problem may have been fixed in a later update. Information about the latest release may be found at www.macros.com/download.htm.

Licensing the Evaluation Version

Macro Express Pro is available as an evaluation copy. The evaluation copy is fully functional for a fixed period of time. To use the program beyond the evaluation period requires purchasing a license.

To purchase a license open the Macro Express Pro - Explorer window and click on Help > Buy Macro Express. Or right click on the Macro Express icon in the notification area and left click on Buy Macro Express. Either method opens the order page on our web site to display several purchasing options.

After purchasing the license for Macro Express Pro you will receive a license code. From the Macro Express Pro- Explorer click on Help > License Macro Express to activate the License Information window. Enter the name and code exactly as they appear on the license instructions received after purchase. Press OK to save the license information. The license status will now display in the lower left corner of the Macro Express Pro- Explorer window.

After successfully entering the license code, the program is fully licensed and will not expire. The Buy Macro Express links mentioned above now change to License Information.

It is recommended that the program be licensed while logged on as the administrator. See the Advanced License Information topic for consideration of Administrator and non-Administrator login rights and Macro Express licensing.

Upgrade Policy

A license key is supplied when Macro Express Pro is purchased. Please keep this key in a secure place as it will be needed if reinstalling Macro Express Pro.

This license key is valid for any version 6.x of Macro Express Pro. Future revisions that bear the "6.x" identifier version are generally for minor enhancements and bug-fixes. Any release containing major new functionality will bear a different version identifier (e.g., 7.x) and an upgrade fee will be chargeable.

The prime location for new releases of Macro Express Pro is our download page at www.macros.com/download.htm.

If installing a newer version of the product, install it over top of the existing version (into the same folder). Close Macro Express completely before installing a new version.

- ✓ **Note:** Uninstalling and then reinstalling Macro Express does not affect the macro file(s). The macro file(s) remain intact. However, as always, we recommend backing up the macro (.mex) file(s) before uninstalling or reinstalling.

Portable Version

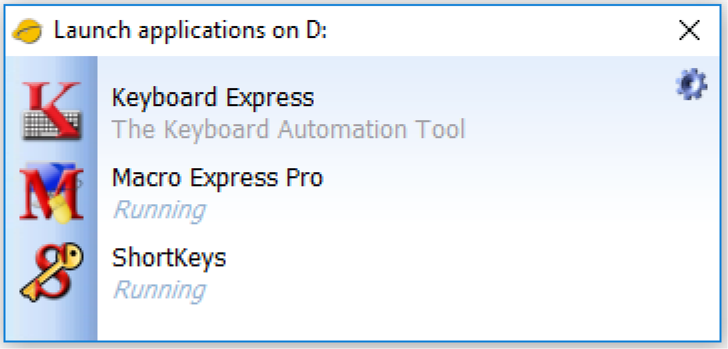
A portable version of Macro Express Pro is available for use on a USB drive or similar portable device. Install Macro Express Pro Portable on the device and run it from that device rather than the local computer. This version will not leave traces of the program on the hard drive of the machine where the program is run.

Macro Express Pro Portable includes the following differences from the standard version.

- No 30-day trial available.
- The options to import or export the program preferences have been removed.
- There is no uninstaller. Use Windows Explorer to delete the files.
- The "Run when Windows starts up" preference option is not available.
- The "Save settings as defaults for all users" preference option has been removed.
- There is no support for playable (.mxe) macros.
- The .mex file association is not linked to Macro Express Pro Portable. You cannot double-click on a .mex file to load it into the program.
- Default locations for the .mex macro file, temp directory, etc. are stored on the portable drive.

To Run Macro Express Pro Portable

When a portable drive is connected to the computer, the AutoPlay menu will typically appear. Click on ‘Open folder to view files’ in the AutoPlay menu. Double click on LaunchApps to display a menu similar to the one shown below. The menu displays the programs you have installed that are recognized by LaunchApps. Click on the program that you want to load.



Or open File Explorer to the portable drive and double click on LaunchApps to open the menu.

AutoRun

You may be able to specify a program to automatically run when the portable drive is inserted into your computer. The file autorun.inf allows this to happen (if it is supported by your version of Windows). Autorun.inf will only load one program at a time and may be disabled if the AutoRun feature in Windows is disabled.

When Macro Express Pro Portable is installed it will create an autorun.inf file in the root directory of your portable drive if one does not already exist. If your portable drive already contains an autorun.inf when you install you may choose to edit it.

The autorun.inf file will load the program LaunchApps. LaunchApps may be configured to automatically load more than one program.

The file autorun_ISS.inf demonstrates how autorun.inf could be set to load LaunchApps.

To close Macro Express Pro Portable

Before instructing Windows to eject the portable drive, close Macro Express Pro Portable and any other open programs. Macro Express Pro interfaces with Windows at a low level and each program that is running loads the Macro Express hook into its memory space. Even after closing Macro Express Pro other programs may still have the hook loaded. If this is the case, Windows will display a message saying the drive is currently in use when you attempt to eject the drive.

If closing all open programs does not release the hook you will need to log off of Windows and then remove the portable drive.

Index