

# Macro Toolworks

Family Products



# QUICK TUTORIAL

Applies to

 **Macro Toolworks**  
and derived products

 **Perfect Keyboard**

 **Clicky Mouse**

 **WinScheduler**

 **MacroToolbar**

# Contents

---

1	Introduction .....	3
2	What is Macro?.....	4
2.1	What is Macro? – Simple Text Insertion Macros .....	5
2.2	What is Macro? – Commands Only Macros .....	7
2.3	What is Macro? – Combined Text Insertion and Commands .....	8
3	Program Basics .....	9
4	Adding Macro.....	12
4.1	Configuring Macro: Macro Properties Tab .....	13
4.2	Configuring Macro: Keyboard Triggers .....	14
4.3	Configuring Macro: Mouse Triggeres.....	15
4.4	Configuring Macro: Scheduled Triggeres.....	16
4.5	Configuring Macro: Toolbar Button (Trigger).....	17
4.6	Configuring Macro: Simple Text Insertion Macro .....	18
4.7	Configuring Macro: Text Insertion Macro with Commands .....	19
4.8	Configuring Macro: Commands Only (Almost) Macro.....	20
4.9	Configuring Macro: More Macro Text Details .....	21
4.10	Configuring Macro: Macro Commands Editing.....	22
4.11	Configuring Macro: More Macro Text Details .....	23
4.12	Configuring Macro: Macro Scope .....	24
5	Adding Clipboard Macro.....	25
6	Recording Macro .....	27
7	Adding Macro Group.....	29
7.1	Configuring Macro Group: General Properties.....	30
7.2	Configuring Macro Group: Scope.....	31
7.3	Configuring Macro Group: Before/After Macro.....	32
8	Adding Macro Toolbar .....	33
9	Program Settings.....	34
9.1	Program Settings: Keyboard .....	35
9.2	Program Settings: Macro Toolbar .....	36
10	Other Information .....	37

# 1 Introduction

---

This tutorial is not intended to go to all the details and all the functionality the Macro Toolworks family products offer. It is rather focused on basic functionality and information. It should help you to quickly understand the program basics and help you to easily create and use your first macros. The tutorial will guide you through these topics:

- What is the macro? How it works? This section should help you to understand what your computer activities can be automated using this macro program.
- Program basics - how the program looks like, how the macros are organized in the program and how they are saved on hard drive.
- Adding macro – how to add macro and how to configure it.
- Adding clipboard macro – there is also special kind of macros.
- Recording macros – macros can be also recorded.
- Adding macro group.
- Adding macro toolbar.
- Program settings.

## 2 What is Macro?

---

Macro has several different meanings in several different contexts. We define macro as a set of steps that mimic what a user is doing on computer while working on some task. The macro can be then used to do user task automatically - in whole or at least partially. Typically the macro is started by user some convenient way - like by a hot-key or click on a macro toolbar button - or it is scheduled to run without user's interaction.

*Example 1:* Simple macro just inserts larger (several paragraphs) predefined text to e-mail, document, chat, etc. when user hits a hot-key or types a text shortcut.

*Example 2:* Macro copies document files modified during last week, put's all of them to an archive file and send's the archive to a remote server. Macro is scheduled to run every Friday night.

*Example 3:* On a macro toolbar button click a macro is started. It displays a form to allow user to insert a few input data and based on them the macro does much more complex steps (for example: updates a file, sends the file via e-mail as an attachment and then fills a report web form).

Macros are mostly useful when the user is doing tasks repetitively. In such case macros provide several advantages:

- Saves **time** because computer is ususally much faster then human.
- Improves **accuracy** because computer is not doing mistakes.
- Improves overall **performance** because tasks are done faster and in better quality.

Users typically create three types of macros:

1. Simple Text Insertion Macros
2. Commands Only Macros
3. Combination of the two above

## 2.1 What is Macro? – Simple Text Insertion Macros

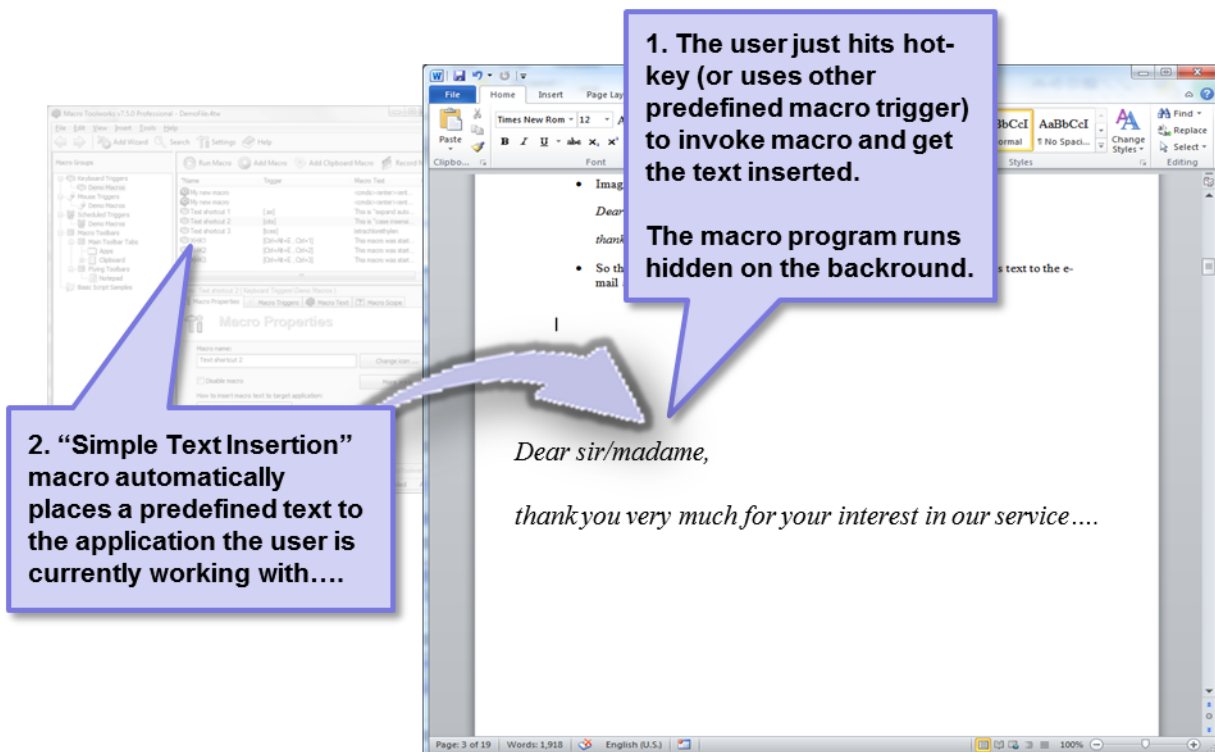
One kind of macros that users use a lot is a “Simple Text Insertion” macro. Such macro just places predefined text to the application that the user is currently working with. The scenario is like this:

- Imagine the user is very often writing e-mails that start with:

*Dear sir/madame,*

*thank you very much for your interest in our service....*

- So the user creates a macro that will “type” this text to the e-mail automatically, on the hot-key that user assigns to the macro as the trigger. \*



The question is how this text insertion is done. There are two possible ways how the text can be inserted to the application:

### 1. Mimic user typing on keyboard

- if this option is used then the application will be receiving sequence of keystrokes the

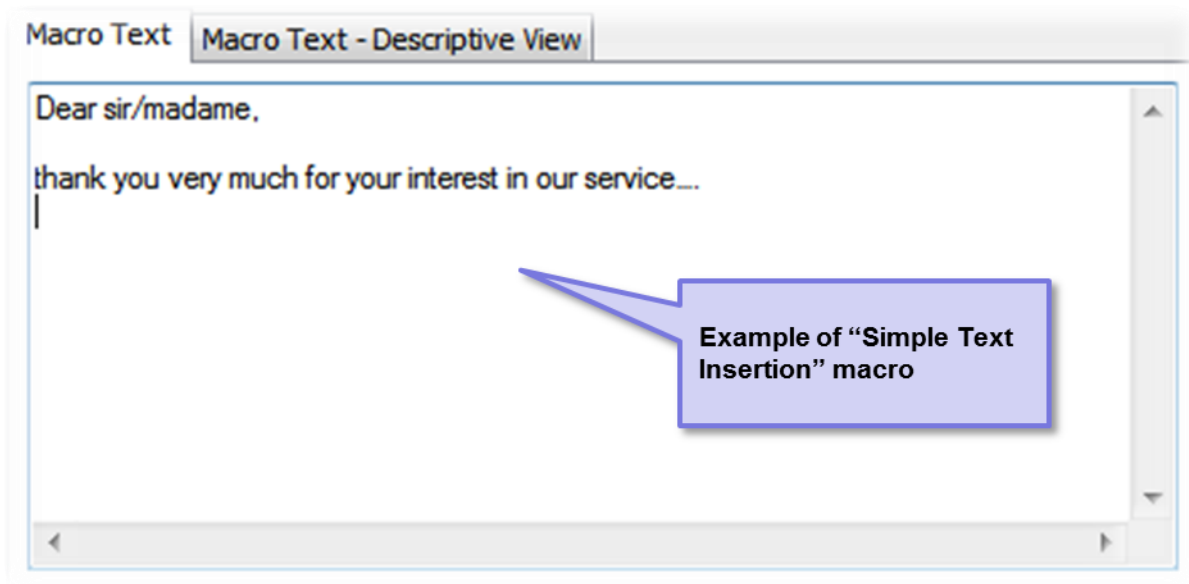
same as if user is typing on the keyboard.

## 2. Mimic user doing copy/paste

- if this option is used then the text is inserted to the application through clipboard – the text is passed to the clipboard and "paste" operation is then automatically invoked in the application.

Each of the above options is more suitable in different cases and it is up to the user to choose which one is better for particular macro. If the option #2 can be used then it is usually much faster than the option #1, however, if some application does not support clipboard operations then option #1 must be used.

*Note: Simple Text Insertion macro is an example of a type of macros that users typically create. It does not mean that Macro Toolworks products are limited just to such simple macros.*



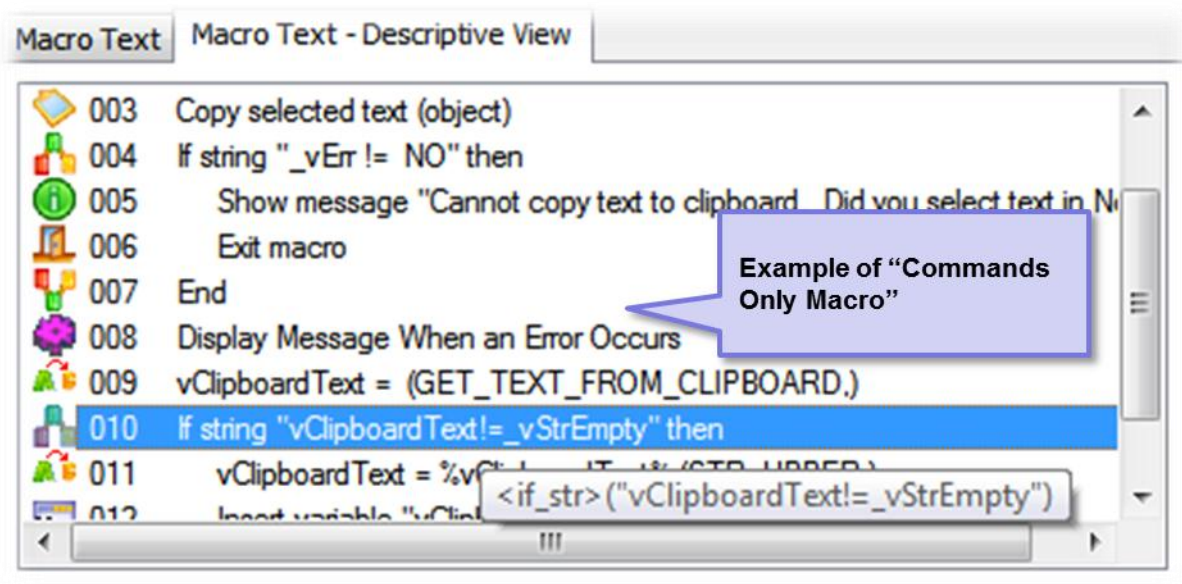
*\* Hot-key and text shortcut triggers are only supported in Macro Toolworks and Perfect Keyboard programs.*

## 2.2 What is Macro? – Commands Only Macros

In opposite to simple text insertion macros there are macros that very often run without user interaction and do background tasks. For example macros that copy/move/delete files/folders, download/uploads file from to servers, modify files, send e-mails, access web pages, etc.

There are a lot of macro commands that allow writing quite complex macros suitable for wide range of IT tasks.

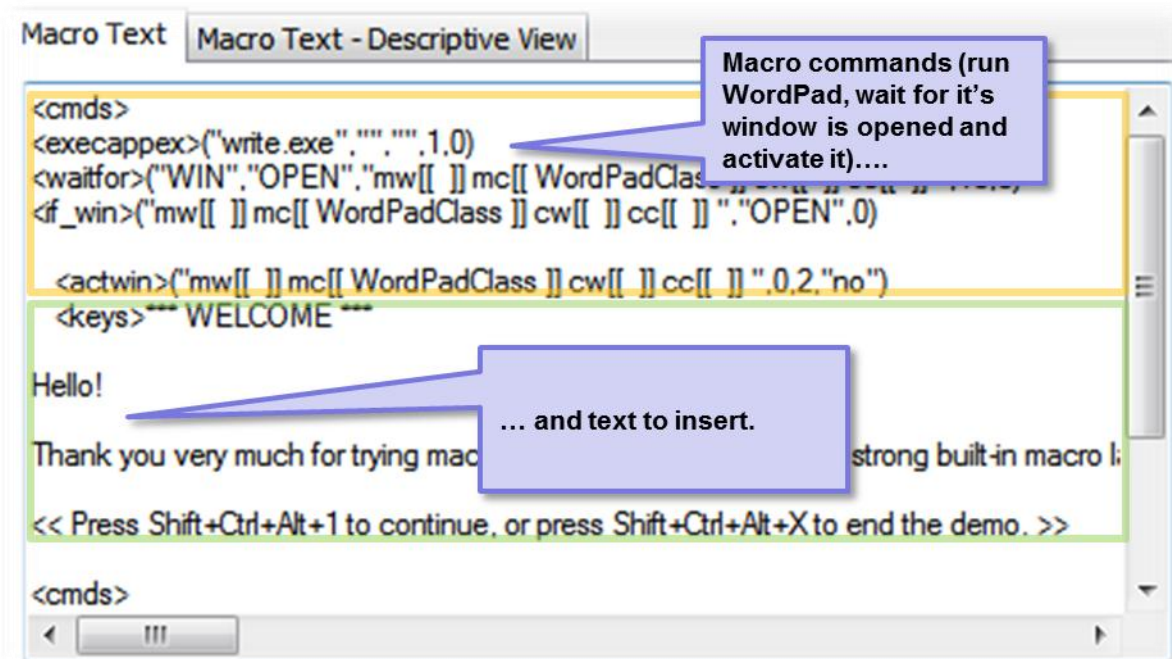
Such macros are very often triggered without user's interaction: at a specified time (for example scheduled to run on weekly bases) or when a specific event occurs (a file is modified/deleted/created, etc.).



## 2.3 What is Macro? – Combined Text Insertion and Commands

---

Of course it is possible to combine text insertion with macro commands without any inconveniences. Very often during the text insertion the macro asks user for some input data through build in “form” command or the macro reads the insertion text dynamically from a file, etc.



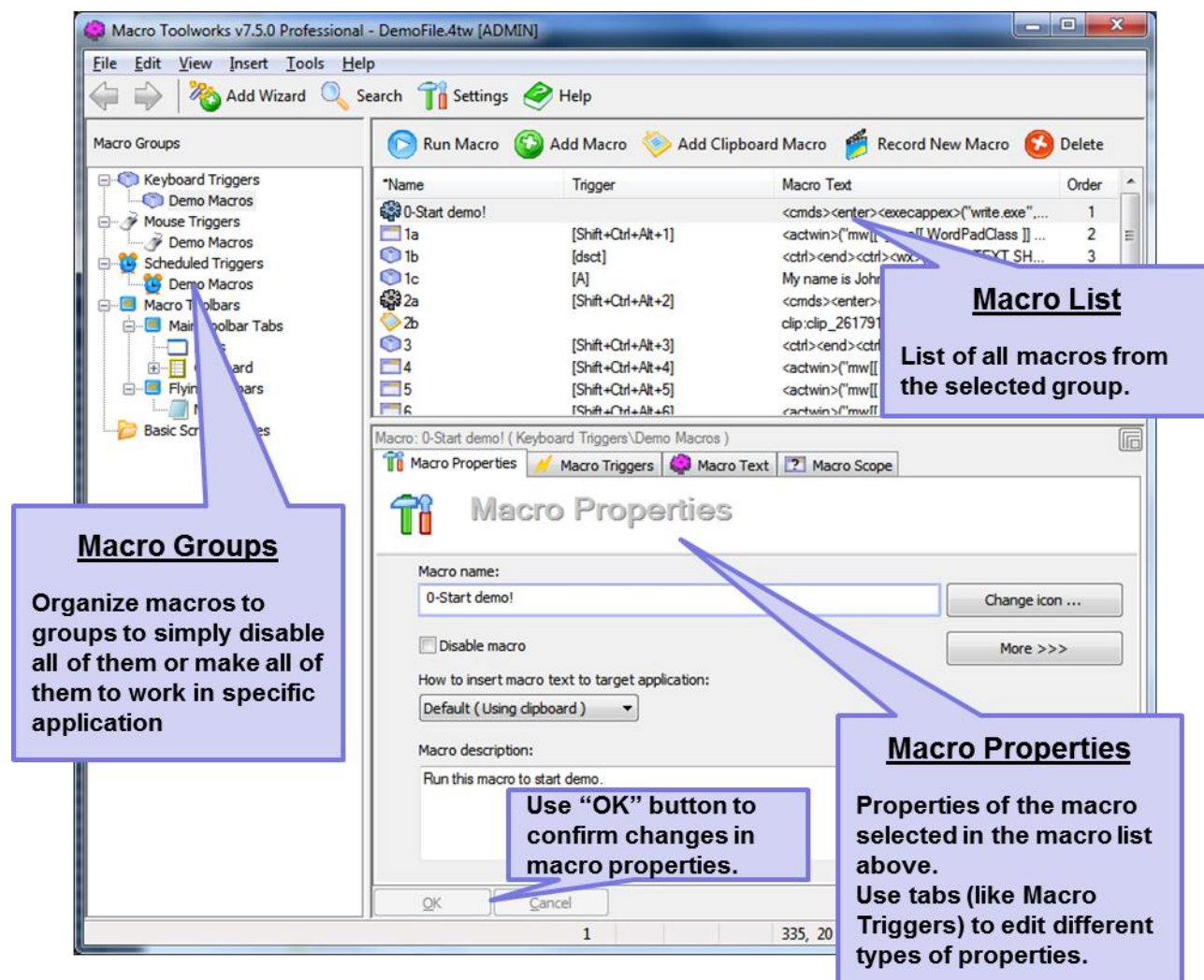
See more about this topic in later chapters concerning macro configuration.



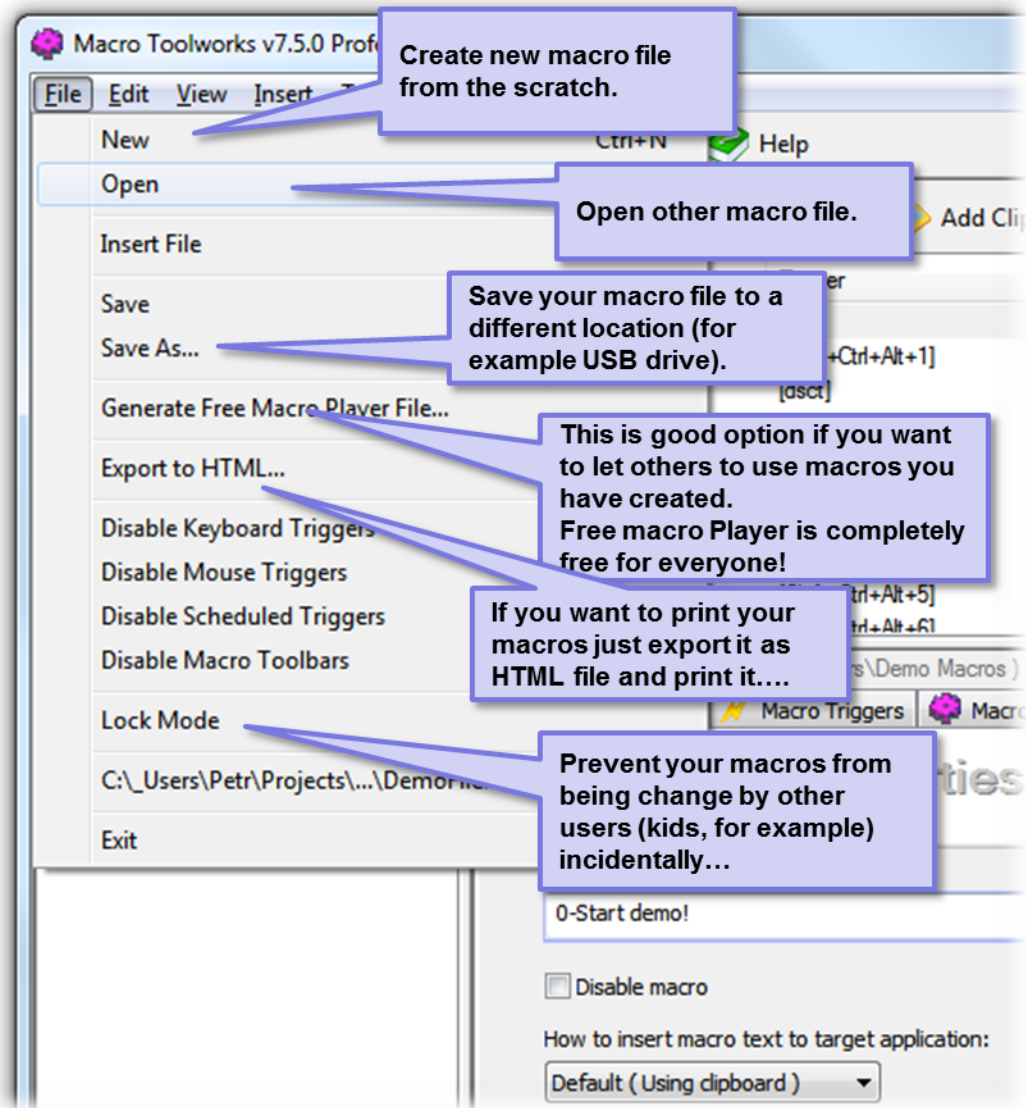
### 3 Program Basics

The macros as described in previous chapters are in the Macro Toolworks organized to macro groups that you can freely create/copy/move/delete. The program main window consists of three sections:

- **Macro Groups** where the groups are organized as a tree.
- **Macro List** that shows all macros from the currently selected group.
- **Macro Properties** that displays properties of the currently selected macro. All the properties are organized in several tabs - Macro Properties, Macro Triggers, Macro Text, and Macro Scope.

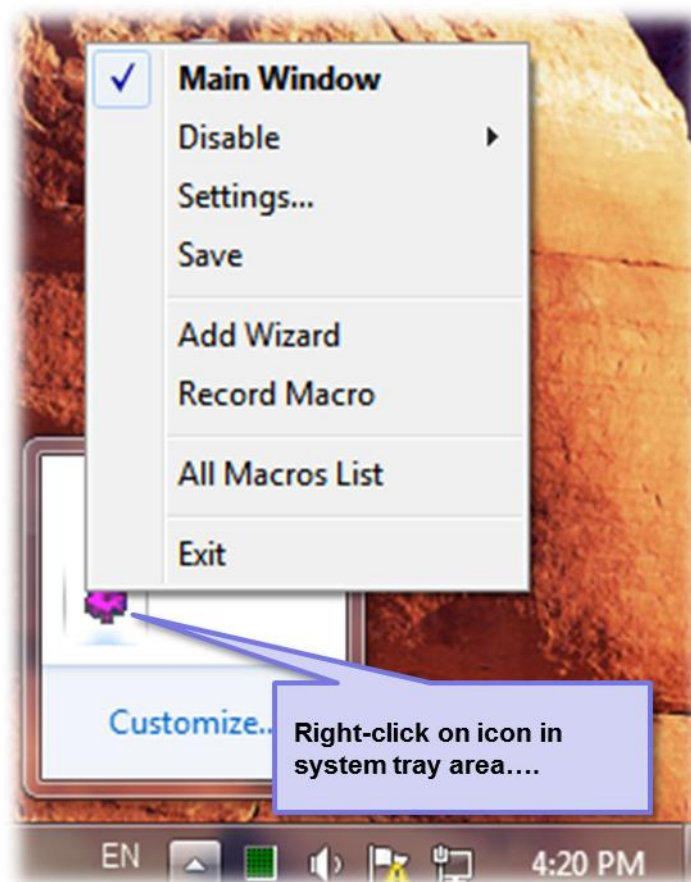


All the macros and macro groups are saved in a single macro file that you can open in Macro Toolworks using “File/Open” menu command the same as you do for example with a text file in text editing software. The same as in other programs; file operations (and others) are available through program menu.



Menu also appears if you right-click on macro group in the tree or on macro in macro list. Through these right-click menus you can access macro properties window, disable macro groups, export macros to text file (and import them from the text file back), etc.

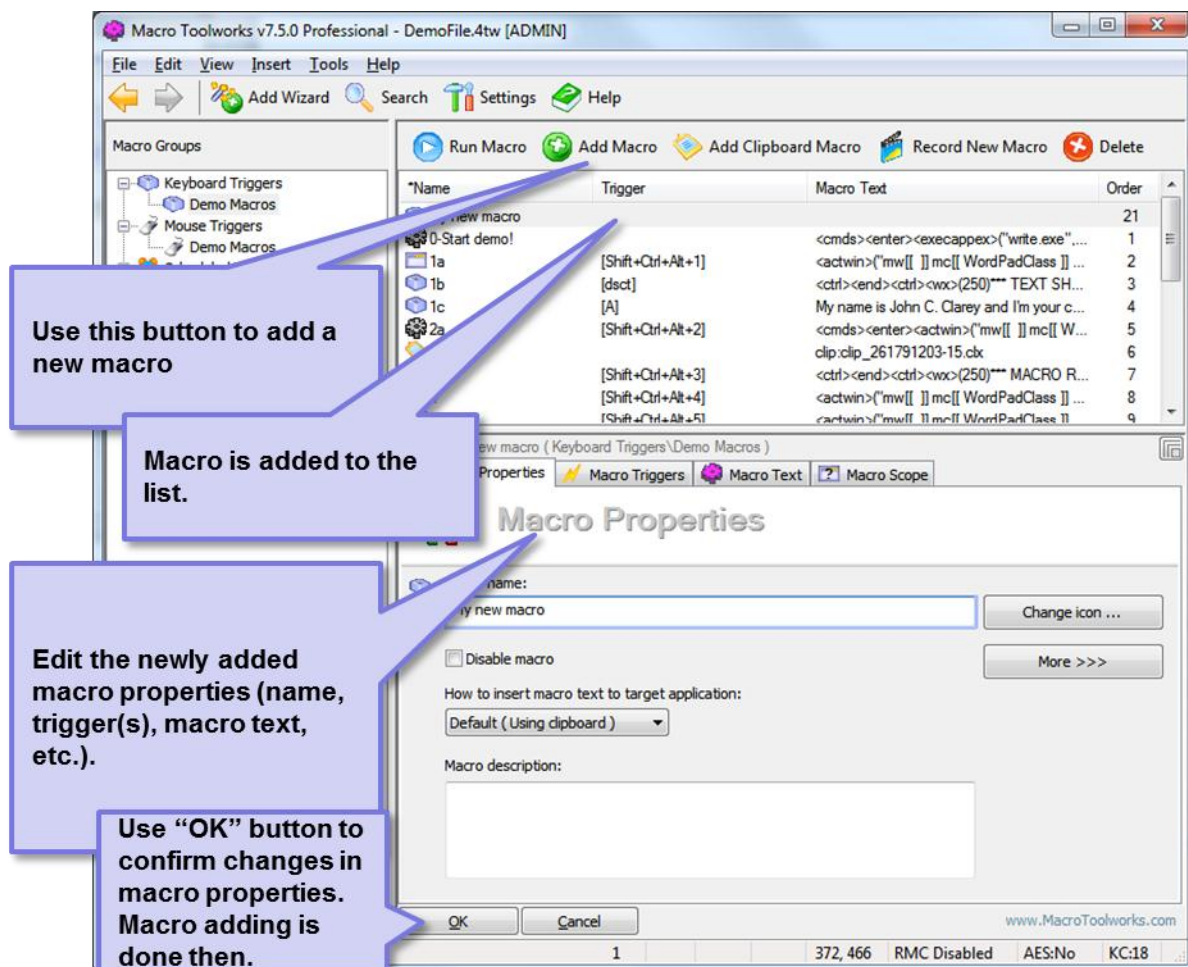
The program icon shows in Windows tray area. Left click cause the program main window is open on screen; if you right-click on the icon then a menu is shown.



## 4 Adding Macro

There are several ways how to add a new macro. The easiest way is to click “Add macro” button as shown on the figure below:

1. Click “**Add Macro**” button
2. Notice the new macro was added to the list.
3. Configure newly added macro in “Macro Properties” area.



The “Macro Properties” area for the macro configuration consists of several tabs – Macro Properties, Macro Triggers, Macro Text and macro Scope. Each of the tabs is described on following pages.

## 4.1 Configuring Macro: Macro Properties Tab

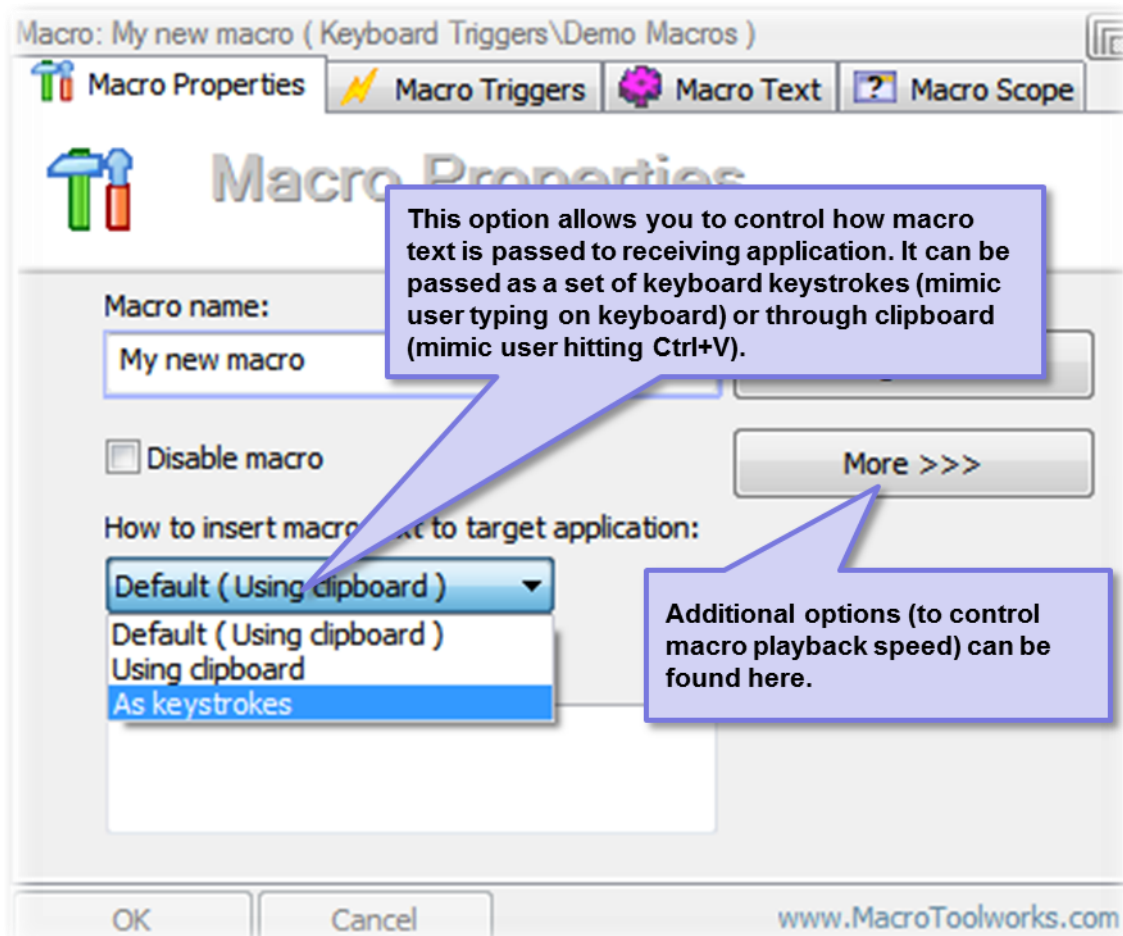
There are several basic macro properties in this “Macro Properties” tab: Macro name, macro icon and several others (playback speed, number of repetitions, etc.) are under “More >>>” button.

There is one important option – called “How to insert macro text to target application” - that defines how the macro insertion text is played back (as mentioned in “What is Macro? – Simple Text Insertion Macros” chapter). Macro text insertion playback can work two possible ways:

3. Mimic user **typing on keyboard**
4. Mimic user doing **paste operation from clipboard**

*Example:* If the macro is very simple “Hello!” text then the option #1 means that the application where the macro is playing back will be receiving sequence of keystrokes the same way as if user is typing “Hello!”. In the same case the option #2 means that the application will receive the “Hello!” text through clipboard – like the macro text is copied to clipboard and then “Ctrl+V” is hit.

This option is important if user has strong preference on how to pass the text to application (for example if the application is not capable of working with clipboard and cannot do “paste” then option #1 from the options above must be used.)

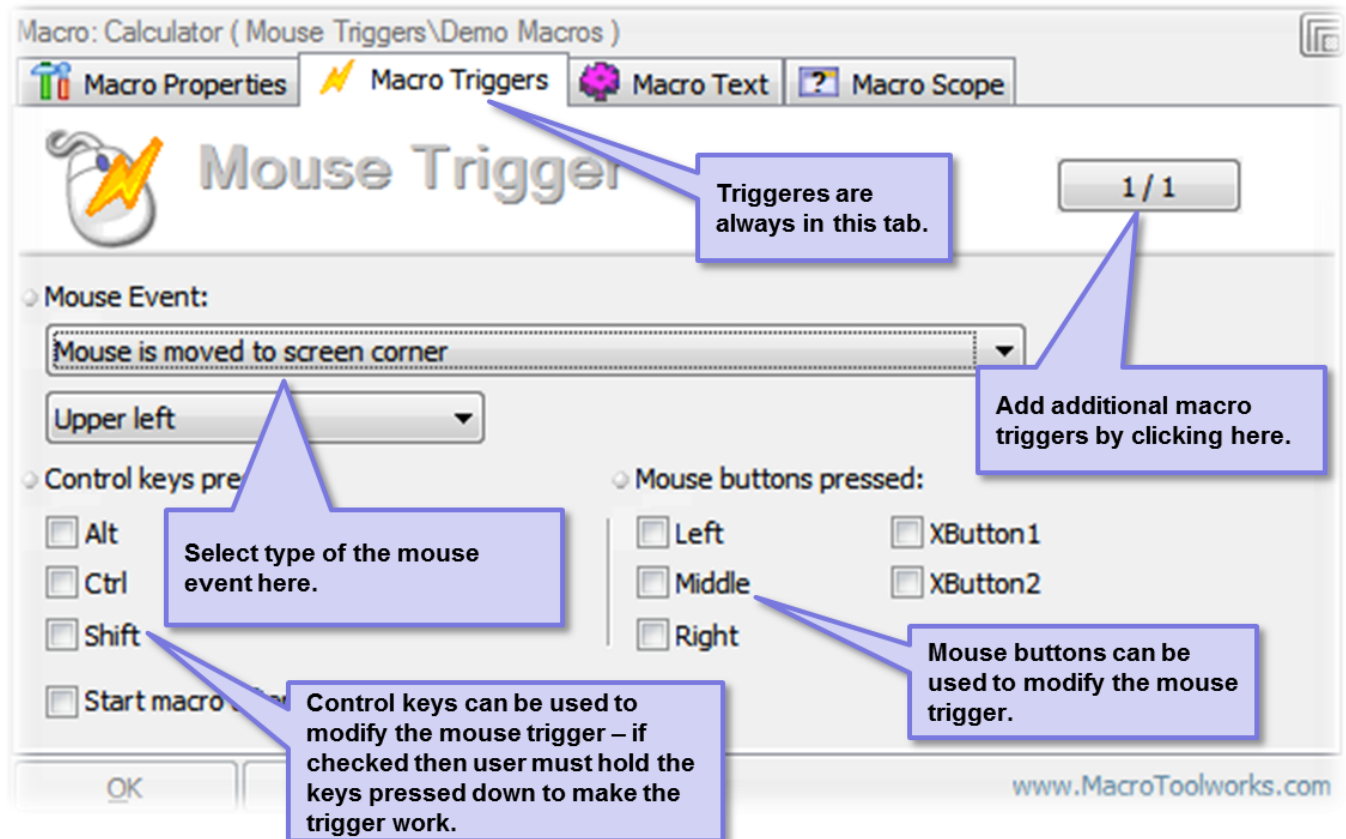






## 4.3 Configuring Macro: Mouse Triggeres

Several different types of mouse triggers can be defined in Macro Triggeres tab as shown on the figure below. *Note: The mouse triggers are only available in Macro Toolworks and Clicky Mouse products.*

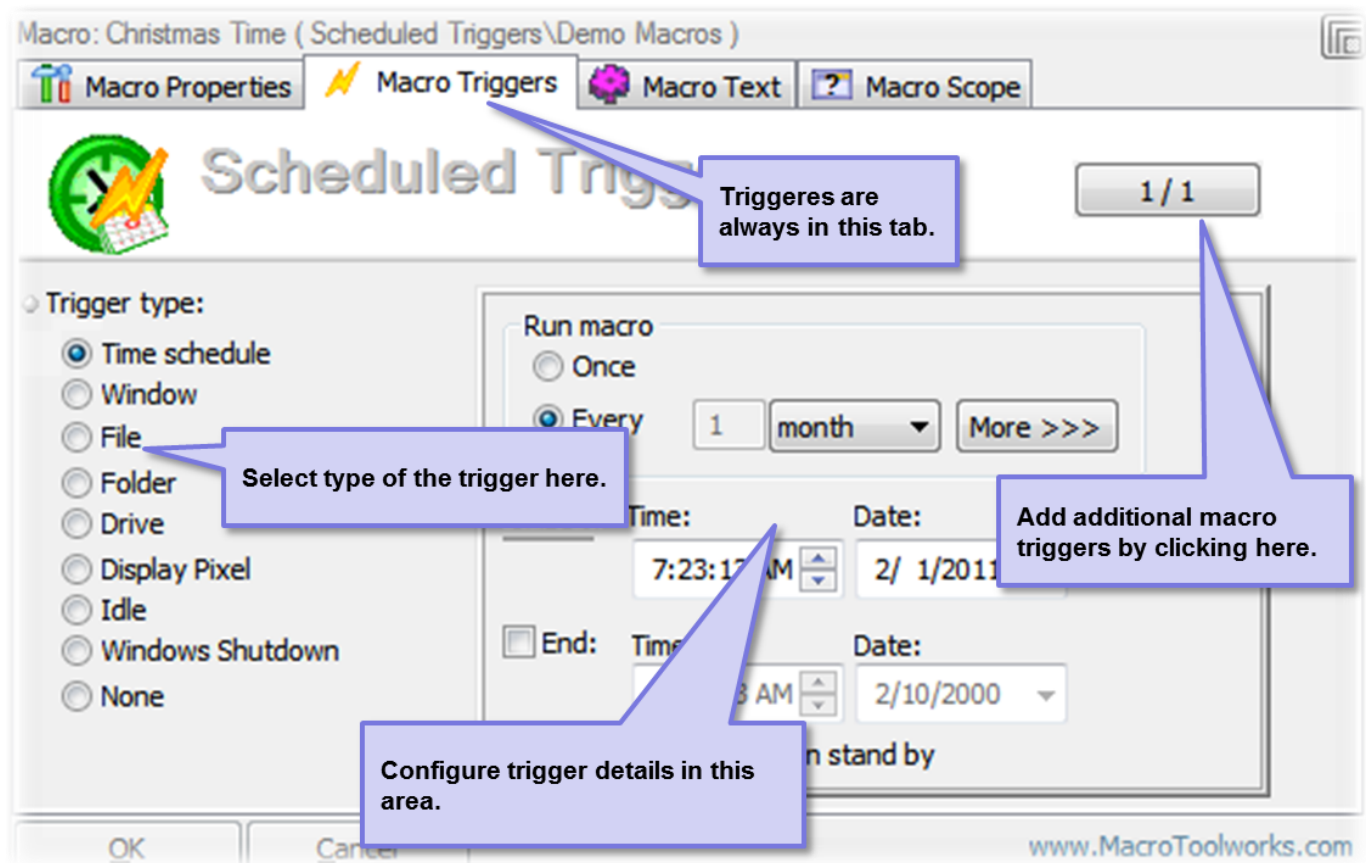


## 4.4 Configuring Macro: Scheduled Triggeres

Several different types of schedule triggers can be defined in Macro Triggeres tab as shown on the figure below. *Note: The schedule triggers are only available in Macro Toolworks and WinScheduler products.*

The schedule triggers can start macro on:

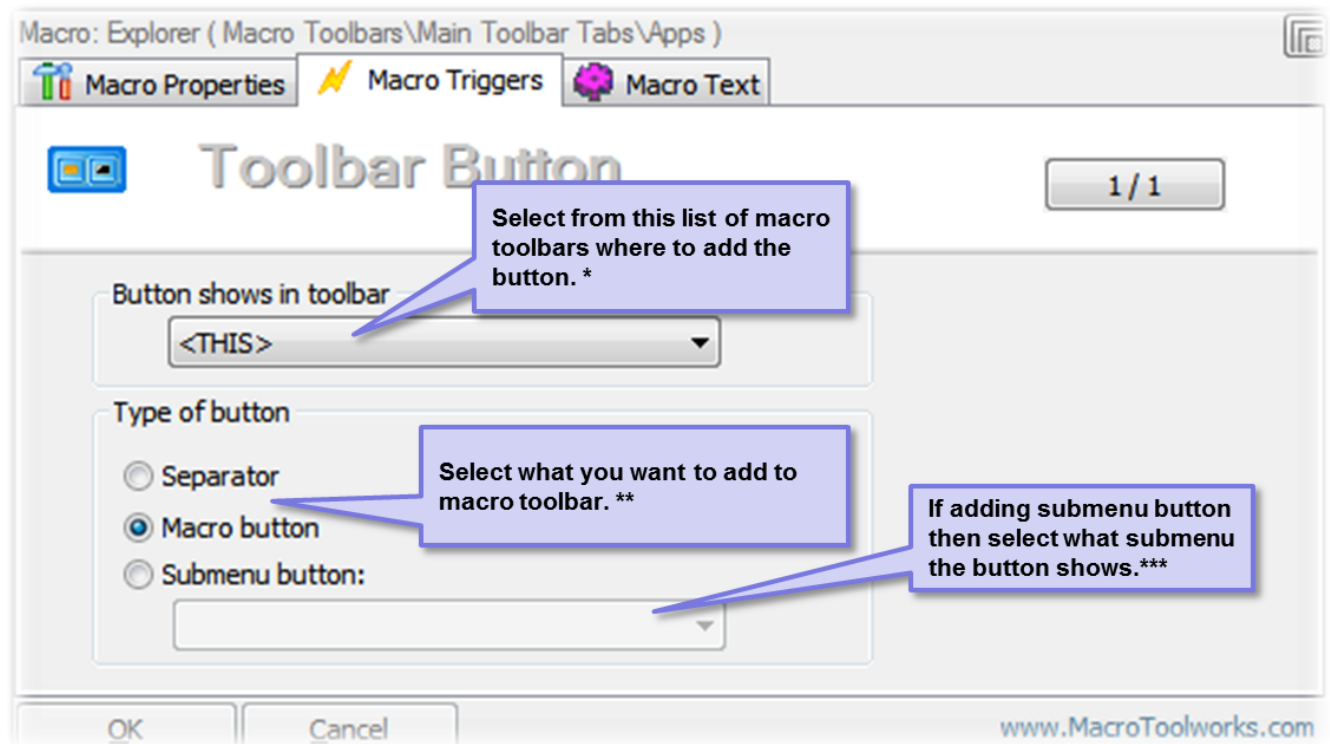
- **Time** - run once, every minute/hour/day/week/month,...
- **Window** is opened/closed/activated/deactivated.
- **File** is created/deleted/modified/bigger than, etc.
- A new file in **folder** is created/deleted/modified or subfolder is created/deleted.
- **Drive** runs out of space.
- **Pixel** on display chnages color.
- Computer is in **idle time**.
- Windows **shutdown**.





## 4.5 Configuring Macro: Toolbar Button (Trigger)

Macros can be added as buttons to Macro Toolbars and menus (described in "[Adding Macro Toolbar](#)") as shown on the figure below. *Note: The macro toolbars are only available in Macro Toolworks and MacroToolbar products.*



\* Macro Toolbar is represented by a macro group (see more about "[Adding Macro Toolbar](#)"). If you want to add macro from this group to the Macro Toolbar represented by this group then use **<THIS>**.

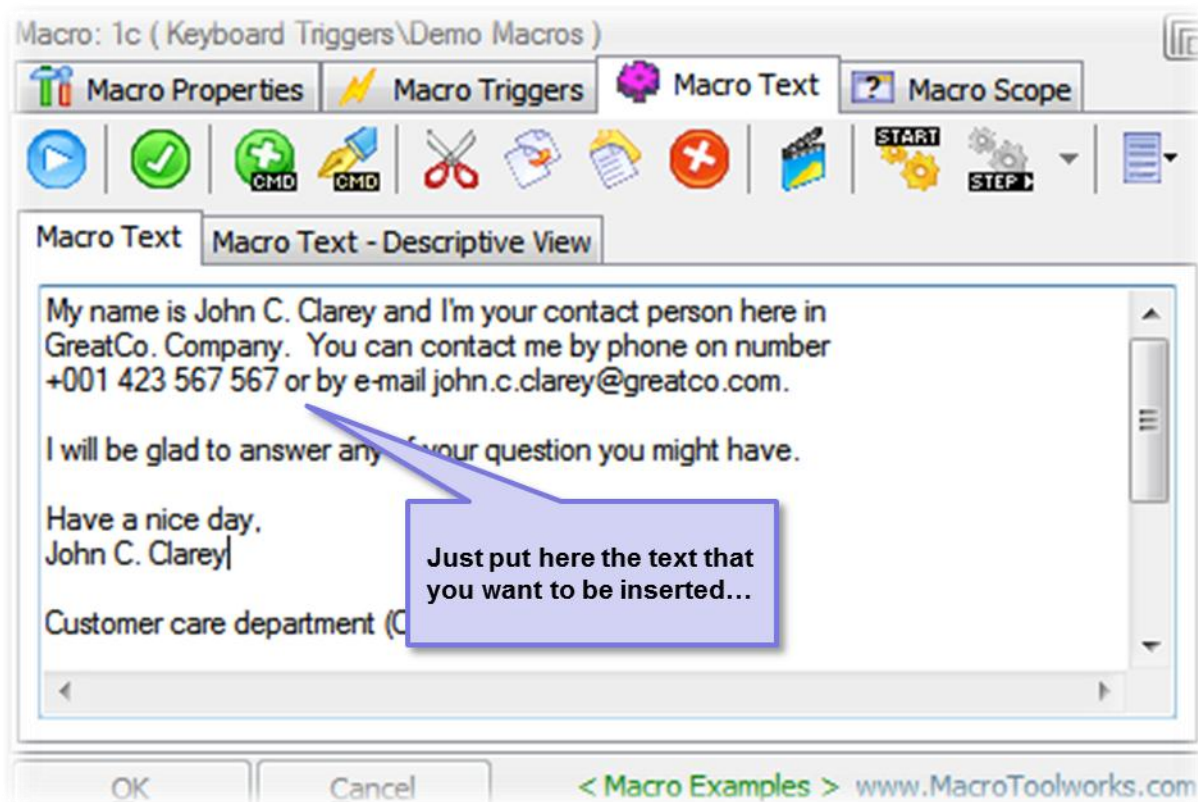
\*\* Macro button, when clicked in Macro Toolbar, runs the macro defined in "Macro Text" tab. If "Separator" or "Submenu button" option is selected here then "Macro Text" should be left empty since it is not used in this case anyway.

The button icon and text is defined in "Macro Properties" tab ("Change icon" button and "Macro name" field).

\*\*\* Submenu, the same as Macro Toolbar, is represented by a macro group (see more about Macro Group Properties).

## 4.6 Configuring Macro: Simple Text Insertion Macro

The simplest macros are those that just insert some text to other applications. People typically define such macros to speed up e-mailing/chatting or basically writing of any kind. Type the text that you want to use for insertion in the “Macro Text” tab as shown on the figure below.



At that point we have working macro and can recap how to get here. The necessary steps are:

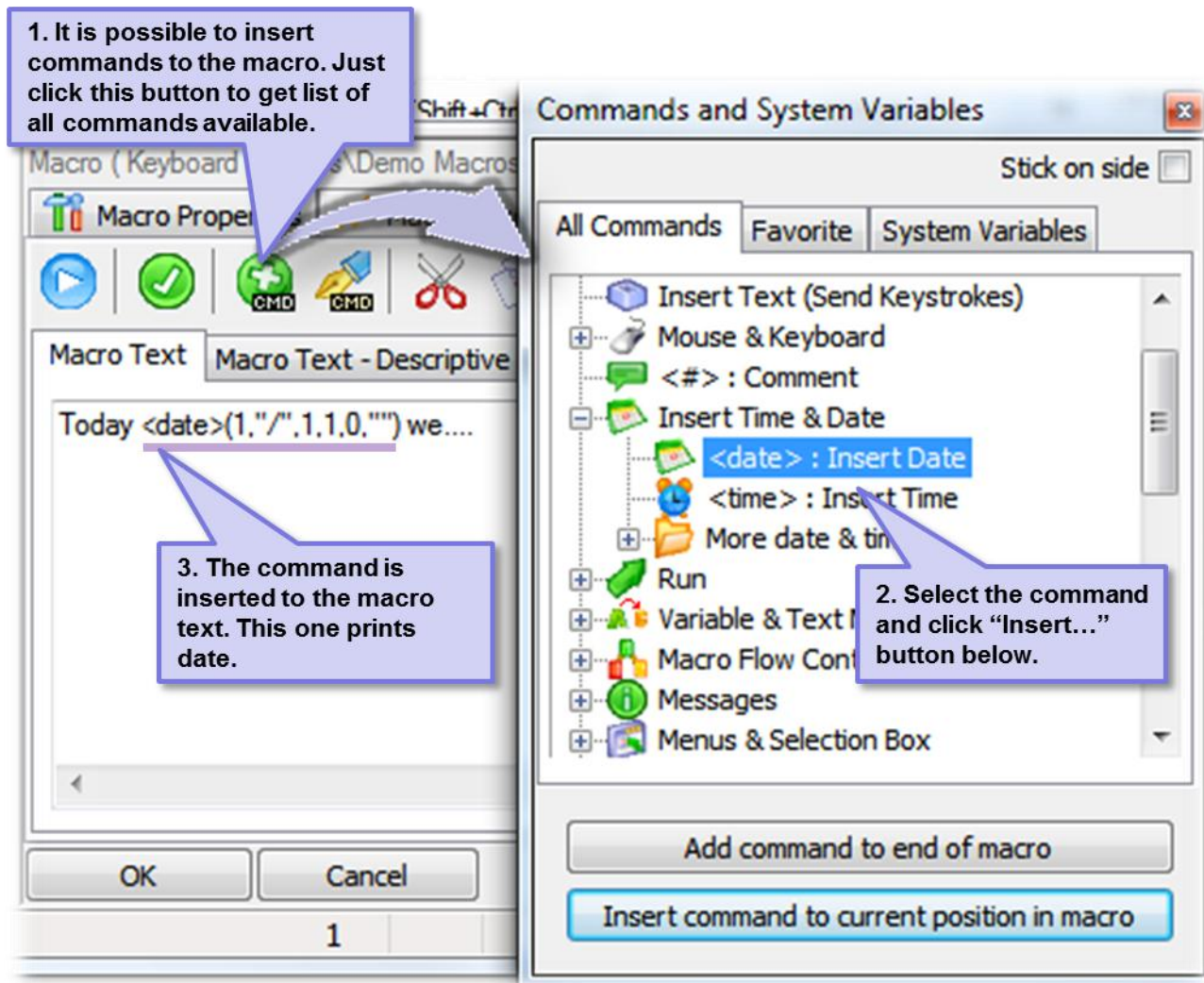
1. Add Macro by clicking “Add Macro” button.
2. Define a macro trigger in the “Macro Triggers” tab. For simple text insertion macros – that you typically use while writing something - this is usually a keyboard trigger (hot-key or text shortcut \*).
3. Put the text that you want to use for insertion into “Macro Text” tab and click OK button. Done.

Now if you in other application use the trigger (such a hotkey \*) the text is automatically inserted.

*\* Hot-key and text shortcut triggers are available only in Macro Toolworks and Perfect Keyboard products.*

## 4.7 Configuring Macro: Text Insertion Macro with Commands

Macros can be much more complex than the simplest one showed as an example on the previous page. It is possible to include macro commands inside the macro. There are many different kinds of commands; all of them are available on a button click as shown on the figure below.



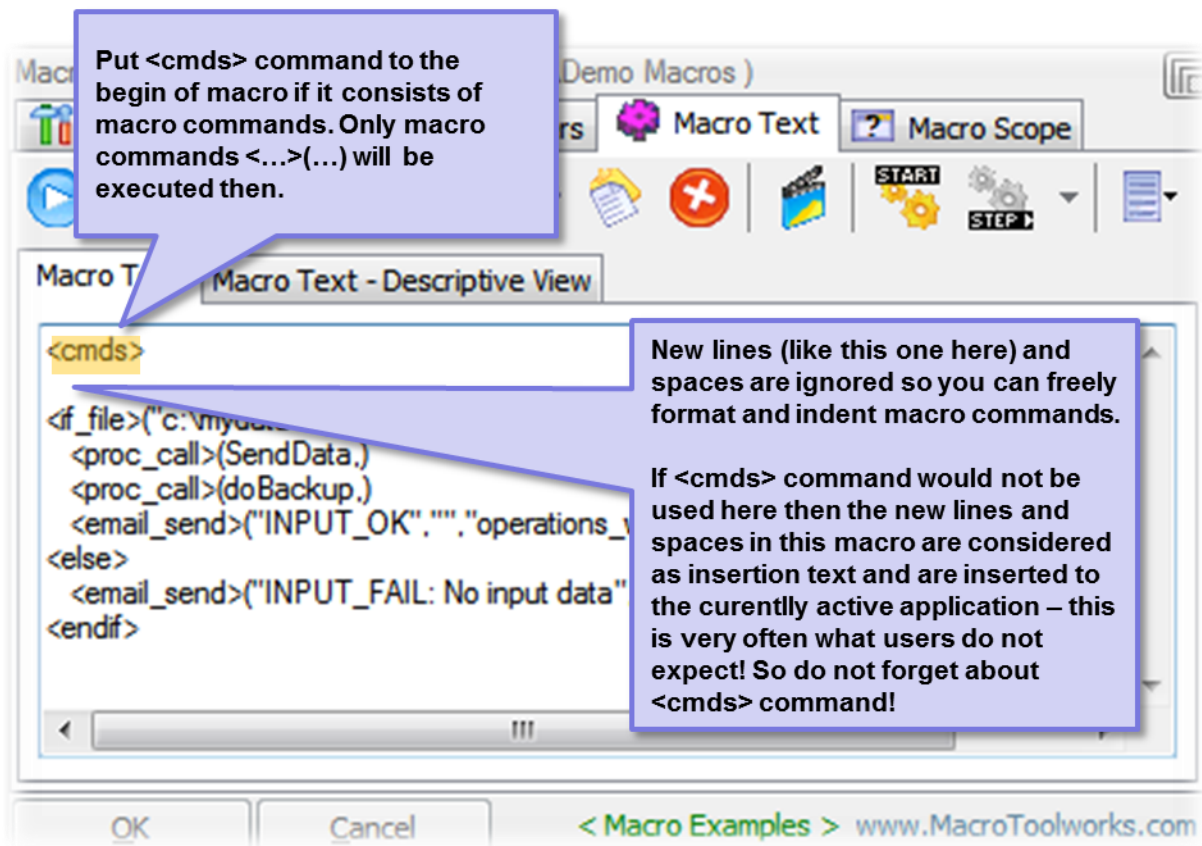
In the text insertion macro there are usually used macro commands to insert information like date/time, or commands to collect user inputs to parametrize macro flow, or commands to wait until something specific happens (like an application window is open or file is deleted, etc.) or, well, others....

Macro commands have this syntax:

**<command\_name>(parameter1, parameter2,...)**

## 4.8 Configuring Macro: Commands Only (Almost) Macro

In opposite to insertion macros, there are macros that mostly (only) consist of macro commands. Such macros usually do a background jobs like copying/moving files, sending e-mails, etc. and do not include any interaction with user. Such macros should have **<cmds>** as the first command before any other command is used. It ensures that only macro commands are executed and what ever other text that could be considered as an insertion text (including new lines and spaces in the macro text) is ignored.



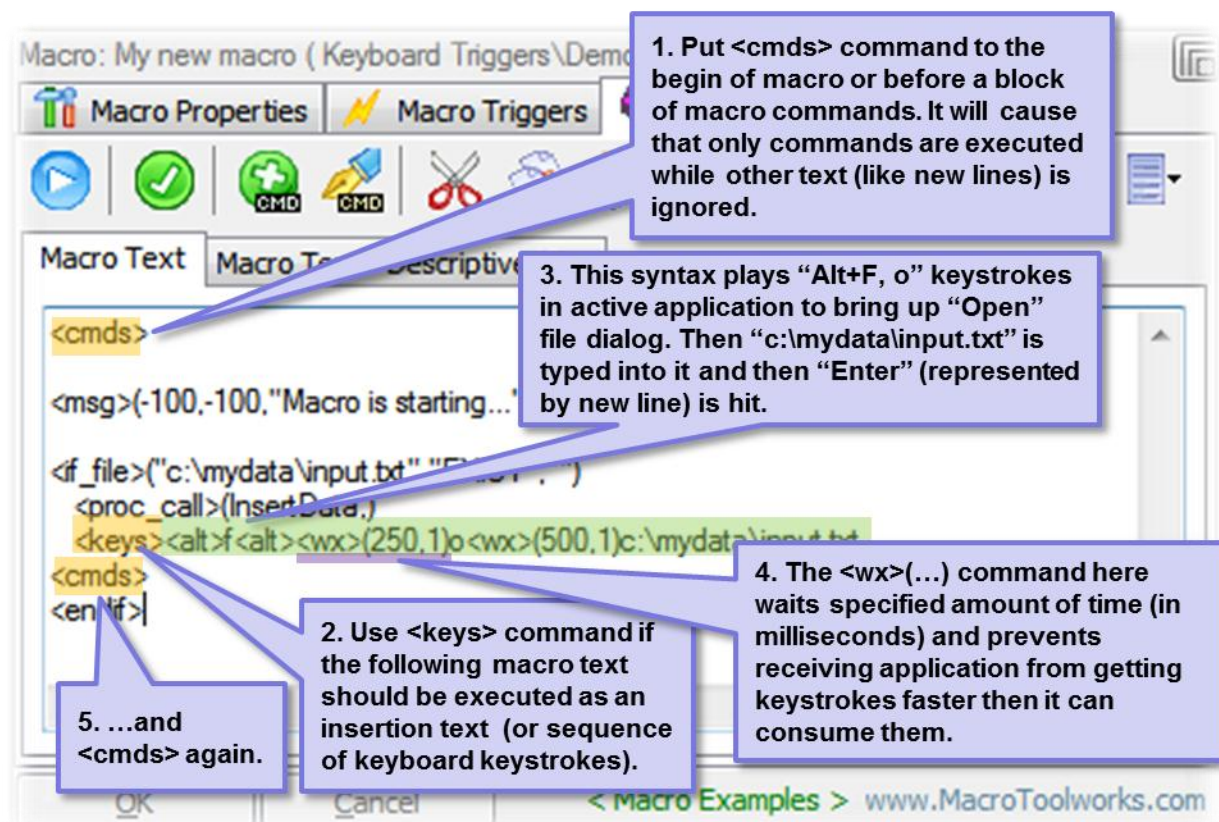


## 4.9 Configuring Macro: More Macro Text Details

If the macro combines both macro commands as well as insertion text then it is convenient to use `<cmds>` command and `<keys>` command to control how macro text is executed.

- `<cmds>` command tells that only macro commands should be executed while the rest of the macro text (that would be normally considered as an insertion text) is ignored.
- `<keys>` command stops the scope of the previous `<cmds>` command so that both insertion text and macro commands are executed.

The `<wx>(...)` command is good to use to make sure the application receiving keystrokes has enough time to handle them (so that it does not happen that keystrokes are sent too quickly and thus ignored by receiving application).

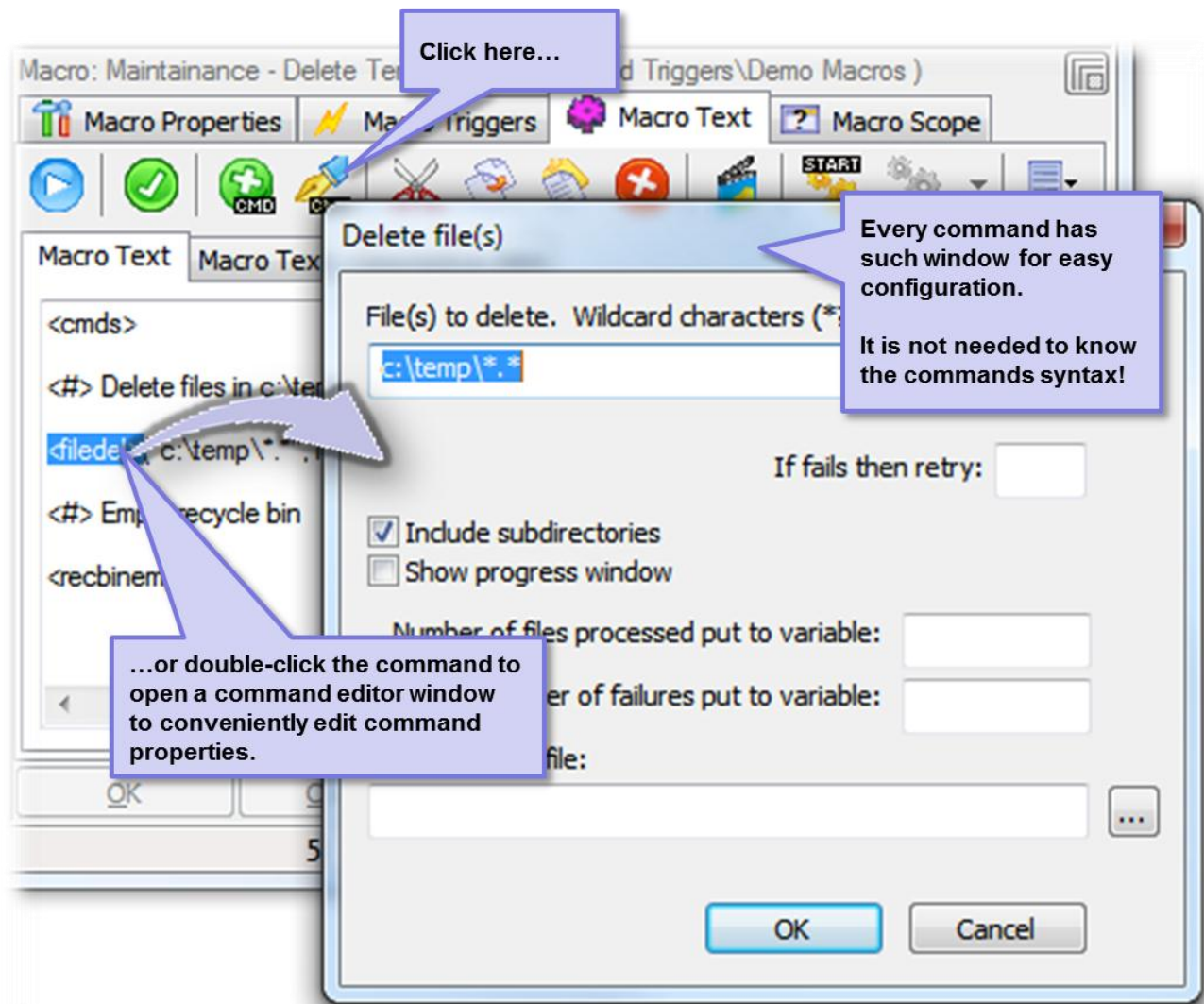


To mimic "Ctrl+A" key combination use `<ctrl>a<ctrl>`. The syntax for a key combined with control key (for example Alt+A, Ctrl+A, Shift+A) is as follows:

Ctrl+A	<code>&lt;ctrl&gt;a&lt;ctrl&gt;</code>	The first control key syntax means "control key down" the second one means "control key up".
Alt+A	<code>&lt;alt&gt;a&lt;alt&gt;</code>	
Shift+A	<code>&lt;shift&gt;a&lt;shift&gt;</code>	

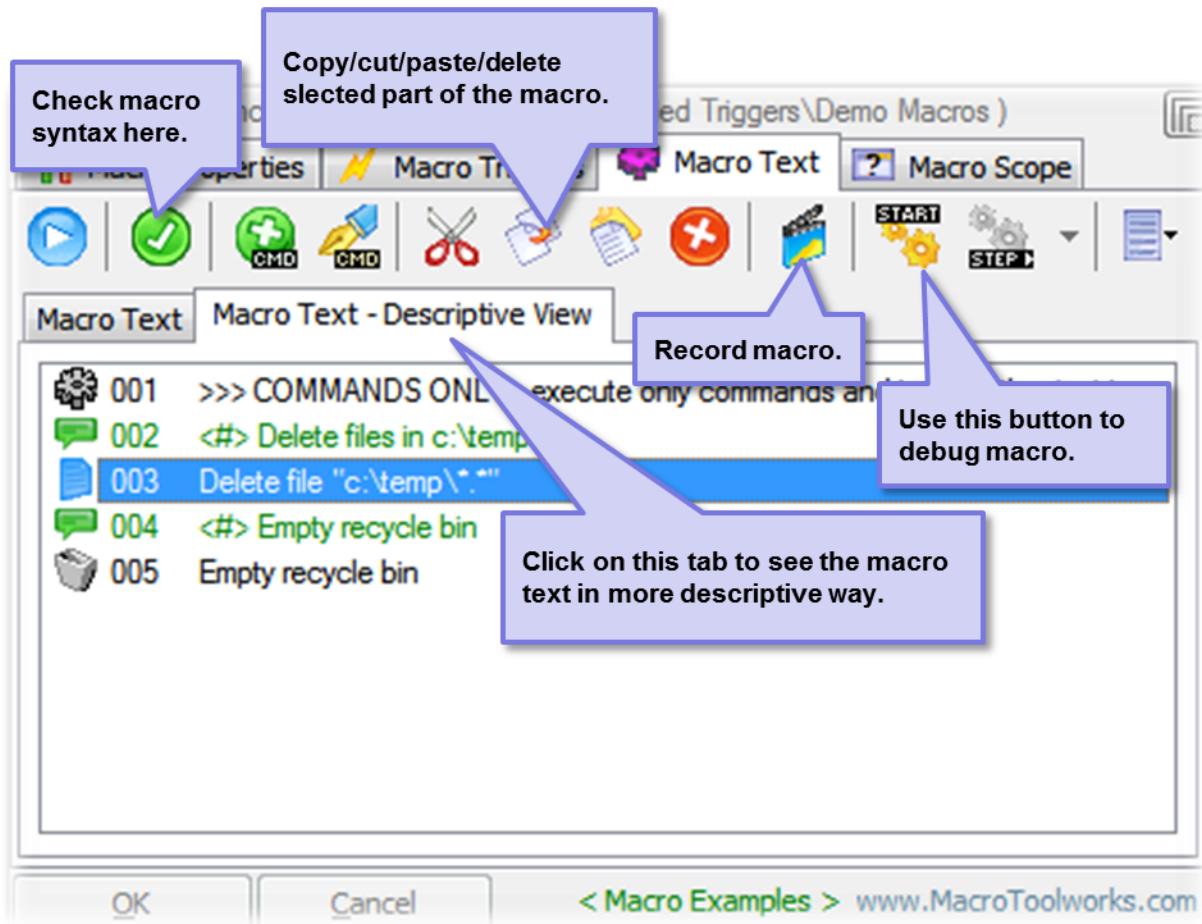
## 4.10 Configuring Macro: Macro Commands Editing

It is not needed you know the syntax of each macro command. The commands may seem complicated; however, if you just double-click on the command a convenient window where you can easily configure the command appears. There is such a window for each command.



## 4.11 Configuring Macro: More Macro Text Details

The macro text can be edited free form (as shown on previous examples) as well as it can be edited and viewed in other, so called, **Descriptive View** – notice the “Macro Text – Descriptive View” tab selected. In this mode the commands and insertion text is added using “Add command” button (the green button with white +)...

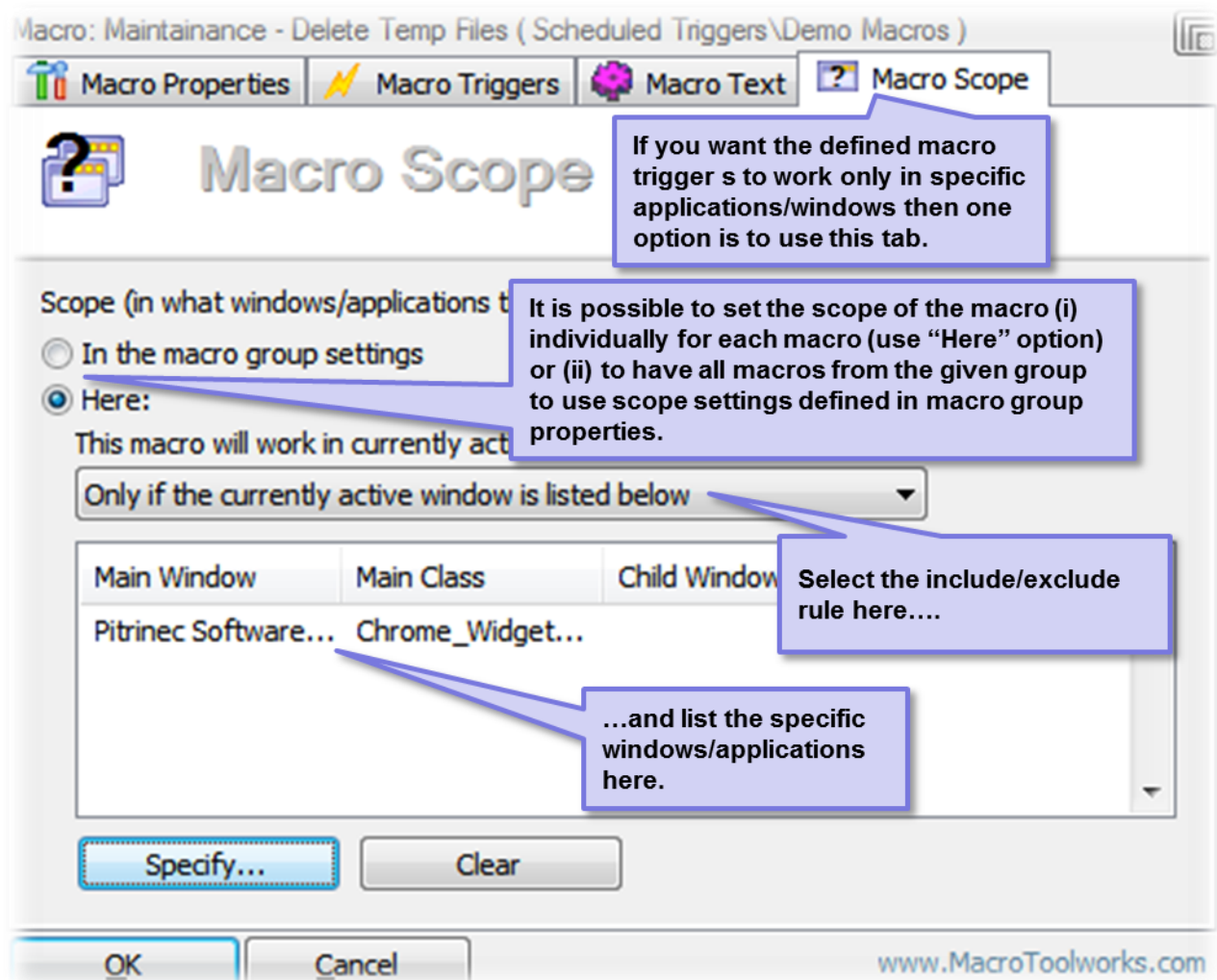


## 4.12 Configuring Macro: Macro Scope

In the simplest case the macro triggers (for example keyboard hot-keys) that you define in “Macro Triggers” tab work in all applications (window). So whatever application you press the hot-key the macro is started. Sometimes it is desired that triggers work only in some applications while not in other (or even start other macros in other applications). This so called “scope of the macro” can be defined in “Macro Scope” tab.

It is possible to specify the scope of the macro:

1. **Individually** for each macro as shown on the figure below.
2. In the **macro properties group**. The scope then applies to all macros from the given group. See more on “[Configuring Macro Group: Scope](#)”.



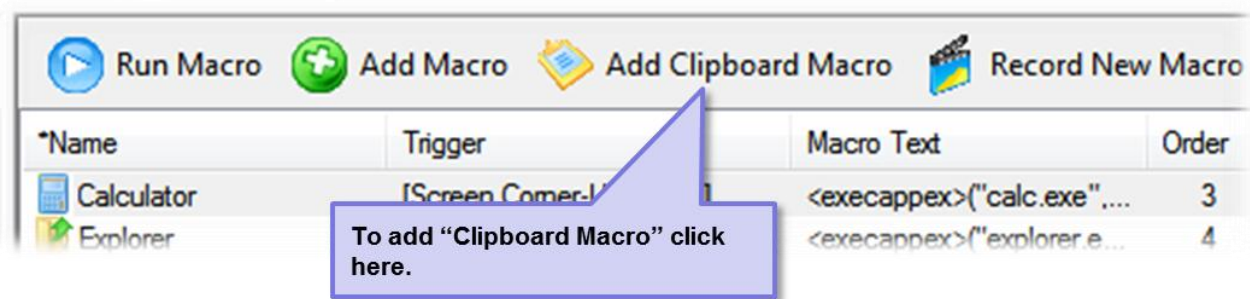


## 5 Adding Clipboard Macro

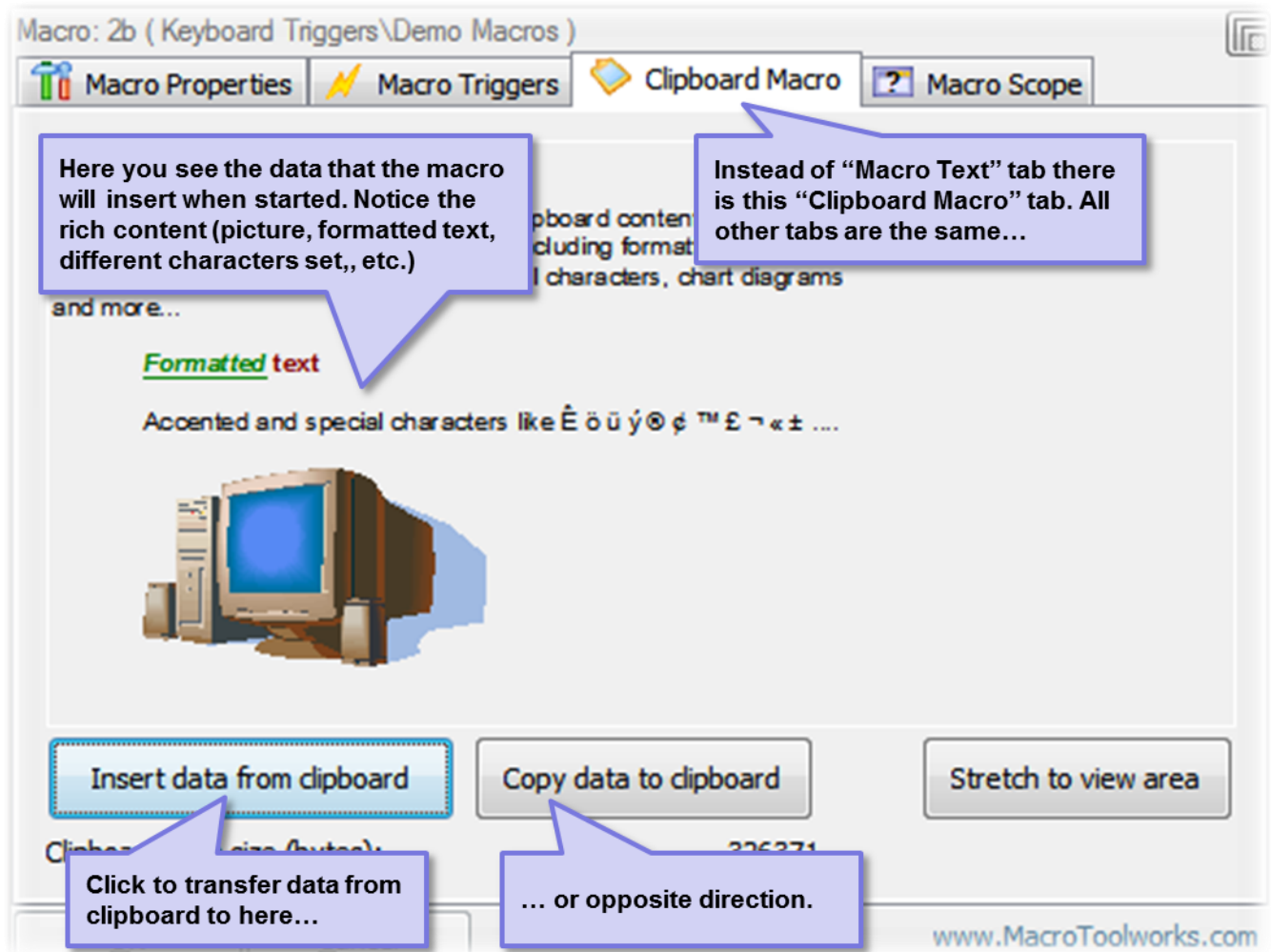
---

Clipboard Macro is a special kind of macro – it inserts data (text, rich text, tables, pictures,..., whatever data that fit to clipboard) to the application you are working with. This is very similar to “Simple Text Insertion” as described in “[What is Macro? – Simple Text Insertion Macros](#)” but it is capable to insert much more rich content (not only plain text). This type of macro is very usefull if you have set of pictures, tables, formatted text paragraphs, texts with web links, etc. that you often include to other documents or e-mails. Or this is good also for texts with foreign languages characters (for example Ыѳ ѡЉ Φ) that are not directly supported by Macro Toolworks products.

Clipboard macro can be added by clicking on “Add Clipboard Macro” button as shown on this figure:



When adding “Clipboard Macro”, everything that is described in “[Adding Macro](#)” chapter applies to Clipboard Macro as well. The only difference is that Clipboard Macro does not have “Macro Text” tab - it cannot contain commands. Look at the following figure to get a better idea how the “Clipboard Macro” looks like.

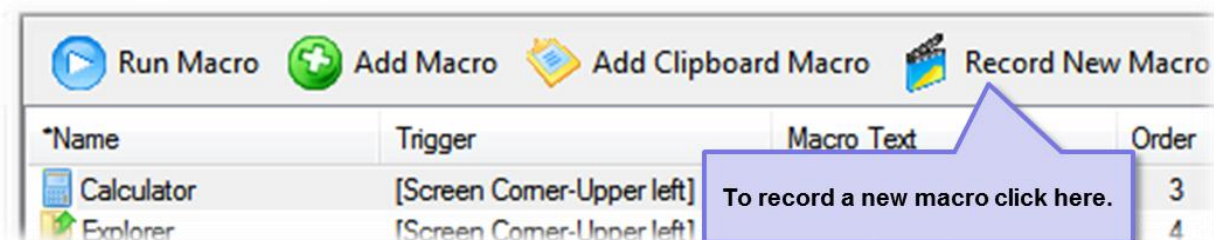


It is possible to run “Clipboard Macros” from other macro using **<run>** macro command. If you have a macro (some macro with commands as described in “What is Macro? – Combined Text Insertion and Commands”) and you also want to insert some rich content data as part of its functionality then you can create a separate “Clipboard Macro” and then run it from within the other macro using **<run>** command.

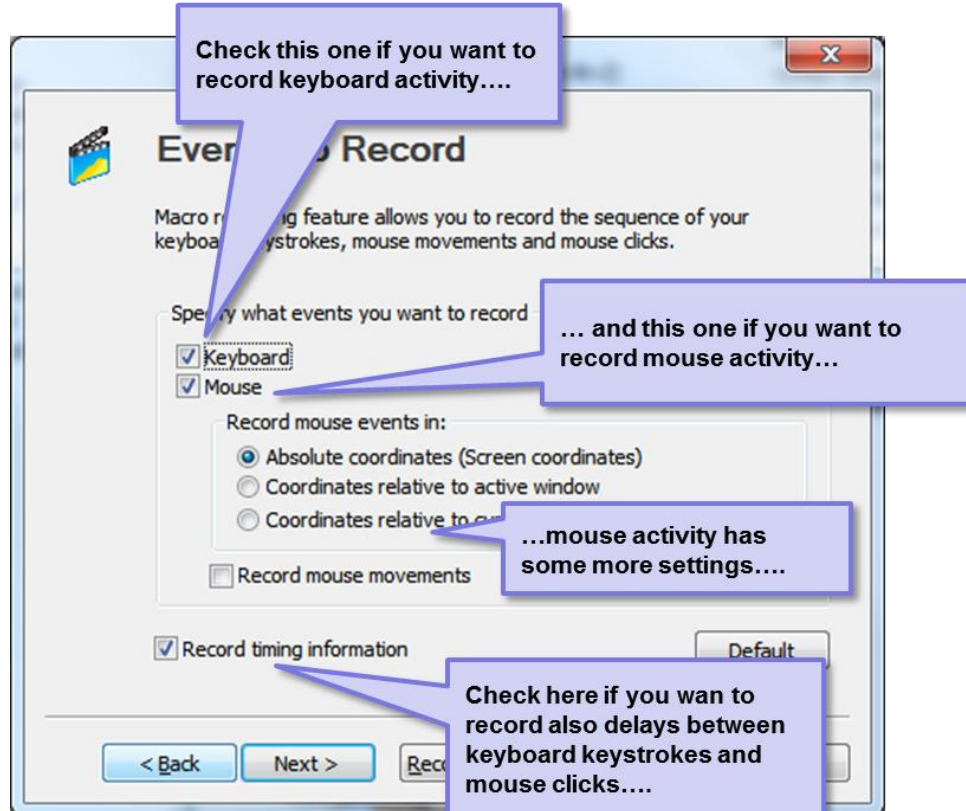
## 6 Recording Macro

One option how to create a new macro is record what user is doing with keyboard and mouse (typing on clipboard, clicking with the mouse, etc.). The recorded macro can then reply it back any time later.

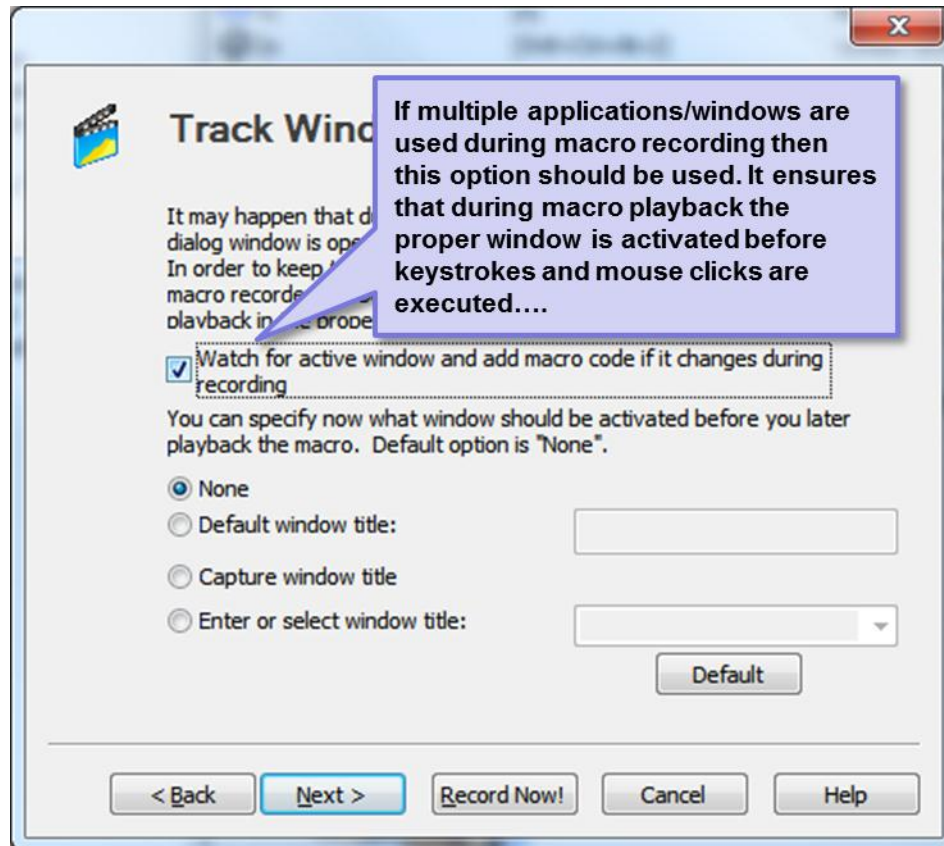
The recorded macro is not different from other macros as described in chapter “[Adding Macro](#)”. The only difference is that macro text itself (the content in “Macro Text” tab as described in “[Configuring Macro: Simple Text Insertion Macro](#)” and following chapters) is created automatically during the macro recording so that user does not have to create it manually. However, it is later possible to manually edit and tweak the macro if needed.



When “Record New Macro” button is clicked, a wizard window with recording options is opened. One important page allows defining what will be recorded:



Other important page is the one that turns on/off active window tracking:



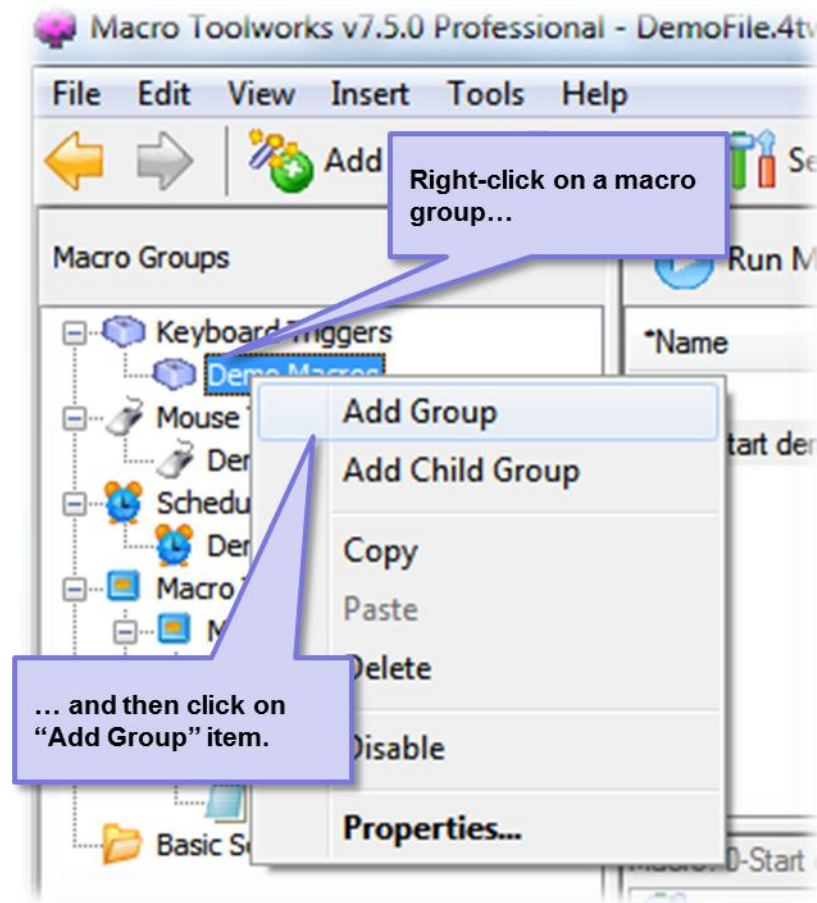
For example, let's imagine you are recording a macro where you type some text in a text editing program and then you switch to presentation program where you continue doing something. In this case it is good idea to use this "Watch for..." option. It will ensure that during the macro playback, when the part of the macro that worked within the text editing program is finished, the presentation program window is properly activated and keystrokes recorded in this program are also played back in this program.

*Note: It is always recommended to rather write macros by hand; especially it is highly recommended to follow rules that are described in the Help file in section "How to write reliable macros?"*

## 7 Adding Macro Group

---

There are several ways how to add a new macro group, one of them is to use mouse right-click on a macro group and from the menu shown select “Add Group”. If “Add Child Group” is selected then a group is added as a child node in the macro group tree hierarchie.



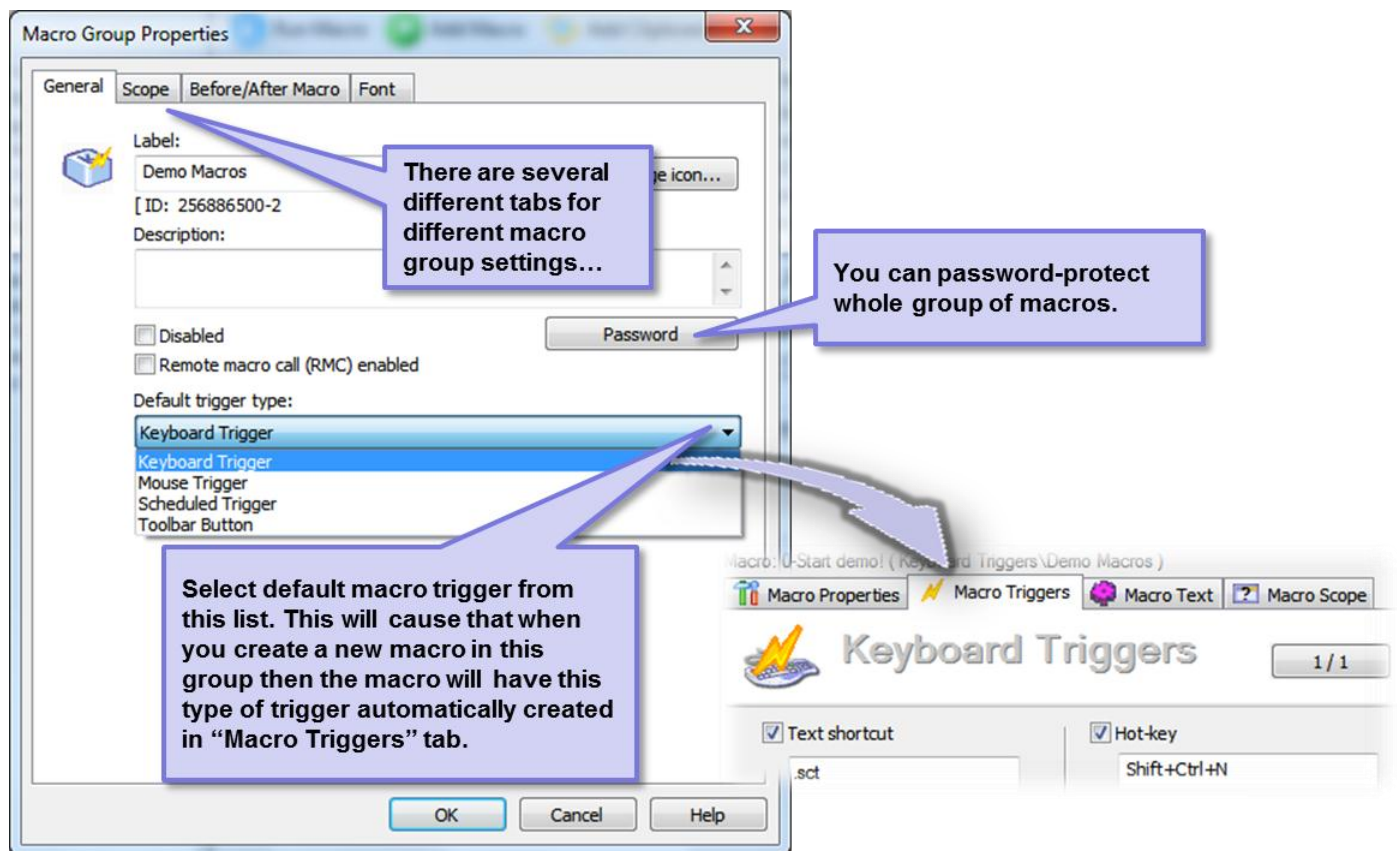
When a new macro group is added its properties window is shown. See on next pages what the configuration options are.



## 7.1 Configuring Macro Group: General Properties

The “General” tab allows you to define some basic properties like group name (label), icon, description, etc.

The “Password” button can be used to password-protect whole the group of macros so that without the password it is not possible to run macros from that group as well as to view them or to modify them (depends on how you configure password protection).

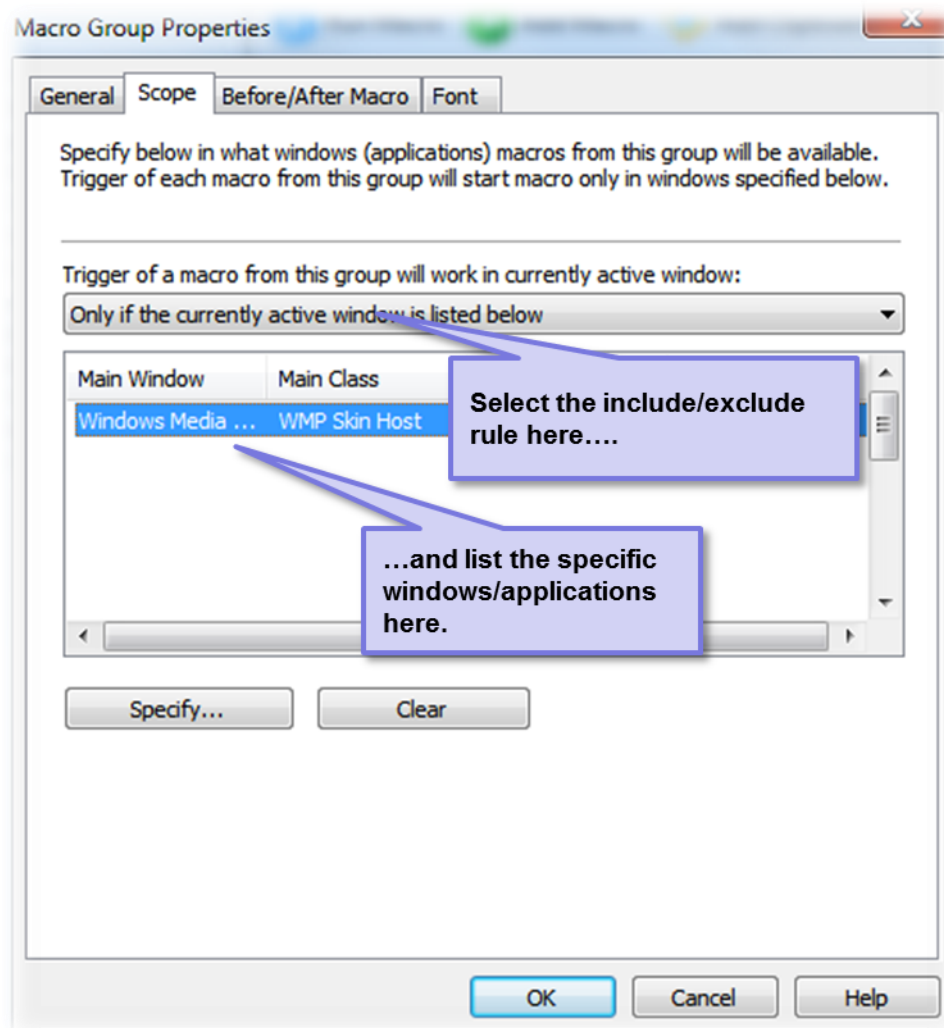


The “Default Trigger type” field is used when you are creating a new macro in the group. Based on this setting, given macro trigger type is automatically created in the “Macro Triggers” tab in newly created macro.

## 7.2 Configuring Macro Group: Scope

The “Scope” tab allows you to specify what are the windows (applications) where macro triggers of the macros from this group will work (or will not work).

This option is very useful if you have several macros that are intended to be used just in one particular application - for example in an on-line chatting program. If the macros have assigned hot-keys as macro triggers then these hot-keys only start macros in the on-line chatting program but have no effect in other programs (or even the same hot-keys can start different macros in different program!).

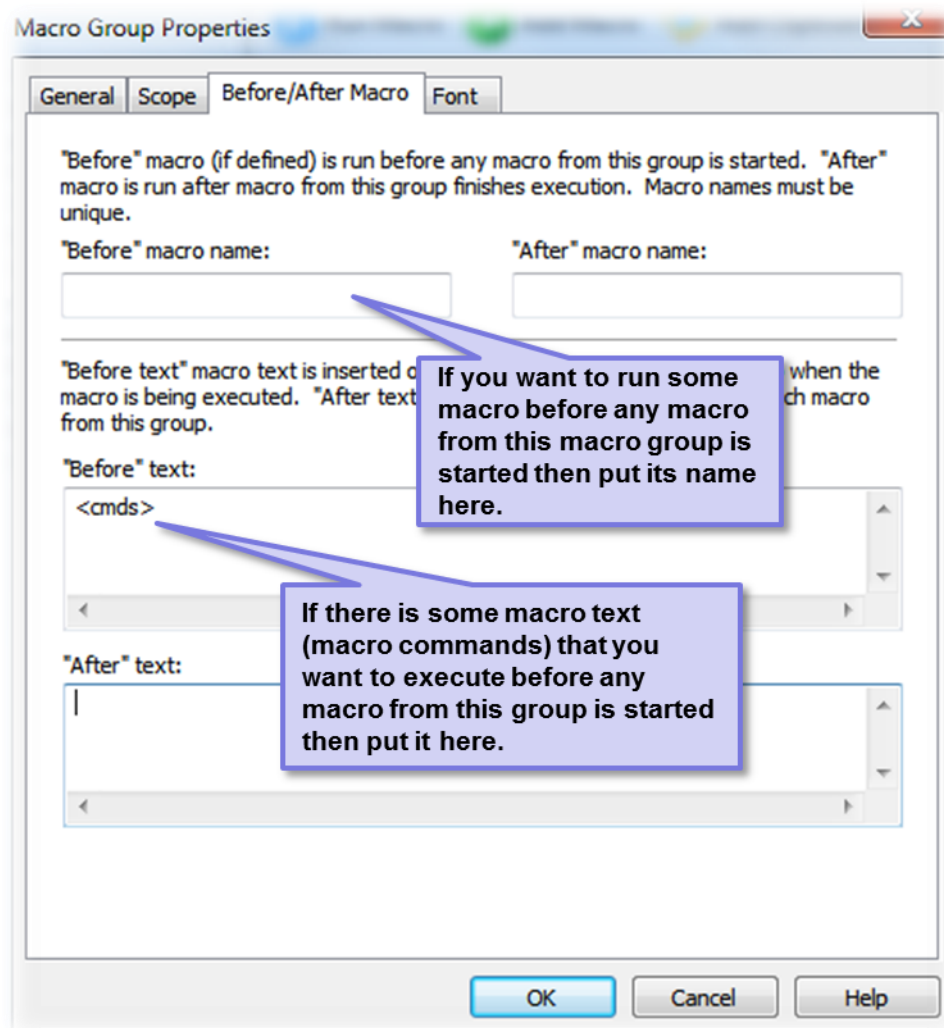


This option is related to “[Configuring Macro: Macro Scope](#)”.

## 7.3 Configuring Macro Group: Before/After Macro

The “Before/After Macro” tab allows you to run macro or some macro commands before any macro from this group is started or after any macro from this group is finished.

This functionality is handy for example if you want to check some conditions that are common to all macros from the group. For example if all macros from the group are doing operations on some particular file then it can be convenient, before any of such macros is started, to check if the file exists. So it is possible to either create a macro for this purpose and put its name to “Before macro name” field or just write the macro code to “Before text” field.

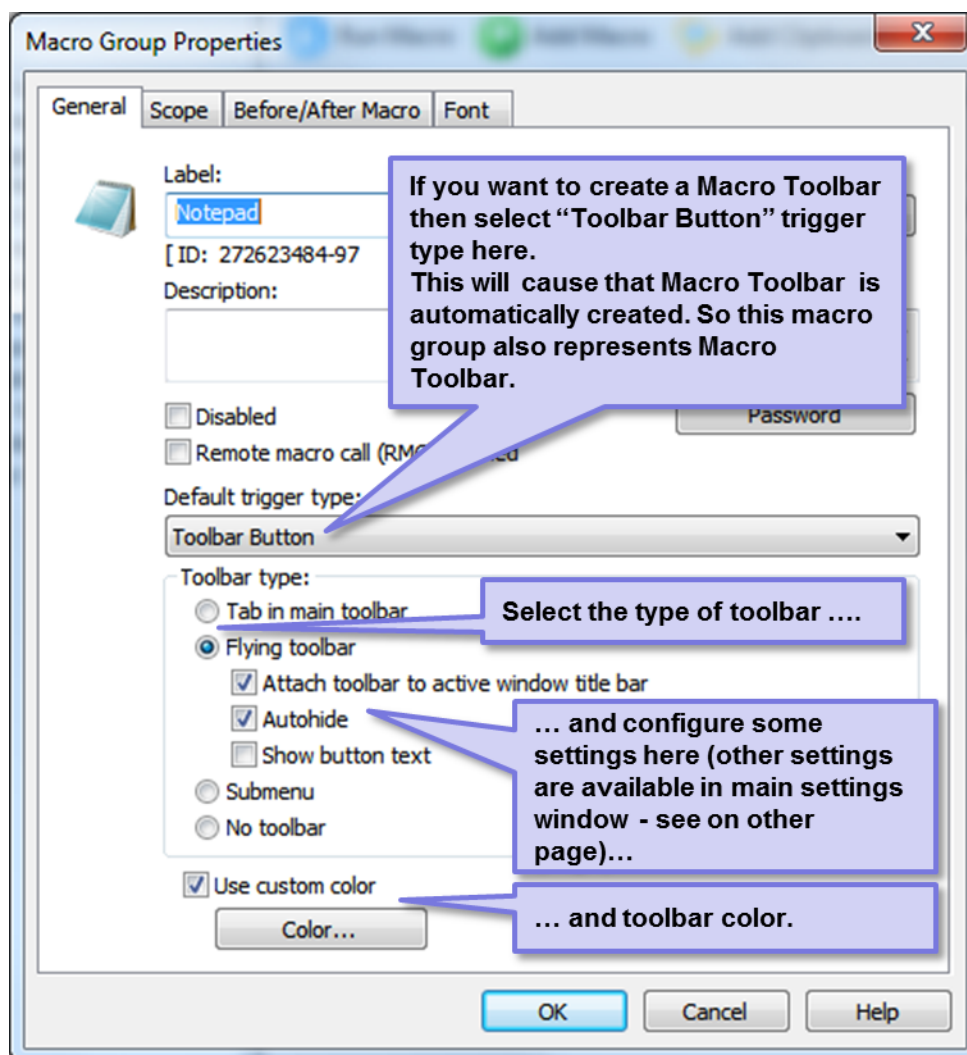





## 8 Adding Macro Toolbar

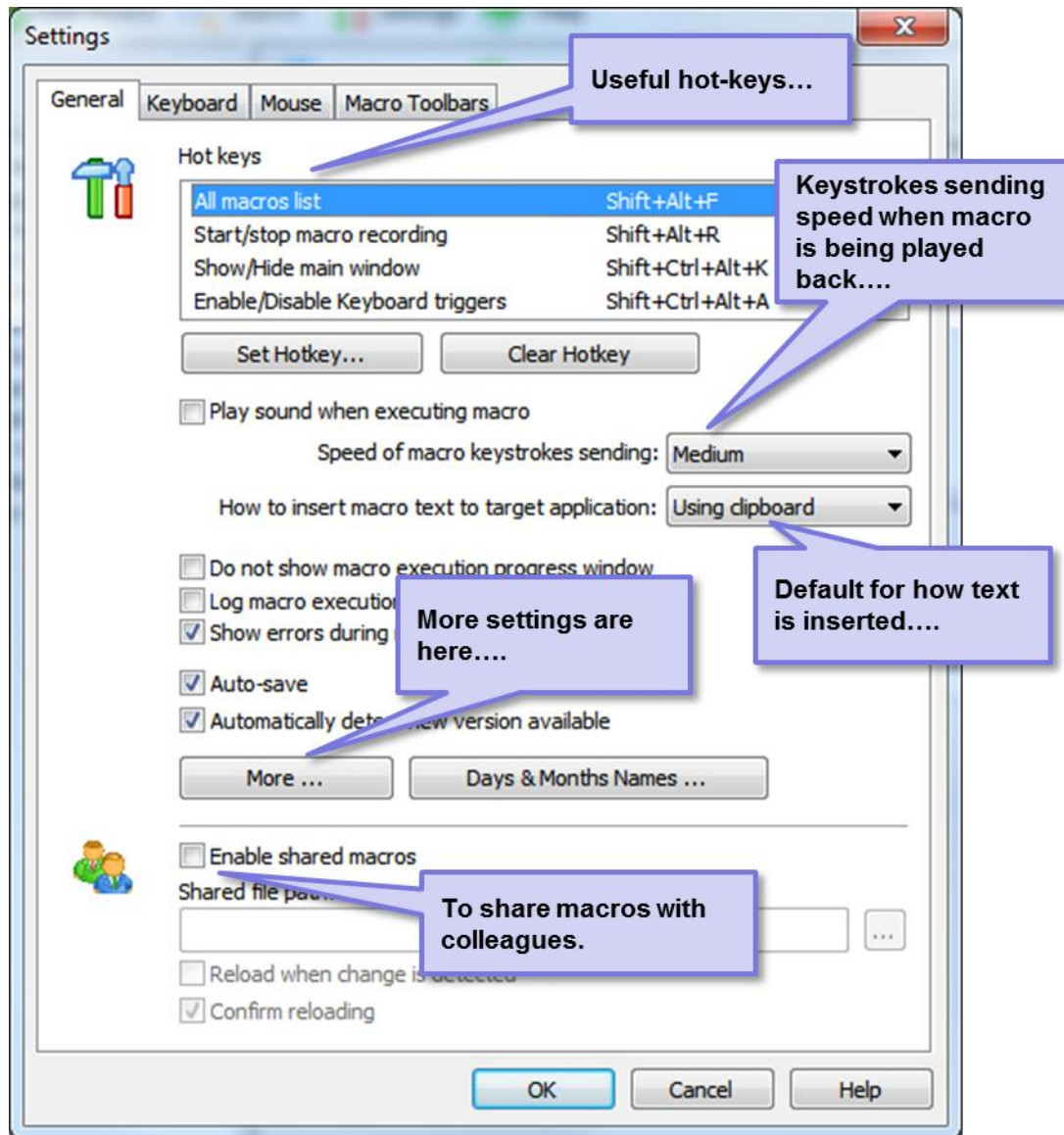
There is one main macro toolbar with multiple tabs and any number of flying toolbars and submenus.

Adding a macro toolbar – which means tab to main toolbar, flying toolbar or submenu - is very easy but not that obvious. To add a new macro toolbar you simply add a new macro group as described in “[Adding Macro Group](#)”. In the macro group properties window select “Toolbar Button” as a default triggers type. This creates a macro toolbar that is represented by this macro group. It means that options applied to this macro group apply to macro toolbar as well. For example, if you disable this macro group by “Disabled” option then the macro toolbar hides. Or if you use “Scope” tab then you can make the macro toolbar to be showing only when defined applications/windows are active (on top of other windows).



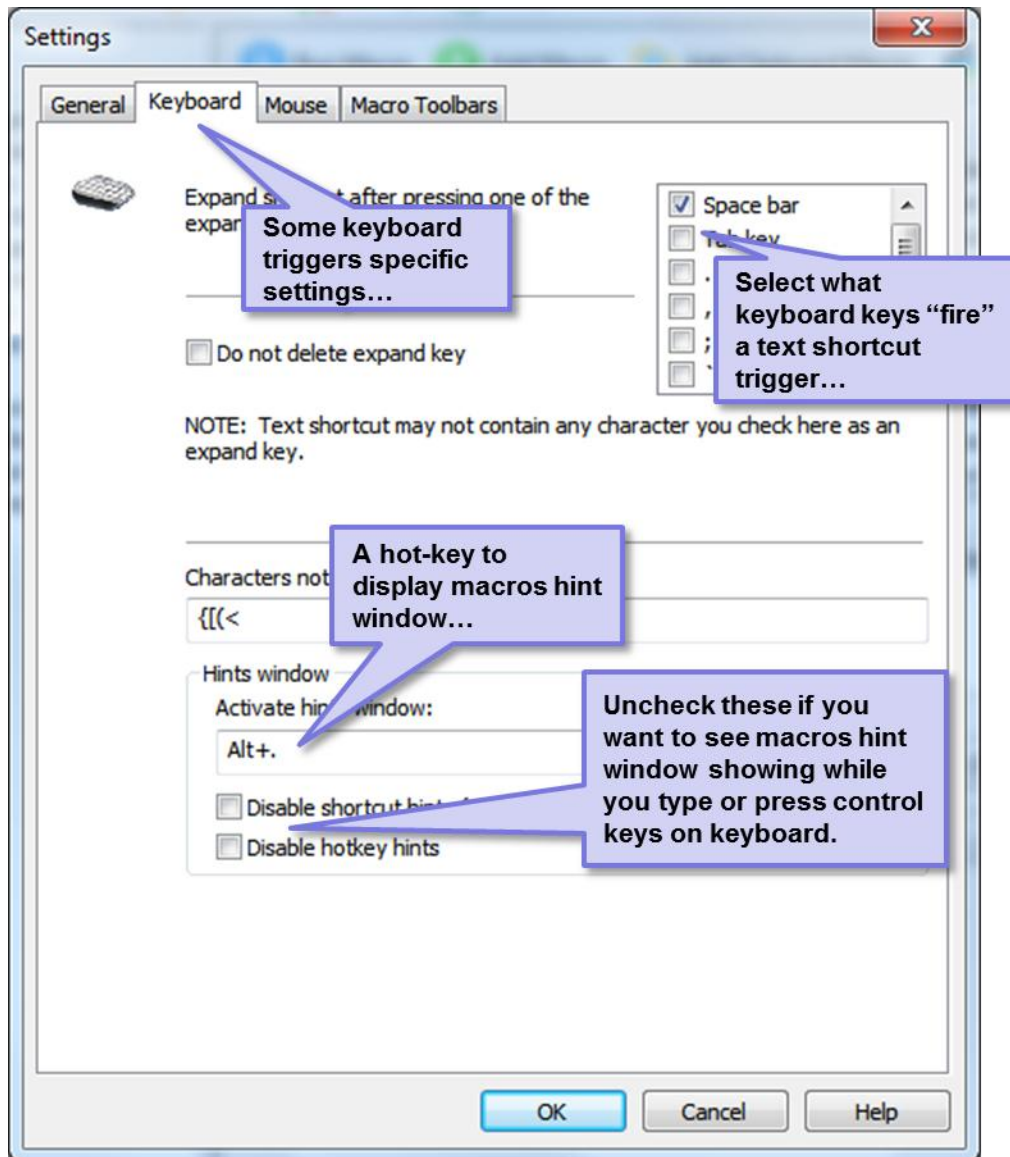
## 9 Program Settings

There are several program settings that are available by clicking on  **Settings** button or by selecting “Tools/Options” menu command (and also on Ctrl+P hot-key).



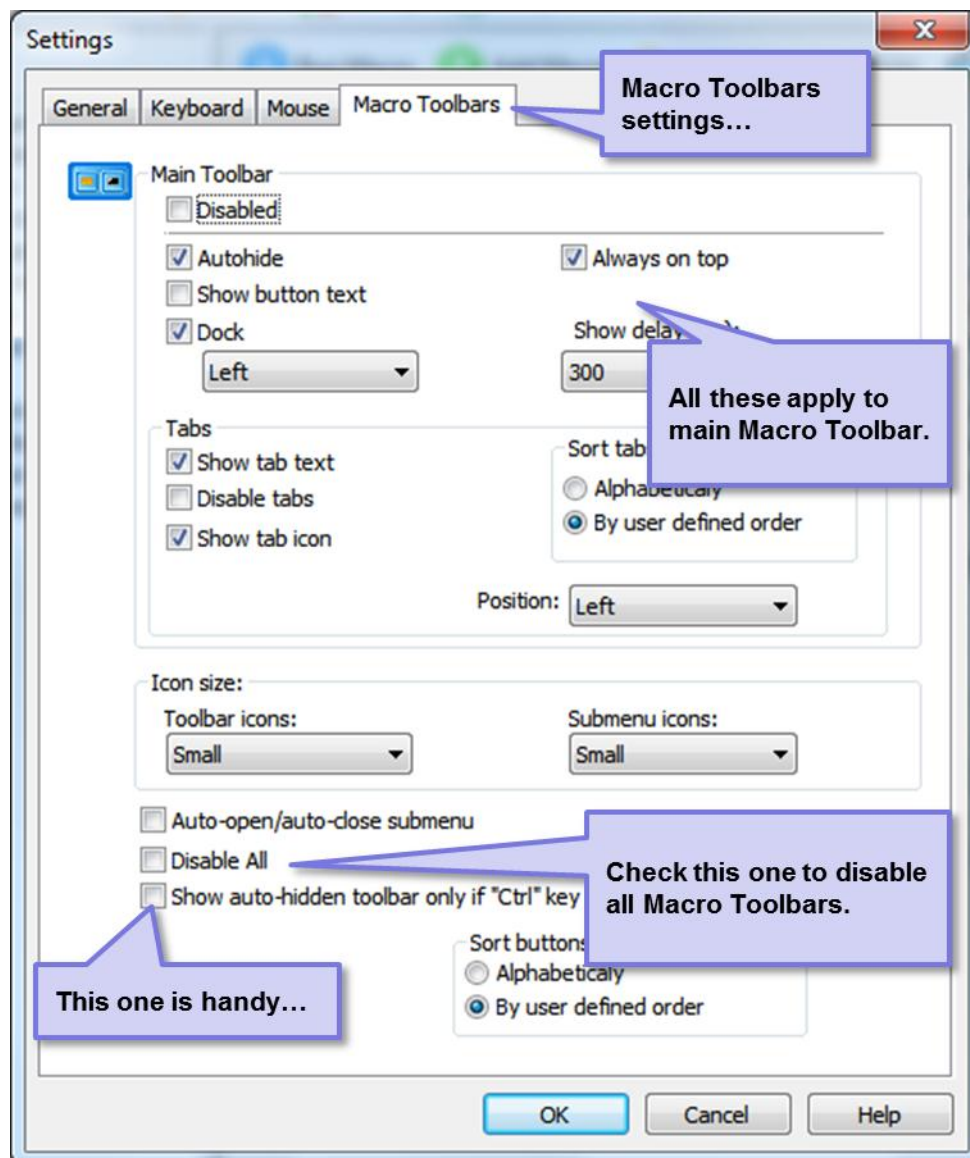
## 9.1 Program Settings: Keyboard

There are some keyboard triggers specific settings in “Keyboard” tab. These settings are only available to Macro Toolworks and Perfect Keyboard products.



## 9.2 Program Settings: Macro Toolbar

There are many settings that allow you to customize look and behavior of the macro toolbars. These settings are only available in Macro Toolworks and MacroToolbar products.



## 10 Other Information

---

The Macro Toolworks family product consists of several individual programs. They all share the same user interface and the same macro engine.

Product	Versions Available	Major Difference
<a href="#"><u>Macro Toolworks</u></a>	Free, Standard, Professional	N/A
<a href="#"><u>Perfect Keyboard</u></a>	Free, Standard, Professional	Only keyboard triggers available
<a href="#"><u>Clicky Mouse</u></a>	Free, Standard, Professional	Only mouse triggers available
<a href="#"><u>WinScheduler</u></a>	Free, Standard, Professional	Only scheduler triggers available
<a href="#"><u>Macro Toolbar</u></a>	Free, Standard, Professional	Only macro toolbars as triggers available
<a href="#"><u>Free Macro Player</u></a>	Free only	The same features as Macro Toolworks, the macro file cannot be modified/viewed, intended as a macros runtime