KeyText 3 User Manual

2015, MJMSoft Design Limited

KeyText 3

Text and Automation Utility for Windows

by MJMSoft Design Ltd

The following people have helped greatly with suggestions and comments, and thanks to them are hereby recorded: Terry Blount, Jeremy Gelber, Philippe Supera, Matt Johnson and Steve Hobberstad.

MJMSoft Design Limited is a private limited company registered in Scotland No. 183702, and having its registered office at 56 Kingsknowe Drive, Edinburgh, EH14 2JX, United Kingdom.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

July 2015 in Edinburgh, Scotland

Table of Contents

Part I

1 KeyText 3, First Things

1.1 Introducing KeyText

KeyText 3, for Windows XP, Vista, Windows 7, 8 or 10
By MJMSoft Design

KeyText - the multi-purpose text and automation utility.

This powerful program locates itself in your system tray: it is available all the time - but takes up very little space on your desktop.

KeyText's basic function is the storing of pieces of text, ready to be typed or pasted into whatever application you are working on. Autotext in your system tray! You can set up hotkeys to all your favorite text items - one keypress and the text is in. Your Word Processor probably has a function like this already, but what about your database program, your desktop publisher, or your e-mail program? The beauty of KeyText is that it's always there waiting in your system tray ("notification area"), and works for most applications.

New to version 3 of KeyText is **Trigger Text**. You can set KeyText to look out for certain sequences of keypresses - and launch a macro when they are typed. For example, every time you type #dt, KeyText could be set to type the date instead. /ad could type your address. Try setting e// to type é or #1/2 to type ½. Use it to type long or difficult-to-spell words - for example, set ptfe to type polytetrafluoroethylene. Or it could even correct your spelling - set calander as the trigger text for calendar.

With KeyText’s **right-click anywhere** feature, form-filling has never been so easy. Hold down Shift, Ctrl or both (depending on how you set it up), right-click where you want your text to go, and KeyText's menu opens up next to it ready for you to click the item you want. Your name, e-mail address, phone number, etc. - no more typing them in: let KeyText do it.

Added to this keyboard macro ability are a number of features which transform it into a multi-purpose utility. Eliminate those repetitive tasks!

SMART SELECT Unique **Smart Select** feature. You already know how to select text and press a key combination to copy it to the clipboard, turn it to bold, and so on. With KeyText's Smart Select you can do much more! Once set up you could, for example, select an e-mail address anywhere - document, text file, database - and press Ctrl-Shift E. Your e-mail program will start with a new message to that address ready! KeyText can also work out what you have selected and do different things depending on what it is. An email address? Start a new email. A zip or postcode? Go to a map. A phrase? Do a phrase search.

• Include in "text" **commands** to fill in the date, run programs, click buttons, change windows, select menu items, play a sound, display a message, and more. Easy wizard-based dialogs help you set it up. You can even specify an Internet address, and KeyText will take you there; hotkey it and you can launch your browser and go to the site on one keypress.

• **Schedule** items to start at specified times. Because the text can include commands (fields), you can run programs at specified times, display reminders or alarms with your choice of sound, and even have an hourly chime. Do you want KeyText to trigger an event on, say, the third last day of every month? That is possible too.

• **Windows automation**. Tell KeyText to look out for certain windows or dialogs, and then run an item. Completely automate password entries, repetitive dialogs, etc. You can also get KeyText to do different things depending on whether a specified window exists/appears or not. KeyText itself includes 6 password levels to protect any passwords, credit card numbers etc. you might store.

• KeyText can simulate **mouse** actions. It can click the right spot on a window - button on a dialog, link on a web page etc. - even if the window is in a different position each time.

• KeyText can operate as a **multiple clipboard**, and makes collecting text easy. You can set up KeyText to monitor your clipboard, and whenever text is copied to it, automatically add the same text to a specified text item. This makes it easy to gather text from various sources to put into one destination. You don't even need to open a text file to get at its text; select it in Windows Explorer and a couple of mouse clicks puts a copy of it in KeyText ready for reuse.

• KeyText can either paste or type your text. Pasting is faster, but "typing" it as a macro, character by character, has advantages. It can include commands, it doesn't require the clipboard, and you can choose the speed. For the benefit of online-chat users it includes "simulate manual typing" settings; add some pauses, mistakes, and backspaces and the person at the other end will never know it's being typed automatically!

• KeyText has a **random link** function. Set up your e-mail signature in KeyText, along with a random link to a range of quotations held in other items. Hotkey the signature if you like. Then, every time you press the hotkey, your e-mail signature will be typed in, followed by a quotation drawn at random.

• KeyText can display a miniature "keyboard" on your screen, which lets you "type" with your mouse or pointing device. This was added for the benefit of those who don't always have a keyboard attached, and asked for the facility to key text with KeyText. The **Keypad** also has a symbols tab; all those tricky symbols like ¼, ©, ™ etc. are only a click away - as are all the regular accented characters.

Up to 234 (or 20 in the Evaluation Version) of your favorite text items can be stored, and are then available to any application with two mouse-clicks or one hotkey press. With the full version you can have several "KeyTexts" running at the same time, giving the possibility of thousands of text items just a click or two away.

It's very unobtrusive... It hides itself away in the tray, near your system clock. A left-click on the icon reveals a menu showing all your text items ready to be typed. A right-click shows all the options available.

1.2 Installing KeyText

KeyText requires Windows XP, Vista, Windows 7, 8 or 10, and needs 2.2MB of space on your hard drive.

KeyText is distributed as a .zip or an .exe file. If you have the former, extract the files to a (temporary) folder, and run SetupKT.exe. Or, if you have the self-extracting .exe version, simply run ktext###.exe (where ### is the version number). During the install process you can choose to have KeyText added to your list of programs which start every time Windows starts; note that you can change this later in Global settings.

If you are upgrading from an earlier version
Simply install over the old version - do not uninstall first. When you run KeyText it will convert your existing data (*.ktt) into KeyText 3 format (*.kt3), so that all your items are available straight away. It does not delete your *.ktt files, and provides an application RestoreOldKt.exe in case you want to go back to the old version. If your old version is already registered, KeyText 3 will recognize your old code and unlock KeyText 3 for 45 days - by which time you should upgrade to a new unlock code (or restore the old version).

The files which are installed in the KeyText folder are as follows:
KeyText.exe: main program executable
KeyText.dll: KeyText system file
KeyText2.dll: KeyText system file 2
KeyText3.kt3: data file with sample items
Samples.kt3: same as above - to refer back to after changing KeyText3.kt3
KeyText.chm: help file
KeyText.dat: install configuration file
chime.wav: sound file, used by sample data file
license.txt: details of your license agreement
pad_file.xml & pad_file.htm: information for vendors/distributors
readme.txt: readme information
uninstall.exe: uninstall program; see below
whatsnew.doc: tells you what's new
RestoreOldKT.exe: installed if you had v2.25, to allow you to restore it

KeyText does not alter or add any Windows system files.

How to Uninstall KeyText
Go to Control Panel - Add/Remove Programs, and select KeyText; or select the Uninstall icon in the KeyText program group. Any KeyText data files (*.kt3) which you have made will need to be deleted manually - KeyText doesn't delete your data!

Part II

2 Using KeyText

2.1 Tutorial

This topic introduces KeyText's main features, and includes some tips to help you in using KeyText. We also suggest that you explore the Insert Field Wizard in the Edit text items window; it may help you think of other possible uses for KeyText.

The KeyText program uses macros or scripts known as items, which are named A, B, C and so on. They can be started from the KeyText menu, from hotkeys or abbreviations (trigger text), automatically or from batch files. Together these items are stored in data files with the extension .kt3.

It is important to note at the beginning that if you want to start a new item you select "Edit text items" on the menu which appears when you right-click the KT icon in the tray - and look along the tabs A, B, C etc. until you find an empty one (or delete an old one). If you want to start a new set or collection of items, you select "New" from the right-click menu - this opens up a new file of empty macros ready to use. But beware of using "New" just to start a new macro - go to Edit text items instead!

Use KeyText to store a favorite text
First of all, an item is the name given to a piece of text stored by KeyText in its data file. In the evaluation version there are 20 items available, labeled A to T; in the full version there are 234, labeled A to Z, 2A to 2Z, 3A to 3Z... 9A to 9Z.

Right-click the KeyText icon in the tray and select Edit text items. Along the top of the resulting window you will see a row of tabs with these labels; with the full version there is also a tab with > which, if clicked, shows the next set of texts.

When you open the window item A can be seen in the edit window underneath the tabs. Click various tabs to see what's in the items - the tabs which are shaded green are empty, and those shaded red have some content - probably the sample items that ship with KeyText.

As you run your mouse over the tabs you will see an indication of the contents of each item - choose one that is empty, or which you do not need. Or start a completely new (empty) set of items by right-clicking the KT icon and selecting New.

You can then edit the item to give your own text, and, if you like, set a hotkey to it by clicking the ⌨ button, then clicking the box to the right of it and pressing the key combination you require. If your keyboard has a Windows logo key it can be included in a hotkey combination (although note that a number of Windows logo key combinations are reserved for Windows use, and you may not be able to set them).

The box to the right of the hotkey box lets you set up Trigger text to start the item - a sequence of characters which KeyText will look out for when you are typing, and will start the item (after deleting the Trigger text characters first). This means you can set abbreviations to your favorite texts.

The right-hand box on the toolbar lets you specify a title for the text - which will appear in the KeyText left-click menu; if you leave this blank, KeyText will make a title using the contents of the text item.

You can also tell KeyText what to do if the item is selected. If it is a large piece of text, Paste is probably best - click the button on the toolbar. Choose the Macro button if you would like KeyText to type it in for you, simulating manual typing, or if your text is going to have special instructions such as button clicks or menu select fields. If the method of entry is not important, or if you would like to be able to change the action on a number of items easily, leave it at default - you can then change the action by selecting the appropriate setting on the right-click menu. See also Paste, macro, run... what's the difference? to help you decide.

***Important:** If your text includes one or more left braces - { - you must change it to 2 together: {{ otherwise KeyText will interpret it as the start of a field. This restriction does not apply to the right brace }. If you often use the brace characters, consider changing the field delimiters to [...] or <...> in the global settings dialog.*

See also How to get text in... for more details.

Use text stored in KeyText
First of all, make sure that the caret is at the place you want your text to go, and that the window is active - normally indicated by the color of the title bar.

Left-click the KeyText icon in the tray, and a menu will appear showing the text items you have stored (with the full version, click Next… on the menu for the next set of items). Click the item you want and it will be inserted at the caret of the active window. Alternatively, open up the menu with a left-click then type the underlined letter associated with the item to be inserted. Small icons on the menu indicate whether the text will be pasted or typed and the KeyText icon in the tray will either show a paste symbol, or mimic a tiny keypad "typing".

There is a shortcut to the text stored as item A - simply double-click the KeyText icon with the left mouse-button (unless you have chosen in Settings / Customize to have a double-click open the Keypad). You can store the phrase you use most often there, for ease of access. Or set hotkeys, trigger texts or desktop shortcuts to your text items. See how to get text out... for full details.

If KeyText is typing text for you, and you want to stop it before the end, either left-click the KeyText icon in the tray, or press Esc. To pause typing, right-click the KeyText icon, or press the Pause or f12 key. Once in pause mode, indicated by an animated icon in the tray, a right-click, Pause or f12 keypress will resume typing, or left-click or Esc will abort.

It is important that you do not use the keyboard or mouse while KeyText is typing.

Tip: When the right-click anywhere menu is on screen - or the menu which appears when you left-click the icon in the tray - you can select an item with your right mouse button, and have something different happen, as set in the customize settings dialog. For example, say you have set the right-click option to add a Return, and item A contains the first line of your address. Select A with your left mouse button, and the line will be typed - suitable for filling in a form. But in a document, try right-clicking it, and a Return is added at the end - ready to start the next line. Another option is to have the right-click take you straight to the Edit window for the item clicked.

Use abbreviations to enter accented characters or symbols
You can set up Trigger text for KeyText to look out for, and when you type it KeyText will start the appropriate macro.

> For example, if you want to type é using the sequence e//:
> Set up an item which only has é in it, and set it to Action:Macro (or Action:Paste, if desired).

> In the toolbar of the Edit text items window, locate the text box 2nd from the right - it will show the tooltip "Trigger text".
> Type into that: e//.
> Now, every time you type e//, KeyText will intercept it and type é instead.

A similar method can be used for symbols. For example, set #half to type the symbol ½.

Tip: Trigger text is case sensitive, so you could set up another item with É and give it the Trigger text E//.

Fill in a form with KeyText
You can do this in the same way as mentioned above; but KeyText also has a "**right-click anywhere**" feature which makes filling in forms - whether on Internet pages or elsewhere - easier than ever.

Using the Edit text items window, set up some texts to try this feature. For example, each line of your address in separate items, your e-mail address, or other frequently-entered information. For quickest entry click the button on the toolbar so that the text is pasted in. Don't worry about setting menu text for the items (right-hand box on the toolbar), because KeyText will simply display the contents of the item itself in the menu automatically.

When you installed KeyText, it asked you to choose the shift key(s) to press with a right-click to open the "right-click anywhere" menu. If you did not set this when installing, you can do so now by going to the global settings dialog in Settings. Choose, for example, "Ctrl", so that right-clicking a text box while you have Ctrl pressed down will open the KeyText menu.

Let's assume you have your e-mail address stored, and Ctrl set for the right-click anywhere menu. Now find a form which asks for your e-mail address - or any form to test it, or simply a word processor or text editor. Hold down Ctrl and right-click your mouse over the point you want the e-mail address to go. On the menu which opens up click, with either mouse-button, your e-mail address - and it will be pasted (or typed) in.

The "right-click anywhere" menu always shows your name at the top, then other KeyText items - except for any which are empty, or which have a schedule set. In the Options dialog you can set it to start at a different place from item A. This is particularly useful in the full version of KeyText; you could keep your form-filling texts further down the list of items, for example starting at 4A; set 4A as the first one to show, and they would appear top of the list in the new menu. Pressing Next… on the menu will take you to the next set of items.

When this "right-click anywhere" feature is enabled in the global settings dialog, you may need to suspend it temporarily. You may, for example, want to press Ctrl right-click in an application, and not have the KeyText menu appearing. You can turn it off temporarily by right-clicking the KeyText icon in the tray and selecting "Suspend right-click anywhere".

Include simple formatting in text
KeyText is designed to work with simple text. However, in macro mode you can specify key combinations which will toggle certain text formats on/off, if your word processor supports it.

> For example, if you want text like this:
> Formats such as **bold**, underline and *italic* can be included...
> Make a KeyText item like this (using Insert Field Wizard if you wish, to assist with the key combinations):
> Formats such as {Ctrl B}bold{Ctrl B}, {Ctrl U}underline{Ctrl U} and {Ctrl I}italic{Ctrl I} can be included...

This should work in most word processors; check your word processor (etc.) documentation for other key combinations that could be included in a KeyText text item.

Use string variables
If you have a piece of text which is frequently used as part of longer phrases, you could consider saving it as a global variable. This is especially useful if that text changes from time to time, and saves you having to change it everywhere it is referenced.

> For example, if you have a "product of the day" which changes every day, and is used in a number of different KeyText items, you could set it up as a string variable.
>
> Right-click the KeyText icon and select Organizer, then click the button "Manage variables". Click the "Strings" radio button, then click New to add a new variable, and call it ProductOfTheDay@. OK the New box, and give it a value; close the dialog and organizer.
>
> Now, every time KeyText encounters {ProductOfTheDay@} in an item, it will fill in the value you have given it. And you can go back to Manage Variables to change it any time. Or even set up an item to change it:
> {ProductOfTheDay@=Input "Product" "Enter new product of the day"}
> This would produce a dialog allowing you to enter a new value.
>
> Note that you can have local string variables too - which only operate within one item (and any items called or linked-to by it). Local variables do not have the @ at the end.

KeyText stores global string variables in a file normally called KeyText.gsv. Multiple users can share this file if it is on a network - and you can password protect it from changes. Say the Product of the day mentioned above is accessed by many users. One user could be nominated to have password control over changing the .gsv file (select "Prevent changes..." on the Manage variables dialog to set a password), and that person could update it every day without worrying about other users changing it.

Regular expressions allow manipulation of string variables. For example:
{ProductOfTheDay@ ".*" "\U&"} would type or display it in upper case.

Get user input
There are four possibilities here. The first one can be used with Action: Paste or Action: Macro; the rest only with Action: Macro. Remember that the fields in the examples below can be set up using the Insert Field Wizard.

1) If you have an item which includes repeated variable information, such as a customer name or product code, while the rest of the text is standard, the Ask field can assist. Ask fields are processed before the item is run or pasted; you specify the title for the dialog and the question to appear on it, and when it is run the dialog offers a text box for you to fill in the information. On the edit toolbar, press to use the Insert Field Wizard to set it up, choosing the ? button and clicking Next for the Ask wizard.

You will be asked for a name for the Ask field, and the question to ask. The name will appear on the title bar of the Ask dialog when the KeyText item is run - and note that after the ask field you can use {name} to repeat the same information.

> An example will illustrate this. If you give name as First name and the question as Enter customer's first name, the wizard will generate a field like this: {Ask "First name" "Enter customer's first name"}. When the item is run, when it is being pre-processed a dialog will appear asking the question, and the result is filled in replacing the Ask field.
>
> Once set up with the help of the wizard, the following in an item set to Action: Paste -

Your order for: {Ask "Product" "Enter product ordered"}
Dear {Ask "First name" "Enter customer's first name"}
I am pleased to inform you that your order for {Product} has now been completed. You should receive…

would, when started, produce 2 dialogs asking the questions, and then paste the result which might be something like:
Your order for: The Latest Gadget
Dear John
I am pleased to inform you that your order for The Latest Gadget has now been completed. You should receive…

Tip: When setting up the Ask field using the Insert field wizard, you can specify that pressing Return in the Ask dialog should "OK" the dialog (rather than start a new line). You can also specify that the result of the Ask is not entered in place of the Ask field - only when the name of the Ask field is used.

2) If you have an item set to Action: Macro, you can use the Insert field wizard to make an Input field. Instead of being asked for the information just before the item begins, a dialog with the title and question you specified appears at that point in the item. The information you enter into the dialog is then typed and the KeyText item continues. This method is best if there is some variable data in an item, or if you have a loop and want to enter manually one piece of data on each repetition.

The result of an Input field can be typed immediately - or assigned to a variable and used later. The name of the variable is used in the same way as the name of the Ask field above.

On the edit toolbar, press to use the Insert Field Wizard to set it up, choosing the button and clicking Next for the "Display a message or ask for user input" wizard.

For example, say you have a text editor open with a list of 10 names, and want to automate adding a description to the end of each line.
{Loop 10}
{Shift End}{Ctrl C}{Description=Input "Enter description" "Description for {Clipboard}"}
{End} - {Description}{Right}{Endloop}

This will select each line and copy it to the clipboard, then display a dialog asking for the description - and using a nested field to show the name in the dialog.
It will then move the cursor to the end of the line, type the description, move to the next line and repeat.

3) If you want KeyText to type a certain amount of text or perform certain actions, and then pause for you to perform some action before continuing - for example, typing in a filename which varies each time - you can use the {Pause 0} or simply {Pause} field. When KeyText reaches there it pauses, you can type in the filename etc., and then either press the Pause or f12 key, or right-click the KeyText icon in the tray, to make it continue. If you want to abort the pause, just press Esc or left-click the icon.

4) 2 above showed the use of string variables. If you use integer variables you can use a field like {#c=Input "Input dialog title" "Enter value for c"}; this will display a dialog with the given title and text and, if you input a number in the dialog it will be assigned to the specified variable. A subsequent field like {#c} would output that variable's value.

For example, the following fields in an item:
{#s=Input "Pause" "Enter number of seconds to pause"}{Pause {#s}}
will result in a pause for the number of seconds you type in the dialog box which appears when the item runs.

Transpose letters in any application

How often do you tpye - sorry, type - a word and find you have mixed up 2 letters? This example shows how you can set a hotkey to swap them over, and should work in most applications.

In this example we will type fields directly into a KeyText item - but note that in most cases the Insert Field Wizard can be used to help set it up.

Right-click the KeyText icon and select Edit text items. Locate an empty item and make a KeyText item like this (using Insert Field Wizard if you wish, to assist with the key combinations):
{Shift Left}{Ctrl X}{Left}{Ctrl V}{Right}
Make sure there is no [Return] typed at the end; it should finish after the closing }. Choose the Macro button to ensure that KeyText types the characters one by one, rather than pastes them. Before pressing OK, click the button, then click the box to the right of it and press a hotkey, for example, [Ctrl Shift S]. Press OK to confirm.

Now place the caret after two characters which need to be transposed and press the hotkey you selected. They should swap places. In fact, what happens is: the {Shift Left} moves the caret one character left while selecting it; the {Ctrl X} is the equivalent of Cut - cuts it to the clipboard; the {Left} moves the caret to before the other character; the {Ctrl V} pastes the deleted character in its new position; and the {Right} puts the caret back to where it was before. If KeyText is set with a fast typing speed set in Options, this will all happen very quickly! Note that the clipboard is used, and afterwards will contain the moved letter.

How to vary text depending on user input

You can use Ask or Input fields in conjunction with If fields and string variables to type alternative pieces of text depending on user input.

For example, say you have a piece of text which should be different depending on whether the person you are writing about is male or female. You can use the Insert field wizard to set this up, or type it in directly. Note that the {IgnoreBlanks} simply helps to lay out the first section more clearly; returns, tabs and spaces are all ignored.

```
{IgnoreBlanks}
{sex=Input "Sex" "Enter sex (m/f)"}
{If {sex} = "m" then}
        {heshe="he"}{hisher="his"}{himher="him"}
{Elseif {sex} = "f" then}
        {heshe="she"}{hisher="her"}{himher="her"}
{Else}
        {Message "Error" "{sex} is not a valid input"}
        {Cancel}
{Endif}
{IgnoreBlanks Off}
When the bell rang, {heshe} returned to {hisher} room. Nobody saw {himher} again that day.
```

The example sets up 3 string variables set to he/she etc as appropriate. To take this example a little further, the next sentence could begin:

{heshe ".*" "\u&"} had fallen into a deep sleep.

This would type He or She as appropriate; the regular expression is used to change the first letter to upper case. The ".*" matches the whole of the string variable, and in the replacement expression "\u&" the \u converts the next character to upper case - and the & includes all the matched text.

Simulate mouse clicks

You can use KeyText's Mouse field to simulate mouse actions such as click, double-click or drag and drop. With the Edit text item window open, move it to a part of the screen where it does not obscure the point you want KeyText to click. Then use the Insert Field Wizard to set it up, choosing the 🖱 button and clicking Next. Again, if the point to click is covered, move the wizard away, making sure that the ⌖ target can still be seen.

Move the mouse over the target ⌖ hold down the left mouse button, drag to where you want the mouse action to happen, and release. You will see the co-ordinates change as you move the mouse on the screen. For convenience you can check the box "Hide wizard while dragging target", which will give you visibility of the whole screen.

You can select whether to work with screen co-ordinates (top left is 0,0) or co-ordinates relative to the window under the target when it is released.

This "relative to window" option gives KeyText the unique ability to mouse-click a button even if it appears on different positions on the screen. As long as the same window (the one over which you released the target) is at the front, and the point to click is consistently placed on the window, KeyText will find it!

> For example, you might want KeyText to type some standard text in your e-mail program, then click the Send button (note that you could also do this by getting KeyText to simulate a menu selection, or a button click where you give the text on the button). Type your text into a KeyText item, click 🪄 to open the wizard, 🖱 and Next. Now in the mouse wizard, select "Left click" and "Relative to window" and drag the target to the button - release it there. Press Finish and KeyText will insert into your text something like {Mouse L,#100,80}. Don't worry about the code; it simply tells KeyText that you want a left-click, and gives relative co-ordinates.
>
> Before you finish you might want to make sure that all this is going to the right window, so you could move the caret to the beginning of the text and use the wizard again to make an Activate field to bring the e-mail window to the front. There are also all sorts of possibilities for automating starting the e-mail program, but you can explore these later.
>
> OK the KeyText edit window; it will probably tell you that your text has fields in it which require Action: Macro to work. In other words, you can't paste mouse clicks! KeyText will change it for you, then OK it again and your new item is ready for use. Left-click the KeyText item and select the item you just made. It will automatically ensure your e-mail window is at the front, type the text and make the mouse click to send the e-mail. And it should hit the right spot on the button, even if your e-mail program window is not in the same position on the screen - although, of course, the button itself must still be on screen.

So relative co-ordinates are useful if the location on screen of the target window is unpredictable; if the point to click is consistent in relation to the target window, then relative co-ordinates should hit the right spot - even if it is, for example, a link on a browser page. On the other hand, if a certain dialog always appears center screen, then screen co-ordinates should be used.

As well as left, right or middle button clicks and left double-clicks, KeyText can also automate drag and drop with the left or right mouse button. If you select such an action in the Mouse wizard a second target appears for you to select the spot where the mouse button should be released.

Maximize or minimize a window using KeyText

In the case of maximizing or minimizing another window, you can do this by mouse click, or through the window's system menu, accessed by pressing {Alt Space}.

See above for how to set up a mouse click action in KeyText. But to do it the other way…

> To bring Notepad to the front and maximize it, select Edit text items. and choose an empty item; click the button to set the action to macro. Then enter:
> {Activate "Notepad"}{Alt Space}x
> Note - do not type anything after the x; if for example you type [Enter] after the x, after KeyText maximizes the Notepad window, it would then type [Enter].

Run a number of programs
Right-click the KeyText icon in the tray and select Edit text items. Choose a suitable empty item and type the name of each program followed by [Return]. The full path is normally required, though not always necessary for programs such as Notepad. You can use the Insert Field Wizard to help you do this; it also helps you to set up any command line or default directory that might be required.

Click the run button for the item or, if the Insert Field Wizard has been used to add all the programs to run you could use the button.

When the item is selected it is treated as a batch file, and each program is run in turn.

> For example, a text item with:
> notepad
> calc
> ...with the run action selected, will normally run Notepad, then Calculator.

Note that if you want to include a command-line parameter or specify a default directory for the program, you must set it up with the help of the Insert field wizard run a program field.

Tip 1: If you give the name of a document etc., the associated program will be run and the document opened.

Tip 2: If you want to start, for example, a new text file, use the Insert Field Wizard, choose the run *button and click Next; then select "Start a new file with the extension...".*

Tip 3: If you want KeyText to check first if the program is running - and if it is just to bring it to the front - see the Insert Field Wizard Run program section.

Tip 4: If using the macro action, note that after the Run field you can add fields to select menu items, click buttons etc., or simply give text which will be "typed" into the program. You may need to add a pause or a wait if KeyText appears to start typing etc. too soon.

See also Paste, macro, run... what's the difference? and Run a program etc.

How to stop a hotkey or trigger text working in certain applications
You can set up a hotkey to an item - a key combination which, when pressed, launches that item. But what if you want to pass through the hotkey for certain applications?

> Say you want Ctrl-Shift D to start an item, except in Microsoft Word where you want its normal double-underline function.
>
> Set up the item with Ctrl-Shift D as the hotkey - by pressing that combination when your cursor is in the

toolbar hotkey window. Then, at the beginning of the item, use the Insert field wizard to insert an "If exe" field. Select in the wizard: Macro control, and click Next; then select "If clipboard text / window title..." and click Next again. Choose to check Exe (process) name, and specify that it contains the text msword - and select if "yes" and "continue". On clicking Finish you will get something like:
{If exe contains "msword" then}

Now add further text to that the start of the item looks like this:
{If exe contains "msword" then}{Ctrl-Shift D}{Cancel}{Endif}rest of item...

If Ctrl-Shift D is pressed when Word is at the front, KeyText will type Ctrl-Shift D again - this time passing it through to Word, and stop. For any other application it will continue with rest of item.

A similar approach can be taken to inhibiting trigger text.

Say you have the key sequence #half set to type the ½ character - but don't want this to happen in Word. You can set this up using the Insert field wizard as above, and what you want to end up with is an item like this (with trigger text set to #half):
{If exe contains "msword" cancel}½

Use KeyText to go to an Internet site

There are two ways of doing this - assuming you have access to the Internet.

Method 1: Choose a suitable item in the Edit text items window, and type the URL, for example:
http://www.keytext.com

Click the run button on the toolbar, and select a hotkey, if you wish, by clicking the button, then clicking the box to the right of it and pressing the key combination you require.

Method 2: Choose a suitable item in the Edit text items window, click ; click the run button on the wizard and select Next; click the bottom button on the resulting dialog and choose Next again. Add the URL to the http:// already there and select Finish. Click the button to set the action to macro, and select a hotkey if required.

Once either of these methods has been used, if you select the KeyText item, or press a hotkey associated with it, your browser will be launched (if not already running) and the Internet address looked up.

The second method has the additional advantage that you can continue with other actions - for example, filling in an online form or clicking a button. Use the pause field to get KeyText to pause a given number of seconds before continuing or, better, the wait field which tells KeyText to wait for the specified window title to appear - which could be all (or part) of the title of the required web page. See use KeyText to collect web e-mail for an example of this.

If you are using Microsoft Internet Explorer version 6 (or earlier) you could get KeyText to detect "Done" in the status bar before continuing - although note that with some web pages which include multiple items "Done" may flash up and disappear even though the whole page has not yet loaded.

For example {Wait 30.0 -"Done"} will cause KeyText to wait for up to 30 seconds, and continue when "Done" is found in the status bar; if it is not found, the item will abort.

Note that this method will not work with Mozilla Firefox or Internet Explorer 7, which use a non-standard Status bar.

Click buttons etc. in a web browser

A KeyText text item can include fields to click buttons - for example, {Click "OK"} will click an OK button on a dialog. But there may be times when a button is not detectable or clickable by this method, possibly because it is Java generated, or has its text "painted" on, so KeyText cannot find it; or you want to click a hyperlink to take you to another web page.

If the Click field does not find the button you want, you could use KeyText mouse clicks, or try the following.

> A little experimentation may be necessary. With the www page visible, press Ctrl-Tab to move the focus to the Address box at the top. Then count the number of Tabs you need to move the focus to the button or link you want. In KeyText what you want to have is a text item, set to Macro action, with the following - assuming your button/link requires 3 tab characters to get to it:
> {Ctrl Tab}{Tab 3}{Space}
> Finish the text item immediately after {Space} - do not press Return. If you are using Mozilla Firefox, change the {Ctrl Tab} to {Alt D}.
> The effect of an item like this on a www page would be to move the focus to the Address, tab to the desired button or link, and press Space to select it.

A similar approach can be used for going to a text entry field on a web page; note that if the field has any preset text in it, put an {End} field in KeyText so that it types at the right place.

Check for text in a file - without opening it

KeyText's Selection field - and "If Selection" - can work on selected text; but they can also work on a selected text file in Windows Explorer.

> The following example asks for the search text you want, and responds with a message box stating whether it was found in the current selection - either text selected in the foreground window, or a text file selected in Windows Explorer.

```
{IgnoreBlanks}
{SearchText=Input "Search text" "Enter text to search for"}
{If selection like "{SearchText}" then}
        {Message "Found" "{SearchText} found!"}
{Else}
        {Message "Not found" "{SearchText} not found"}
{Endif}
```

> When run, the item above will ask for your search text, then examine the selected text or text file (using a "regular expression") and report if it is found or not. Note that the tabs and returns used to lay out the fields above are not required, but give a more understandable layout; the {IgnoreBlanks} means that they are ignored.

Set a hotkey to the above example and you have a convenient method for checking if specified text is in a text file - without first having to open it in an editor.

Check for text in a web browser

Note that KeyText cannot directly detect text within a browser window - but you may be able, by using the Wait field, to automate detection as follows. This example applies to Microsoft Internet Explorer, and the fields detailed - while looking complex - can be easily constructed using KeyText’s Insert Field Wizard.

Say you have your browser open at a news page, and every five minutes you want it to refresh and alert you if the word "economy" appears.

> With the help of the Insert Field Wizard, create a KeyText item with the following script:
> {Activate "News page"}{Ctrl f5}{Pause 40}{Alt e}f{Pause 0.5}economy{Alt f}{Wait 5.0 !"Microsoft Internet

Explorer" if not found link B}{Return}{Esc}

Also have macro B with:
{Esc}{Wave "c:\alert.wav"}

Then, for the first item above, click the Edit schedule button on the toolbar of the Edit text items window, click "Start text item X..." and select Minutes. Select 5 minute intervals, and the "First start by user" radio button.
With the browser window open at News Page (which should also appear in the title bar) the above item will, every five minutes, bring the browser window to the front and press Ctrl f5 to refresh it. 40 seconds then gives time for the page to refresh, then {Alt e}f{Pause 0.5} opens the Find dialog and gives it half a second to display. economy is typed, and the Alt f presses Find next.
If the text is not found, Microsoft Internet Explorer produces a dialog which says "Finished searching the document"; it has the title: "Microsoft Internet Explorer". The KeyText item waits up to 5 seconds for that exact title to appear (the ! indicates an exact match is required) - and if it does, the item continues and the {Return}{Esc} clears the two dialogs. The item will run again in 5 minutes.
If the waited for dialog does not appear, this indicates that the text is found, and the item will therefore link to item B (or other specified item), press Esc to clear the Find dialog, and play your wave file alert.wav. It will also check again in 5 minutes.

That is one approach to this problem, which you may be able to adapt to other uses.

Use KeyText to collect web e-mail

This example assumes you are using Microsoft Internet Explorer to collect e-mail from Hotmail, that you have a permanent internet connection or have automatic connect set for dial-up networking, that you have not asked your browser or Hotmail to remember the password, and that you have asked your browser not to offer to AutoComplete passwords. If these are not the case, the item below will need to be adapted.

First of all, here is the script; make sure you have Action: Macro selected in the toolbar for the item you put it in - and remember that this can all be set up with the help of the Insert Field Wizard:
{Run "http://www.hotmail.com"}{Wait 90.0 "Please sign in"}
Name{Tab}Password{Return}{Wait 20.0 "Hotmail"}
{Message "E-mail check!" "KeyText has opened your e-mail page; remember to disconnect from the Internet after checking your messages."}

Before finishing the item, you could set some friendly menu text for it in the top right box of the Edit text items window, or set a hotkey for it - or a schedule (see below).

Taking it step by step, the Run field will launch the Hotmail site, starting your browser if it is not already running. At the time of writing this, the web page which opens includes " Please sign in " in the title - if this has changed, change the macro accordingly. KeyText will wait until that appears, and if it has not appeared within 90 seconds will abort the item.

It then fills in Name and Password - change these to your own. Finally, when your mailbox appears, KeyText produces a message telling you that you are now connected and the web-site is open - check your browser and read your e-mails, then disconnect.

If your browser does ask if you want it to remember this password - and if you don't - you could add the following:
{Wait 10.0 "AutoComplete"}{Click "No"}
immediately after {Return} in the macro above.

If you want this script to run at set intervals, say every 30 minutes, set a Schedule for the item by clicking the button and selecting "Start text item x: Minutes". Choose 30 minutes, and OK the schedule.

Within a few seconds KeyText will run the script, and will time the next run for 30 minutes after the item is complete - that is, when you press the OK button on the "E-mail check!" message. Or, if the item failed, say because your computer could not connect to the Internet, 30 minutes after it failed.

Now that you have a personal password stored in KeyText, you might want to password protect your KeyText data file.

Use KeyText to start a new e-mail message

There are three ways of doing this - assuming you have access to the Internet and your system recognizes the "mailto" prefix.

Method 1: SMART SELECT Use KeyText's Smart Select feature! Set up a KeyText field with:
{Run "mailto:{Selection}"}
and have it set to Action: Macro, and with a hotkey of, say, Ctrl-Shift E, then if you select (highlight) an e-mail address in any application and press Ctrl-Shift E, your e-mail client will start with a new e-mail ready to that address. See Smart Select for more information

Method 2: Choose a suitable item in the Edit text items window, and type "mailto:" plus the e-mail address, for example:
mailto:someone@somewhere.com
Click the run button on the toolbar, and select a hotkey, if you wish, by clicking the button, then clicking the box to the right of it and pressing the key combination you require.

Method 3: Choose a suitable item in the Edit text items window, click ; click the run button on the wizard and select Next; click the bottom button on the resulting dialog and choose Next again. Add the e-mail address to the mailto: already there and select Finish. Click the button to set the action to macro, and select a hotkey if required.

Once either of these methods has been used, if you select the KeyText item, or press a hotkey associated with it, your e-mail program will be launched (if not already running) with a new message ready.

Automate Start menu functions

If you want to automate a function on the Start menu, such as shutting down Windows, make a KeyText item (set to Macro) which begins with {Win} which presses the Windows logo key on your keyboard - even if you don't have one! Work out what sequence of keypresses - underlined letters, cursor keys and [Return] - can be used to run the function, and enter it in the KeyText item.

For example, the following script, in English version of Windows NT, will automate shutting down the computer:
{Win}us{Click "OK"}
The u is the underline letter in "Shut down...", the s makes sure that the "Shut down the computer?" option is selected, and "OK" is then clicked.

Automate a password dialog

This example is taken from the Authentication dialog in Microsoft® Internet Explorer - the password request some sites set up to allow entry to some or all of their pages, and assumes you have not asked the browser to automatically enter passwords. Even if you have, the principles described here can be used to automate many other repetitive windows events. Note that there are 2 alternative step 2's below.

Step 1 - make a careful note of what you do each time you go through this dialog; try doing it without the mouse - just the keyboard - and work out how to fill it in with the help of the [Tab] and [Return] keys. Then

choose a KeyText item and enter it there in exactly the same way. For example, if your Username is John and your password jb1759, start with an empty text item and type John, press TAB, type jb1759 and press [Return]. Or, if you like, use the Wizard to set up the TAB key etc., in which case you might end up with a text item like:
John{Tab}jb1759{Click "OK"}

If you find you have to use the mouse, use KeyText's mouse wizard to set it up - as long as the mouse position to be clicked is consistent either in relation to the screen, or to the application window the dialog belongs to.

Press the button on the toolbar to select "Macro" as the action for this text item, then press OK to store the new text in KeyText. You can try it out so far; the next time the dialog appears, left-click the KeyText icon in the tray and select the item you set up. The password dialog should fill in automatically. If not, double check that you have filled in all the details correctly.

Step 2 - automate the password. This requires you to set a schedule for the text item you have just set up. Go back to the Edit text items window for the item, and click the button - or right-click the KeyText icon, select Organizer, click the item required and click the "Add schedule" button.
Click the button against "Start text item ? whenever this window title appears", and give the title "Authentication"; select "Match whole of title". Leave the button "Wait for a new window..." selected.
Almost there, but you may have several dialogs which give the title "Authentication" - you have to tell KeyText which specific one to fill in. Look for some unique static text on the dialog (maximum 20 characters); it may follow the word "Resource:". For example, against Resource: it might say "Computer Bookstore Customer". In the KeyText set schedule dialog, click the Advanced button and type Bookstore (or sufficient words to identify it correctly), then OK that dialog, OK the set schedule dialog, etc.
From then on - assuming KeyText is running with that .kt3 file open, the dialog will be filled in automatically every time it appears. If you go to the Edit window for the item you have set up, you will see a green surround on the Schedule button, to remind you a schedule is set for it.

Alternative Step 2. Have KeyText run the URL in the first place, and use the Wait field before proceeding. The KeyText item might then look like this:
{Run "http://www.secureplace.com"}{Wait 30 "Authentication"}{Pause 1.0}John{Tab}jb1759{Click "OK"}
If necessary, you can specify additional static/fixed text in the wizard, so that KeyText can better identify the correct window. For example, if a waited for window has the same title as its parent window, additional identifying text may help to distinguish it.
KeyText will then launch the site and wait for a window with "Authentication" in the title (if not found within 30 seconds, abort). Then a 1 second pause - just to make sure the found window has a chance to display fully, then the password information.

Tip 1: Once you have a password stored in KeyText, you may wish to set a password for KeyText itself - for the current .kt3 file. Right-click the KeyText icon in the tray, select Settings and choose the Password tab. Select a level of password protection as appropriate.

Tip 2: Your Password request may be in the form of a "normal" window, rather than a dialog - for example, a page at an Internet site which requires certain input before letting you move to the next page. Experiment with the Ctrl-Tab, Tab, End and Return keys to see how the information can be filled in with the keyboard, and follow the above procedure, specifying the page title as the title for KeyText to look for. You may need to include a Pause in your text, to allow the whole page to display before KeyText begins typing. In the "If successful" section of the Set Schedule dialog, make sure you specify a time to wait. If you specify "wait for a new window with the same title", KeyText will only respond to that Internet page once - if you go back to the page, it is still the old window, even if the title has changed in between. Also note that you should be careful about specifying text in the Advanced schedule settings; the text in the browser page is not static, and no other application - including KeyText - can gain access *to it. Only fixed static text should be specified.*

Tip 3: If you want to start KeyText with the scheduler suspended hold down the SHIFT key while KeyText starts. Or if you want to temporarily suspend scheduled items, right-click the KeyText icon and select "Suspend scheduled items".

For further information, see "On Window" Schedule, Wait, Advanced "On Window" Settings and Password.

Obfuscate an email address

If you contribute to online blogs, forums or newsgroups, you may want to obfuscate your email address to help protect it from the "email harvesters" used by spammers. One common way of doing this is to change, for example, support@mjmsoft.com to support at mjmsoft.com. If you have to do this many times, you could set up a KeyText item to do it automatically.

This example takes an email address on the clipboard, and types it - replacing @ with at (with a SPACE character before and after):

```
{Clipboard "\(.*\)@\(.*\)" "\1 at \2"}
```

For fastest processing, set the item to Action:Paste, so that the result is pasted; and you may want to set a hotkey to it so that the procedure becomes as easy as copying and pasting.

The example uses regular expressions to work on the text on the clipboard. The \(and \) surround tagged groups which, in the replacement expression, are output using \1, \2 etc. The first tagged group, \(.*\) matches any number of characters, up to the @ sign which follows it. The second tagged group matches all the characters after the @ sign. The replacement expression, \1 at \2 therefore outputs the two tagged groups, with " at " in between.

Use Regular Expressions to rename files

Regular expressions allow you to manipulate text from the clipboard, current selection, etc. before "typing" it using KeyText. The example below is the most complex in this tutorial, so we'll go through it line by line.

This example helps you to rename a series of jpg files in Windows Explorer. It assumes you have a series of files such as PA235012.jpg; if, for example, you enter the prefix pool_ it would rename the file pool_012.jpg

```
{IgnoreBlanks}
{Ask ^-"Prefix" "Enter prefix for pictures"}
{Loop}
        {F2}
        {Ctrl C}
        {If clipboard contains "{Prefix}" goto end}
        {Clipboard "^[[:alnum:]]*\([[:digit:]]\{3\}\).\([Jj][Pp][Gg]\)$" "{Prefix}\1.\2"}
        {Return}
        {Down}
{Endloop}
{:end}
{Return}
```

{IgnoreBlanks} means that all the Return, Tab and Space characters in the item are ignored (except for the deliberate {Return}). This helps to lay out the script better.

{Ask ^-"Prefix" "Enter prefix for pictures"} produces an input dialog with title Prefix and text " Enter prefix for pictures". The response is stored as string variable Prefix. The ^ indicates that pressing Return when the dialog is open will "OK" it, and the - stops it outputting the response at that point. {Prefix = ^"Prefix" "Enter prefix for pictures"} would have the same effect.

{Loop} starts a loop which will keep going indefinitely, or until Esc is pressed. But later we have a way of detecting the end, which will break out of the loop.

{F2}. The idea is that the script will be started when the first file to be renamed is highlighted. The F2 keypress will therefore put it into rename mode.
{Ctrl C} copies the selected filename to the clipboard.
{If clipboard contains "{Prefix}" goto end}. When we reach the bottom of the list in Explorer, KeyText will try to rename that file again. So we check to see if it has already been renamed (contains what we entered for Prefix), and goto the label {:end} if it has.
{Clipboard "^[[:alnum:]]*\([[:digit:]]\{3\}\).\([Jj][Pp][Gg]\)$" "{Prefix}\1.\2"} is where the renaming takes place using Regular and Replacement expressions - and a nested field - {Prefix}. The regular expression is made up as follows:

^ anchors the search pattern at the beginning of the clipboard contents
[[:alnum:]]* matches any number (from zero) of letters or digits
\([[:digit:]]\{3\}\) sets up a tagged group of 3 digits - which will replace \1 in the replacement string
\. matches the dot in the filename
\([Jj][Pp][Gg]\) sets up tagged group \2, which will match jpg (or JPG, jpG etc)
$ anchors the search pattern to the end of the clipboard contents

The replacement expression assembles our Prefix, group one, dot, and group 2 in order to give the new filename.
{Return} at the end of the Clipboard line presses Return to complete the rename.
{Down} moves down to the next file.
{Endloop} moves control back up to the top of the Loop.
{:end} is the label which the "goto end" will jump to.
{Return} presses Return once more to take the last file out of rename mode.

If you wanted to restrict the rename to a specified number of files you could change the {Loop} line to:
{Loop {Ask "File rename" "Enter number of files to be renamed"}}

If/then/else based on existence of a window

Say you want KeyText to do different actions depending on whether a specified window exists or not. If the window exists you want it to continue the current item, but if it does not you want it to start another item instead.

Use the Insert field wizard to insert a Wait field. Note that although the Wait field can be used to wait for a specified window, you can also use it with a short wait time to establish whether the window exists or not. Select in the wizard: Wait for a window whose title includes (and enter the window title), Check all windows, found, and Start another item. Select a short wait time, click Next and choose the other item. On clicking Finish you will get something like:

{Wait 0.1 *!"My Window" if not found link X}Continue with window found actions…

If the window exists the item continues as above, but if not it will link to item X and run it instead. Note that you could use Call X instead of Link X if you wanted some extra actions to be done if the window is not found - and then continue with the current item. The * above indicates check all windows, and the ! indicates match the window title exactly.

The above item could be scheduled to run at a certain time each day. It could also be easily altered to base its condition on the existence of a specified file.

Another example shows how you can get the same hotkey to type different text depending on the active window.

Say you want KeyText to type Notepad if it finds Notepad at the front, or Word if it finds Word; if neither, do nothing.

Use the Insert field wizard to insert an "If title" field. Select in the wizard: Macro control, and click Next; then select "If clipboard text / window title…" and click Next again. Choose to check Window title, and specify that it contains the text Notepad - and select if "yes" and "continue". On clicking Finish you will get

something like:
{If title contains "Notepad" then}

Note that you can just type the field in directly if you wish. Then add further text so that your complete macro becomes:
{If title contains "Notepad" then}Notepad
{Elseif title contains "Word" then}Word
{Endif}

Set a hotkey for the item - say, Ctrl-Shift Z. When you press the hotkey, KeyText will type Notepad, Word, or nothing - depending on the title of the foreground window.

Set up alarms or reminders

You can turn a text item into an alarm/reminder by setting a schedule for it. The text item should contain just a "display a message" field, which may be preceded by a "play a sound" field. Choose a suitable empty item in the Edit text items window and follow these steps. Note that you can also schedule programs to run, etc.

Step 1 - optional. If you have a suitable sound file (.wav), select the Insert Field Wizard, click the 🔈 button and press Next. Use the browse button to locate your .wav file. Note that if you put this field after the message, the sound will not play until after the OK button on the message has been clicked.

Step 2. Click the Insert Field Wizard button, click 🗔 and select Next. In the resulting dialog, enter a title and text for the alarm/reminder message you want, and click Finish.

Step 3. Click the ⏱ button on the toolbar and select "Start text item ?:", with the appropriate interval in the combo box - click the arrow to the right of the box to show the options: choose Weekly, Yearly, Once, etc. Then select the time and date you want and click OK - and OK again to close the Edit text items window.

If KeyText is running with the correct .kt3 file open at the time selected - and the scheduler has not been suspended - the alarm / reminder will be displayed on top of any other windows. Note that KeyText will not do anything else until the reminder has been cleared by pressing OK or Cancel; if another scheduled event occurs while the first reminder is displayed, KeyText will wait until the first reminder is cleared before displaying the second (see Tip 3 below).

Tip 1: You can have the current date or time as part of the message displayed; after setting up the field with the Insert Field Wizard, use the Wizard again to insert a date/time field at the point required.

Tip 2: You can specify a monthly reminder (or run a program etc.) that will run on, say, the 3rd last day or every month. Set up a monthly schedule and, on the date of each month, scroll down the list to "Last day - 3". You can choose Last day minus 7 up to the Last Day.

Tip 3: To avoid KeyText being locked up until you click OK or Cancel, you can turn your message box into a prompt which appears for a specified length of time and then disappears. You can set this up in Step 2 above by selecting "Automatically close after n seconds".

See also the KeyText organizer and the Set Schedule Window.

Set an hourly chime

Step 1 - Choose a suitable sound file for the chime, for example chime.wav (supplied with KeyText). In the Edit text items window select the Insert Field Wizard, click the 🔈 button, press Next and use

Browse to locate the .wav file you want - then click Finish.

Step 2 - Click the ⏲ button on the toolbar and select "Start text item ?:", with the hourly interval set in the combo box, and 0 minutes past every hour selected. If you don't want it to chime every hour, un-select some of the hours shown on the dialog. OK that.

If KeyText is running with the correct .kt3 file open at the time selected - and the scheduler has not been suspended - the chime will play. Note that if a KeyText dialog is displayed at the time the chime is expected, it will be delayed until the dialog is closed.

Tip: A chime doesn't have to be on the hour; you could set up 3 more items - for 15, 30 and 45 minutes past the hour, with a different .wav.

Collect pieces of text
As noted above you can add text using the Edit text items window - and note that you can drag and drop text files onto that window if you wish.

KeyText also includes features to copy text to items without going through this window, either replacing text already there, or adding it to the end (or the beginning, if you change the setting in Options).

For example, if you have text in a database which you want to add to the end of a KeyText item, select (highlight) the text in the usual way, left-click the KeyText icon, move the mouse to the "Add selected text to end of >", and when the list of items appears, click the one you want. The text is added (note that the clipboard is used for this).

If you have several pieces of text to collect, possibly from different applications, consider using KeyText's Auto-add feature, which you can switch on and off from the Options dialog. This normally keeps adding text to the end of an item, but you can change this in Options. When on, every time you copy text to the clipboard - pressing Ctrl-C should do it - KeyText will automatically add a copy of it to the end (beginning) of the item you specified. A yellow icon in the tray accompanied by a stapler sound alerts you every time this happens - in case you forget it is on.

Note that when adding text to an item (rather than replacing it), you can choose separating characters etc. in Options.

You can paste or type the accumulated texts in the usual way.

For more information, see How to get text in...

Use KeyText for online chat etc.
Some KeyText features were developed in consultation with online chat users, who reported that the "macro" feature meant that text could be typed beforehand in a KeyText item, then "played back" online using KeyText. This worked well in early (pre-release) versions - almost too well, and the person "at the other end" could tell that it was being typed automatically because of the speed and regularity of typing.

The "simulate manual typing" speeds were developed with this in mind. In Options choose a speed to the left of center (the further left, the slower it will be). When typing, KeyText will vary the gap between characters, depending on what the characters are. For example, capital letters and punctuation marks will attract a slightly longer gap than lower case letters, etc.

Pauses can also be useful. They can be included in text - see Insert a pause - or invoked by pressing the Pause or f12 key during typing, or right-clicking the KeyText icon. Press Pause or f12, or right-click the icon again, to resume. You could include a pause, for example, at a point where you want to type manually to discuss preceding text with the other person.

Tip: The inclusion of deliberate mistakes and pauses adds to the illusion. For example, you could include in the text item:
{Speed 3}The incul{Pause 0.5}{Back 2}lusion of delibea{Pause 0.5}{Back}rate mistakes...
The Insert Field Wizard helps you set up any key combinations.

Add a random quote to an e-mail signature

Say you have your e-mail signature stored in item A of a .kt3 file, and you want to have one of four quotations etc. added to the end of it. Store the quotations in items B to E, then go back to item A, place the caret at the point you want the quotation to begin, select the Insert Field Wizard, click and press Next.

In the resulting dialog, if you want the quotations to be added in sequence - the first time you select item A, quotation B is added, next time C is added, then D, E, B, C etc., select Sequential, or Random to have one chosen at random. In the box below, select items B, C, D and E and click Finish.

This adds a link to item A; whenever A is selected, one of your quotations is added. Note that the items can be set to macro or paste, one can be macro and the other paste, etc.

Tip: The random function isn't restricted to text. You could have a KeyText item with only a link to one of a range of items; they could, for example, be set to run a game, go to a www site, etc. Select the item with the link and guess which game, www site etc. will come up next!

Manipulate time and date

KeyText offers the ability to type a date after today's date. For example, {Date+1} will type tomorrow's date.
But if you want to add, say, a specified number of hours, then integer variables can be used. Here is an example, and we will examine it line by line.

```
{IgnoreBlanks}
{#y@year}
{#n@month}
{#d@day}
{#h@hour}
{#m@minute}
{#a=DaysInMonth {#n},{#y}}
{#v=Input "Hours" "Enter number of hours to add (up to 24)"}
{if #v>24 cancel}
{#h+=v}
{if #h>23 then}
        {#d+=1}
{Endif}
{if #d>a then}
        {#d=1}
        {#n+=1}
{Endif}
{if #n>12 then}
        {#y+=1}
{Endif}
{Message "Result" "{#n "MMMM"} {#d} {#y}, {#h "HH"}:{#m "02"}"}
```

The {IgnoreBlanks} makes sure that white space - including returns and tabs - are ignored.
The next 5 lines associate the integer variables y, n, d, h and m with year, month, day, hour and minute.

That means that for the rest of the item these variables, if used as in, for example, {#y}, will give the current year, etc. (Unless that assignment is broken by setting it to something else - for example, {#m=0} - m would no longer return minutes.)
{#a=DaysInMonth {#n},{#y}} sets variable a to the number of days in the current month.
The next line produces a dialog asking you to enter the number of hours to add, and sets variable v to the number you give; if that number is greater than 24, the item cancels in the next line.
The subsequent lines add the v to the current hours, then increment the day, month and year if necessary.
Finally a dialog is produced with the result, showing the new contents of the variables in date format. It might display, for example, July 1 2015, 22:06. Note that "MMMM" indicates you want the full month name; "MMM" would give Jul.

Type with your mouse

For convenience, KeyText offers a Keypad, which lets you "type" with your mouse or other pointing device. Right-click the KeyText icon and select KeyText Keypad. The resulting window has 2 tabs, one resembling a miniature keyboard, and the other listing most of the symbols and accented characters you may need. The Keypad will stay on top of other windows until you close it. See KeyText Keypad.

Work with more than one KeyText data file

KeyText stores its data in files with the extension .kt3. You could have lots of these - personal.kt3, business.kt3 etc. To start a new .kt3 file, right-click the KeyText icon and select New. A new (empty) Edit text items window appears; once you have entered some text and clicked OK, you will be asked for a name for the new file.

In the Evaluation Version and the full (unlocked) version, the right-click menu shows the current .kt3 file as number 1, and (up to) the previous 4 as numbers 2-5. If you click on one of them, the indicated .kt3 file is opened.

The possibility of setting hotkeys that open the left-click menu is very useful if you work with more than one .kt3 file. Go to Options and set a different hotkey for each .kt3 file, and check the "Keep hotkeys active" box in global settings. Then look at the right-click menu and note that these hotkeys are listed to the right of the .kt3 files listed as 1-5. Note that from version 1.2, if your keyboard has a Windows logo key it can be included in a hotkey combination (although note that a number of Windows logo key combinations are reserved for Windows use, and you may not be able to set them).

This gives a "transparent" way of switching from one .kt3 file to another. For example if you set Ctrl-Shift-P for your personal.kt3 and Ctrl-Shift-B for business.kt3, pressing the appropriate hotkey brings up the correct left-click menu - ready to select the item you want. (Note that hotkeys to individual text items are only active for the currently open .kt3 file).

If you purchase KeyText, you get 224 extra items per .kt3 file - and also the possibility of running more than one at the same time. You could have one set up with random quotes for your e-mail signature, another for scheduled items and another for your favorite texts - all running at the same time. Up to 9 instances can be run; see multiple instances for further information.

Use KeyText to populate a database / address book

Some databases, address books etc. do not have an import function; if you already have all the data in text form - or can export it from an existing database as a text file - this can be very frustrating. You may have to do some computer programming, or type it all again.

KeyText can help! Try typing a few records to the database / address book manually, noting all the keypresses required to move from field to field and record to record. If a dialog appears for each new record, note also how the Tab key functions, and what buttons may need to be clicked.

Now take the text file with your data, and adapt it to mimic a manual entry. This may need to be done in a text editor with search/replace functions. Often [Tab]s between fields and [Return] at the end of each record will do it, but you may need to use fields such as {Click "New..."} to add a new record, etc.

Experimentation is required, but even if it takes a little time, it should be worth it. One beta-user has written: "A particular application was the migration of a client's word-based mail list into Exchange Address books. We designed an exchange form, exported a flat file into KeyText, and an hour later had a fully populated database - without the hassle of visual basic or other coding."

Note also that you can use KeyText's Loop field to repeatedly perform all or part of a KeyText item; you'll find it in the Insert field wizard Macro control ⏱ section.

The if/then/else functionality of KeyText’s Wait field may also be useful in this context.

> For example if, when you simulate the [Return], your database occasionally gives a warning with a title bar "Duplicate entry" and text "An entry with field xxx already exists. Add anyway?", you may wish to click Yes and continue every time. Normally KeyText would fail, because it would have continued to the next item.
>
> After the {Return} which KeyText sends to complete the entry you could add:
> {Wait 1.0 *"Duplicate entry" if found call B}{Endloop}
> Item B would contain:
> {Click "Yes"}{Pause 0.5}
>
> If, within 1 second, the "Duplicate entry" dialog appears, KeyText will call item B which clears the dialog, and then continue the first item to endloop - when it will begin the loop again.

Use KeyText in a network environment

One full version copy of KeyText may be used by a single person who uses the software personally on one or more computers. Site licenses are available, as well as quantity discounts.

In a network environment it is recommended that KeyText be installed and run on each computer, but KeyText data files can be on a shared folder on another computer if desired. Click Open... on KeyText's right-click menu, and locate the appropriate file through Network Neighborhood.

KeyText can be used to standardize texts, for example, at a customer support center giving support by e-mail. It is recommended that once the contents of the shared .kt3 file have been finalized, it should be set to read-only (using Windows Explorer). This will avoid the possibility of one user altering data shared by a number of people.

Additionally, KeyText's string variables have been set up with network use in mind. Global strings can be shared from a single gsv file on the network, allowing common text strings between users. For example, say a company has a featured product of the day, which is set every morning. A user with password access could set a global variable ProductOfTheDay@ to the featured product name, set it to prevent changes, and then every user could have access to it - and it could be referenced in a number of different items.

Note that KeyText can be told from a command line to start, run a specified item, and exit. If you can send a command line to a remote machine, then you can have KeyText perform functions for you remotely.

KeyText can also be unlocked in this way; a command line could be run remotely supplying the name and unlock code required to unlock KeyText on the target PC.

Another useful feature in a network environment is the Wait field, where KeyText can be set to wait for a specified file to exist - wait-to-go. Where two or more PCs are involved in a procedure and one has to wait until the other has completed its part of the task, if the latter creates a dummy file to signal its completion, KeyText can use that to trigger the rest of the macro on the waiting machine.

> For example, say a software build process involves 2 PCs, and the first creates a file c:\ready.txt to indicate its completion. If a KeyText item is running on the second machine with:
> {Wait 0 $"c:\ready.txt"}... rest of procedure
> KeyText will check if the file exists every 15 seconds, and if it is found, will continue the rest of the procedure. Note that the 0 denotes an endless wait; if the file does not appear it will wait until canceled by pressing Esc or left-clicking the KeyText icon in the tray. Alternatively specify the number of seconds to wait, up to 1000000 (over 11 days!).

2.2 Startup

What happens when KeyText starts?

The first time KeyText is run, it asks for your name. Your name will subsequently appear at the top of the left-click menu, to give a quick and convenient way of entering your name in an application without typing it.

The startup box then appears, to welcome you to KeyText. It will also remind you when your Evaluation Period is about to expire. When you OK the startup box, you may think that KeyText has disappeared, but look at your system tray - there should be a new icon there, next to the time.

If you are running KeyText for the first time, click the icon using your left mouse button. A menu appears with the sample text items which demonstrate some of the KeyText features. Try out some of these, starting with item A, and then trying the others with a text editor or word processor running. You can change these sample texts to anything you like. Right-click the KeyText icon and select "Edit text items" to see where they are stored - you can delete or edit them there.

KeyText 3 stores its data in files with the extension .kt3. When first run it looks for KeyText.kt3; after that it locates the .kt3 file you were last using. This may be KeyText.kt3, or another .kt3 file if you have used "New..." to make new .kt3 files.

KeyText therefore always offers you the last .kt3 file you were using. If you want a different one, you can open it using the previous file list on the right-click menu - or by pressing a hotkey if one is indicated on the right-click menu, or through the Open... dialog.

Note that in the Options dialog you can specify a text item which will start automatically when that .kt3 file is opened. This may be useful if you want KeyText to automate some of your logon procedure; an item could be set to pause for a set time (to allow other applications to start), then activate various windows in turn, and perform any specified actions.

Also note that if you have used the Scheduler to set an item to start after startup and at n minute intervals, it will first run a few seconds after KeyText starts, and the next time n minutes after it has completed, and so on.

If you have a full version of KeyText (purchased and unlocked), you can run more than one copy at a time - known as multiple instances. You can turn this on in the Global settings. Use this in combination with different icons to have up to 9 KeyTexts running at the same time, should you wish. To make it easier to run more than one copy at a time, you can set KeyText to automatically run multiple instances, using your previous file list to locate the files.

Note that it is recommended to Exit multiple instances either by letting Windows do it for you at the end of a session, or by choosing Exit All KeyTexts from the right-click menu. You can close each one using Exit, but because KeyText loads the last closed .kt3 file first, the order of loading may alter.

If you have set KeyText to run multiple instances automatically, you can override this by holding down the SHIFT key while KeyText starts. This also suspends the scheduler.

SHIFT key and start
If you hold down the Shift key while KeyText starts, this will have 4 effects.
The Scheduler will be suspended - scheduled items will not start.
Any Auto-start item set in the Options dialog will not start.
If you have set additional KeyText instances to start, in Global settings, they will not start - only the current instance.
If KeyText is set to run with no icon in the tray, an icon will display - convenient if you have forgotten the hotkey you set!

Setting KeyText to start when Windows starts
On installing KeyText, the option is given to add it to your Startup menu, so that it is started automatically each time Windows starts. You can change this later in the Global settings, adding or removing it from your Startup menu. Note that the startup setting applies to the current user.

Using the "number of extra KeyTexts to run" setting in Global settings, you can have multiple instances starting automatically - based on your previous file list.

If you want KeyText to always start with the same .kt3 file, a different approach may be required - normally KeyText will start with the last .kt3 file you were using. To start one (or more, if purchased) .kt3 file each time, add the .kt3 file to your startup folder, rather than KeyText.exe.

To do this, right-click the Windows taskbar and select Properties; choose the "Start Menu Programs" tab. Click Add... and follow the instructions, adding the .kt3 file and selecting "Startup" (or similar) for its location. Alternatively, using Windows Explorer, right click and drag the .kt3 file; drop into the StartUp folder; choose "Create Shortcut Here" from the popup menu.

2.3 How to get text in

Each instance of KeyText has 234 text items you can use (20 in the evaluation version). This page describes how you can place text in an item, ready for reuse later. The 234 items are grouped in the left-click menu and the Edit text items window in 9 groups of A-Z; that is, A-Z, followed by 2A-2Z up to 9A-9Z. For convenience you can give a name to each A-Z group, which will appear in the title bar of the “Edit text items” window, and in the left-click menu label and “Go to group>” option; you can set these names by clicking “Group names” in the Organizer window.

Edit text items
On the right-click menu, select "Edit text items" or, to go directly to edit a certain item, choose "Edit text in item >" from the left-click menu (in the full version you can click Next… on the left-click menu, then "Edit text in item >" for items beyond Z). An even quicker way to reach the item you want to Edit is to right-click it in the left-click menu - but make sure that right-click to edit item is set in customize settings.

The Edit text items window will appear - a window with a row of tabs along the top, one for each text item. Add or edit text as you would with any text editor, or drag a text file from Explorer and drop it into the window. Don't forget that the edit window is fully re-sizeable, and if you have an Intellimouse, you can scroll using the Wheel. The Insert Field Wizard on the toolbar gives access to all the fields available for various functions and features.

Select text in a document

If your text is already in, for example, a word processor document, simply select it in the normal way (so that it is highlighted), left-click the KeyText icon, and move the mouse cursor to "Copy selected text to >" if you want to replace any existing text in the item, or "Add selected text to beginning (end) of >" if you want it added to any existing text. When the mouse is over either of these menu items, a further popup menu appears with a list of items - click the one you want (an "e" or "empty" indicates the empty items). A copy of the selected text is placed in the selected item. (Note that the clipboard is used to do this, but that you do not require to copy the text to the clipboard - selecting it is all that is required for KeyText to find it).

If you want to set up a particular item to add to - saving a keypress or mouse-movement compared with above - go to customize settings and select the required item in the combo box beside "Include - Add selected text to item:". From then on, to add text to that item, left-click the KeyText icon or press the left-click hotkey, then select "Add selected text to item A (etc.)" or press the underlined number. Note that when KeyText is first run, this option is switched off in customize settings; check the appropriate box and the option will appear.

The Options dialog determines how text is added; you can specify the beginning or end of the text item, and choose how you want it separated from existing text - new line, horizontal line, a space, etc.

Directly from Windows Explorer

Select one or more text files in Explorer, left-click the KeyText icon and - as above - move the mouse cursor to "Copy selected text to >" if you want to replace any existing text in the item, or "Add selected text to beginning (end) of >" if you want it added to any existing text. When the mouse is over either of these menu items, a further popup menu appears with a list of items - click the one you want, and the selected file(s) are placed in the selected item - with options as above.

This provides a very easy way of getting text from a closed text file into an application. Select it in Explorer, left-click and move the mouse cursor to the appropriate point to copy it into KeyText. Then switch to your application, left-click KeyText again and click the item you stored it in - it will be pasted or typed in for you.

Using Auto-add clipboard text

If you want to assemble pieces of text from various locations to make a larger text ready to go somewhere else, try Auto-add. Right click the KeyText icon and select Settings, where you can switch Auto-add on and off, and choose the item to receive the clipboard texts. When on, the KeyText icon in the tray will change to show a small + sign in the top right-hand corner. From then on, until you turn Auto-add off, any text you copy (or cut) to the clipboard will also be added to the selected text item - to the beginning or end, and with separators as specified in Options.

For example, to get some paragraphs or text together from different applications, turn on Auto-add, select and press Ctrl-C for each. That's all - they are now assembled in a KeyText item ready to be pasted or typed somewhere else. Copying a text file in Explorer will also add the text in it to KeyText.
It is easy to forget that Auto-add is on! Therefore a visual reminder (yellow icon) is shown each time KeyText is activated in this way, along with a stapler sound. A reminder is also given after some time to indicate that it is still on (you can choose not to be reminded again).

2.4 How to get text out

IMPORTANT - firstly you need to know how to abort KeyText if an item is running as a macro (typing text for you), and you want to stop it before the end. Either left-click the KeyText icon in the tray, or press ESC. To pause typing, right-click the KeyText icon, or press the Pause or f12 key. Once in pause mode, indicated by an animated icon in the tray, right-click or press Pause/f12 to resume typing, left-click or Esc to abort.

Once you have your text stored, there are various ways of getting it out, to either paste or type it into the foreground window, or to run it (as if a batch file) if you have selected the run option.

Left-click the KeyText icon in the tray, and a menu will appear showing the text items you have stored. Click the item you want and it will be inserted at the caret of your active window. Alternatively, open up the menu with a left-click then type the underlined letter associated with the item to be inserted. If you have the full version you can click select Go to group> and choose the group (A-Z set) you want to jump to; or click Next… on the left-click menu to show the next sets of A-Z - and then Back… to move back. The "+" and "-" keys, including those on the numeric keypad of your keyboard, are equivalent to Next… and Back… - so navigation of the menus and selection can all be done from the keyboard.

A similar menu can be set to appear using KeyText's "right-click anywhere" feature. You may have set this up when KeyText was installed, or you can set it up or change it later in the Global dialog. Set it so that Ctrl and/or Shift with a right-click where you want the text to go opens up a KeyText menu; select the item you want, and it's typed or pasted in at the point you clicked. This cuts down on the number of mouse clicks you need; no longer do you have to left-click the box on a form, then left-click the KeyText icon and select the item you want. One click with a keypress and another click to select your text is all that's required. Note that the right-click anywhere menu does not show empty items or items with a schedule, and that you can set it to start at a different place from item A. Your form-filling texts could therefore be set to start at, for example, 4A, and would appear top of the list in the new menu.

Or you can specify a keyboard shortcut to an item. In the Edit window, click the Hotkey button, then click the hotkey box beside it and press the key-combination you wish to use. From then on, any time you press that combination of keys, the associated text item will by typed or pasted.

New to version 3 of KeyText is Trigger Text. Instead of setting a hotkey, you can define a sequence of keys for KeyText to look out for. To set it up, type it into the second text box on the right in the Edit text items toolbar. KeyText will then monitor keypresses and, on finding a match, will delete them - and start the relevant item. You can use this to set abbreviations to launch commonly used KeyText items; see trigger text for more information.

You can choose to have KeyText react differently if a KeyText item is right-clicked in the right-click anywhere, or in the left-click menu. The choices, which are set in customize settings, are 1) no difference (right-click is the same as left-click), 2) do the alternative action (an item set to Action: Macro is pasted, and an item set to Action: Paste is treated as a macro), 3) add a Return, 4) add a Tab, or 5) go straight to the Edit text items window for the item. So, for example, if you had the Tab setting and were filling in a form using the left-click menu, you could place the cursor in the Name field, then right-click your name on the menu. It would be typed or pasted in, and move you to the next field automatically - and so on.

You can also start a specific KeyText item from a command line, or you can set up desktop shortcuts to individual KeyText items which, depending on the setting chosen, will run and then leave KeyText running, or run then exit automatically. See Desktop shortcuts for more information.

There is a shortcut to the text stored as item A - simply double-click the KeyText icon with the left mouse-button (unless you have chosen in Settings / Customize to have a double-click open the Keypad). You can store the phrase you use most often there, for ease of access.

You can also set up a keyboard shortcut to bring up the left-click menu. Then type the letter associated with the item you want and the text is inserted. Your hands have never left the keyboard! This option is also useful if you do not normally have your Windows taskbar visible. Go to the Options dialog to set this up.

A powerful feature of KeyText is that if you have more than one .kt3 file, KeyText remembers the ones you have used, and can keep previous left-click hotkeys active - they are indicated in the right-click menu. So you can invisibly switch from one .kt3 to another simply by pressing the left-click hotkey for the .kt3 file you want.

See Organizer for ways in which a text item can be set to run itself automatically.

2.5 Left-click Menu

Clicking the KeyText icon with the left mouse button opens a popup menu showing the text items you have stored; where items are empty, this is indicated and the option is grayed (disabled).

A left-click on the icon while KeyText is typing text will cause it to abort - similar to pressing [Esc].

Note that a left double-click on the KeyText icon will automatically select text item A (unless you have chosen in Settings / Customize to have a double-click open the Keypad).

At the top of the left-click menu is the **User Name** you entered when you first ran KeyText, or entered in the User Information dialog. If you have purchased KeyText, it will be the name given with your code.

If you have specified a **group name** for the current A to Z group in the Organizer, it will be displayed as a label under the name.

Below this are **items A to Z**, with the menu text for the item (or the first 25-30 characters of text if the menu text is blank) also shown. If graphic menu style is on (customize settings) an icon indicates whether the item will be typed, pasted or run.

To select an item, either click it with the mouse, or press the underlined character. If a hotkey is set, this is indicated on the menu. Selecting a valid item - will cause the action indicated - Macro, Paste or Run. Note that a right-click on the item can be set to give a different result: do the alternative action (an item set to Action: Macro is pasted, and an item set to Action: Paste is treated as a macro), add a Tab or a Return at the end of an item, or open the Edit text item window for the item. You can set which one you require in customize settings).

After items A-Z you will see **Back…** and **Next…** (in the full version) so that you can move forwards and backwards through the groups. Click these to display 9A-9Z, 2A-2Z and so on.

The "+" and "-" keys, including those on the numeric keypad of your keyboard, are equivalent to Next… and Back… - so navigation of the menus and selection can all be done from the keyboard.

Go to group is the next option (in the full version), and it opens a submenu which can take you directly to the selected (A-Z) group; it will display any group names you have set in the Organizer.

Below this are four popup submenus (which can be turned off in customize settings); move the mouse over the item and the submenu appears showing a list of items and indicating which are empty. Click one of them, and one of the following occurs:

Edit text in item >
The Edit text items window is opened at the item specified. *Once you have got used to it, this is the quickest way of getting to an item to edit it. Click the* Next... *button until the item you want is displayed, rest the mouse over the* Edit text in item > *and click the item to edit.*

Copy selected text to > *
Any selected (highlighted) text in your active window (Word Processor for example), or selected text file (s) in Windows Explorer, will be copied to the item specified, deleting any text already there.

Add selected text to beginning (end) of > *
Any selected (highlighted) text in your active window (Word Processor for example), or selected text file (s) in Windows Explorer, will be added to the item specified, at the beginning or end, and with a separator, as defined in Options.

Add selected text to beginning (end) of X *
This item is only present if the "Include - Add selected text..." option in customize settings is set. This does the same as "Add selected text to beginning (end) of >" above, but without the popup submenu selection - instead the text is added to the designated item. The purpose of this setting is to save a keypress or mouse movement - useful if you are adding a lot of text snippets to an item.

Right-click menu
Note that you can choose to omit this in customize settings. Selecting this simply opens the right-click menu. The purpose of including this is mainly so that all KeyText functions can be accessed without using a mouse. A keyboard shortcut can be set to open the left-click menu; then pressing the underlined number beside "Right-click menu" will take the keyboard user to the right-click menu and all its functions.

*Note that the clipboard is used for selections above marked *, but that you do not require to copy the text to the clipboard yourself - selecting it is all that is required for KeyText to find it.

2.6 Right-click Menu

Clicking the KeyText icon with the right mouse button opens a popup menu showing various options.

Purchase today... select this to go to the MJMSoft Design www site where you can purchase KeyText online if you wish. This item is only shown if KeyText detects that you have a browser, and access to the Internet; it disappears once the unlock code is applied.

Help... displays help.

About opens the About dialog - similar to the welcome box which appears when KeyText starts. If you are using KeyText during the Evaluation period, this box also tells you how many days to go. If your KeyText is unlocked, the About dialog will include a button to allow you to check for updates.

Shift Icon shifts the KeyText icon to next to the clock, which may be convenient if you have a number of icons in your tray.

New... creates a new set of KeyText text items, and opens up the Edit window so that you can begin defining the new file. On selecting OK, you will be prompted for a name for the new file, with the extension .kt3. If Cancel is selected, the .kt3 file being used immediately before selecting "New" is restored. If there was no .kt3 file being used before, which may happen if KeyText could not find its .kt3 file on startup and you asked it to start a new one, the Cancel button becomes Exit.

Open... displays the open file dialog box so that you can locate and open a previously stored .kt3 file.

Note that there is no Save option, as the .kt3 file is automatically saved when OK is selected from the Edit window

File List shows the current .kt3 file as number **1** and any previously used .kt3 files, numbered **2** to **5**. Note that the previous files are not shown in the expired version, nor in any extra instances of KeyText. If more than one KeyText is running, only one will display the previous file list.

Note that previous .kt3 file hotkeys can be kept active if that option is set in Global settings. Any set hotkeys are indicated here, and this powerful function means that clicking any hotkey indicated will open the left-click menu for the appropriate .kt3 file. This switching of .kt3 file in the background means that various sets of text items can be available at the press of a key.

Password lock now is only visible if a password has been set and enabled (except for password level 4), and offers a quick way of locking KeyText. A password is then required before the left- or right-click menu will open, and before any hotkeyed or scheduled items start. Once a password has been supplied, KeyText reverts to the password level previously in force.

Edit text items... opens the Edit window, which allows you to define, view, or edit the text items you wish to store. The first time it is opened it will take you to item A; after that it will go to your last edit point.

Settings... opens the Settings dialog.

Organizer... opens the Organizer.

Suspend scheduled items is only visible if scheduled items are set, and provides a quick way of temporarily stopping KeyText from running items set to start at a specified time or regular interval, or when a certain window appears.

Suspend right-click anywhere is only visible if the "right-click anywhere" feature has been set when KeyText was installed, or later in Global settings. It provides a quick way of temporarily stopping KeyText from generating it's "right-click anywhere" menu if the right mouse button is clicked while the set Ctrl and/or Shift keys are pressed.

Suspend trigger text is only visible if one or more trigger texts have been defined (in the second text box from the right in the Edit text items toolbar). It provides a quick way of temporarily stopping KeyText from monitoring the keyboard for trigger text.

KeyText Keypad... opens the Keypad

The following two items determine the action to be carried out when a text item is selected on the left-click menu, by hotkey, or by Scheduler - and has "Default action" selected. The currently selected option is indicated.

Set default action to Macro - the text will be inserted at the caret of your active or foreground window by simulating keyboard input, at the speed determined by the setting in Options.

Set default action to Paste - the text will be copied to the clipboard and automatically pasted in at the caret of your active or foreground window. This option is generally much quicker than Macro.

Exit All - this item only appears if you have more than one copy of KeyText running (multiple instances - full version only). If you want to close down several KeyTexts during a Windows session it is recommended that you use Exit All, so that the next time they are run together, they start in the same order. Otherwise - last exited is the first to start, etc.

Exit exits the program.

A right-click on the icon while KeyText is typing text will cause it to pause; right-click again to resume.

2.7 Paste, macro, run... what's the difference?

The quickest way to get text from KeyText into your application is by setting it to paste. Select this in the edit window, or, if the default action is specified, you can change the default to paste in the right-click menu.

Pasting does just that. It automatically copies the specified item to the clipboard and pastes it in at the caret of the foreground window. Before it does so, it checks for any comments, which are removed, any "Include text", Clipboard or Selection fields which are expanded, and any Ask, Date or Time or variable fields, which are filled in.

Macro simulates typing, at the speed set in the Settings dialog. It is slower than pasting, but has other advantages. The clipboard is not used, thus preserving its contents; and if your word-processor has auto-correct or auto-format capabilities, these will be available. For example, if your word-processor automatically superscripts ordinals such as st, nd, rd, th etc., this will only happen if the method chosen is Macro.

Watching the text appear is also intriguing, while the KeyText icon gives a visual clue that typing is taking place.

When KeyText is "typing" in your text, you can abort by pressing [Esc] or by left-clicking the KeyText icon.

If you want to pause typing, press Pause or f12, or right-click the KeyText icon - and again to resume. A visual clue is given of the pause status.

The other big advantage of Macro is that the text item can include fields - instructions to carry out certain actions - and works almost like a script. So you could set up an item to type some characters, then pause, then select a menu item, fill in a dialog, switch to another window, play a .wav file, and so on.

It can also include combinations of keys. For example, include {Ctrl B} in an item, and it will toggle bold on and off, if your word processor supports it.

The Insert Field Wizard on the toolbar of the Edit window makes it easy to set up such instructions.

The speed of typing varies according to the speed of your system, whether your hard-drive is being accessed, the complexity of your word-processor, and other factors. You may for example notice a pause after words if your word-processor is busy checking whether they need to be auto-corrected.

You can adjust the speed of typing by going to the Options dialog box. By experimenting with different speeds you can find one that works best.

Note that the slower speeds in the Options dialog simulate manual typing by varying the speed depending on the characters being typed. For example, slightly longer after a capital letter or punctuation; even longer after a new line, etc. Online chat users report that this - combined with some pauses and deliberate errors which are immediately corrected by using {back} to delete them - convinces the person at the other end that they are watching text being typed manually; the KeyText user meanwhile is enjoying a cup of coffee!

Tip: If you have an item which you sometimes wanted pasted, but at other times typed, there are two ways of doing this easily. One is to set it to Action: default - in which case the setting on the right-click menu can be used to change the action. Or you could set it in customize settings so that a right-click on the item in a menu makes KeyText perform the alternative action to the one set. With this, you could have it so that a left-click pastes an item while a right-click types it, or vice versa.

Technical note: As KeyText developed, there was the choice of making it "take over" your computer - in which case "typing" could only be stopped by pressing a particular combination of keys (Ctrl-Esc to be precise) - or not taking over, but leaving you much more control. It was decided that it was more "friendly" to let you keep control, giving the convenience of aborting etc. at the press of one key or one mouse-click. The downside of this is that you can interfere with the operation of KeyText - for example, by clicking another window and therefore changing keyboard focus. KeyText will carry on typing regardless; it may sometimes appear that KeyText is typing into thin air, if for example you click the desktop or taskbar while KeyText is typing. Such typing is harmless; simply press Esc to abort.

The "run" option is not available as a default, but must be specified in the Edit window. It treats each line as a command to be run, as with a batch file. If you have access to the Internet and a browser installed, you can specify a URL in an item, along with Run action, and every time you click it you will be taken to the specified www site.

If you want to include command line parameters or a default directory for the run program, use Macro mode and set up "Run" fields using the Insert Field Wizard.

2.8 Trigger text

Instead of setting a hotkey, you can define a sequence of keys for KeyText to look out for - type it into the second text box on the right in the Edit text items toolbar. KeyText will then monitor keypresses and, on finding a match, will delete them - and start the relevant item.

You can use this to set abbreviations to launch commonly used KeyText items; if desired you could start all your Trigger texts with a special character, for example #, but this is not required.

Trigger texts can be viewed in the Edit text items window, and they also have a column in the KeyText Organizer. Note that in both places and SPACE characters in the trigger text are displayed as · so that they can be seen. You can suspend all trigger texts using the KeyText Organizer, or by choosing "Suspend trigger text" in the right-click menu.

The following special keys can be included in trigger text: {f1}, {f2}….{f11}, {f12}, {Num *}, {Num +}, {Num -}, {Num .}, {Num /}, {Num 1}, {Num 2}, {Num 3}, {Num 4}, {Num 5}, {Num 6}, {Num 7}, {Num 8} and {Num 9}. For example, if you wanted the f10 key followed by 123 to start an item, use Trigger text:
{f10}123
Note that if you have changed the characters to designate a field (in global settings), you must use [f10] 123 or <f10>123.

You could set #dt to type the date - whenever KeyText detects that sequence of characters it would delete them and type the date.

/ad could type your address.

Try setting e// to type é or e\\ for è. Trigger texts are case sensitive, so you could set E// to type É

Use it to type long or difficult-to-spell words - for example, set ptfe to type polytetrafluoroethylene. Or it could even correct your spelling - set calander as the trigger text for calendar.

Note that KeyText does not record your keystrokes - it discards each keystroke after it has checked for a match.

If you want to prevent an item with Trigger text from starting in certain applications, use an If title or If exe field at the beginning of the item, making sure that the action is cancel.

For example, if you do not want an item with Trigger text to start in Microsoft Word or when editing a file called exclude.txt, put this at the beginning of the item:
{If exe contains "msword" cancel}
{If title contains "exclude.txt" cancel}

Note that there can be as many of these lines as you require, but they must be at the beginning, and the action must be "cancel". You can use the Insert field wizard to set these up, if required.

2.9 Smart Select

Smart Select is set up using the Insert field wizard button on the toolbar and selecting: Include string variable, clipboard, selection etc.

SMART SELECT As a Windows user you already know how to select text and press a key combination to copy it to the clipboard, turn it to bold, and so on. With Smart Select you can do much more with selected text.

The first two options on the wizard dialog allow you to work with the contents of the clipboard in an item, or to use any text currently selected in the foreground application. The latter gives KeyText its Smart Select feature.

These Clipboard and Selection fields will work with selected text - and also with text files copied from or selected in Windows Explorer.

The fields {StartClipboard} and {Selection} are filled in before the item starts to run, while {Clipboard} uses the current clipboard contents while the item is running. All three fields can be used as part of another field.

> For example, if you have a KeyText field with:
> {Run "mailto:{Selection}"}
> and have it set to Action: Macro, and with a hotkey of, say, Ctrl-Shift E, then if you select (highlight) an e-mail address in any application and press Ctrl-Shift E, your e-mail client will start with a new e-mail ready to that address.

Or you could select a cell in an Excel spreadsheet, and start a KeyText item which has:

> {Activate "MyApp"}{Selection X}{Tab}{Activate "MySpreadsheet"}

The first thing KeyText does is check if there is a Selection field in the item; on finding it, it copies selected text in the foreground window (via the clipboard) and uses it to fill in the Selection field - in this case the contents of a spreadsheet cell. KeyText will then start the item and switch to MyApp, type the cell contents and press [TAB], then return to MySpreadsheet. The X at the end of Selection tells KeyText to remove any carriage returns or line feeds from the end of a selection - Microsoft Excel, for example, adds a carriage return to the text format of a copied cell, and the X option ensures it is removed before being typed into MyApp.

This idea can be developed further so that KeyText can do different things depending on the contents of the selection.

For example:

```
{If selection like "^[[:alnum:]][[:alnum:]._-]*@[[:alnum:]._-]+\.[[:alnum:]]\{2,4\}+$" then}
        {Run "mailto:{Selection}"}
{Else if selection like "^[[:digit:]]\{5\}$" then}
        {Run "http://maps.google.com/maps?oi=map&q={Selection}"}
```

This uses regular expressions to check if the selection matches an email address; if it does, a new email is started to the selected email address.
Otherwise, if the selection matches a zip code, KeyText will take you to the Google Maps page for that zip.

Using KeyText's ability to extract text from selected text files, you can, for example:

Select one or more files in Explorer, copy them to the clipboard using Ctrl C, then use the following KeyText item:

```
{Clipboard}
```

If you have the item set to Action: Paste, then it will paste the contents of the text file(s) into the current application.

Note that in all cases the clipboard is used; the StartClipboard field uses the starting text contents of the clipboard, Clipboard uses the current text contents, and Selection copies the selected text to the clipboard first.

2.10 Edit text items window

This is the heart of KeyText - the place you go to add to or edit your text items, and set up options on each.

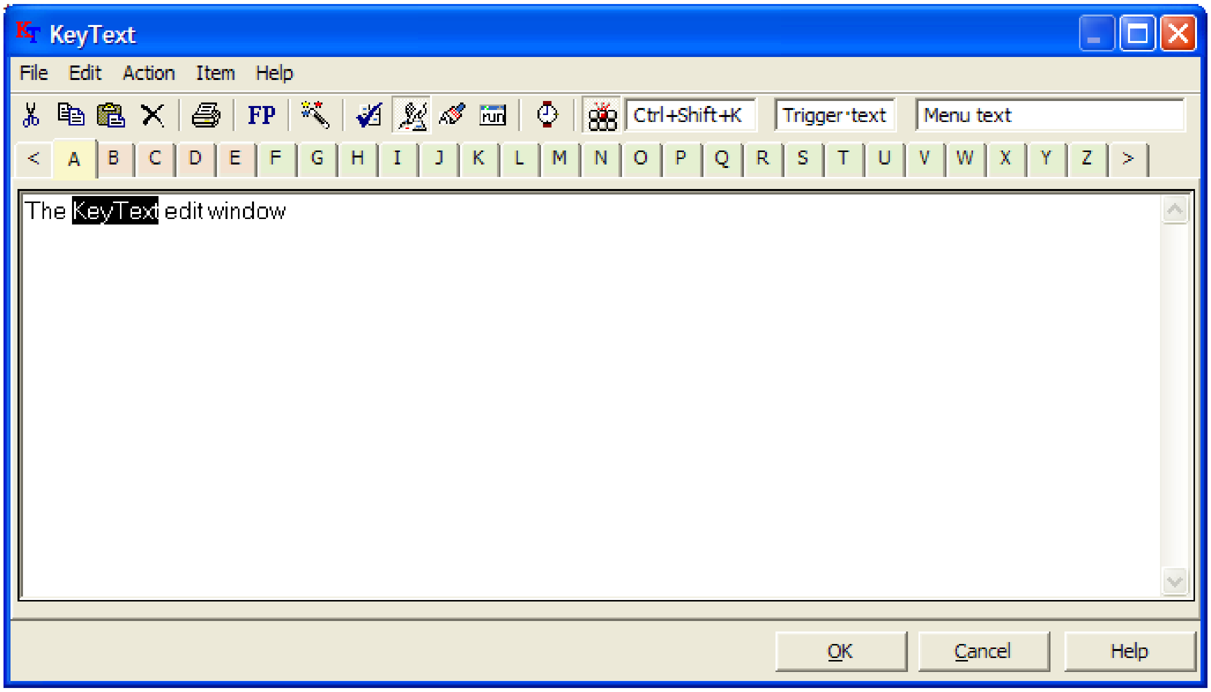
Tip: To navigate quickly to an item you want to edit: make sure you have the “right-click Edits item” option chosen in Settings / Customize. Then locate the item you want to edit in the left-click menu or right-click anywhere menu - and right-click it.

Tip: To quickly open the last item you were editing, select Edit text items from the right-click menu. Or - even quicker - set an "Edit hotkey" in Settings / Customize. Press that hotkey to open the last edited item - and the same hotkey to save it (or bring it to the front).

On the right-click menu, select "Edit text items" or, to go directly to edit a certain item, choose "Edit text in item >" from the left-click menu. Alternatively, if you have the “right-click Edits item” option chosen in Settings / Customize, locate the item you want to edit in the left-click menu or right-click anywhere menu - and right-click it. If you have the full version you can click Next… on the left-click menu to show the next set of A-Z, and then choose "Edit text in item >" for that set. The editor will appear - a window with a row of tab, one for each text item. (Note that as you move the mouse over the tabs, a ToolTip indicates the contents of that item.) Add or edit text as you would with any text editor, or drag a text file from Explorer and drop it into the window. Don't forget that the edit window is fully re-sizeable, and if you have a Microsoft® Intellimouse, you can scroll using the Wheel.

The menu bar gives keyboard users access to the toolbar settings, etc., without using a mouse.
Keyboard users also note the following shortcuts:
Ctrl-TAB and Ctrl-Shift-TAB to moves from item to item.
Ctrl-Shift-A takes you to item A, Ctrl-Shift-B to B, etc.
Alt-O and Alt-C press the OK and Cancel buttons respectively; Ctrl-S is also equivalent to pressing OK.

Click buttons on the sample Edit Window below to find out what they do.



Ctrl+Shift+K

Trigger text

Menu text

The KeyText edit window

Note the three colors displayed on the tabs. The green shade indicates an item which is empty, while the red shade indicates not empty. Items with a yellow tab have changed since the Edit text items window was opened.

Clicking the > on the right-hand tab moves to the next set of 26 text items (full version only). Note that if you right-click the main window, the edit menu popup will appear.

Because there is a shortcut to the text stored as item A - simply double-click the KeyText icon with the left mouse-button - you should store the text you use most often there (unless you have chosen in Settings / Customize to have a double-click open the Keypad).

2.11 KeyText fields

2.11.1 Insert Field Wizard

Click the Insert Field Wizard button in the Edit text items window toolbar to start the wizard.

Select the field you require, then click Next>.

Some of the fields will only work if the action specified at the top of the box is chosen. For example, play a sound or display a message will not work if the text item is pasted - only if it is set to Action: Macro.

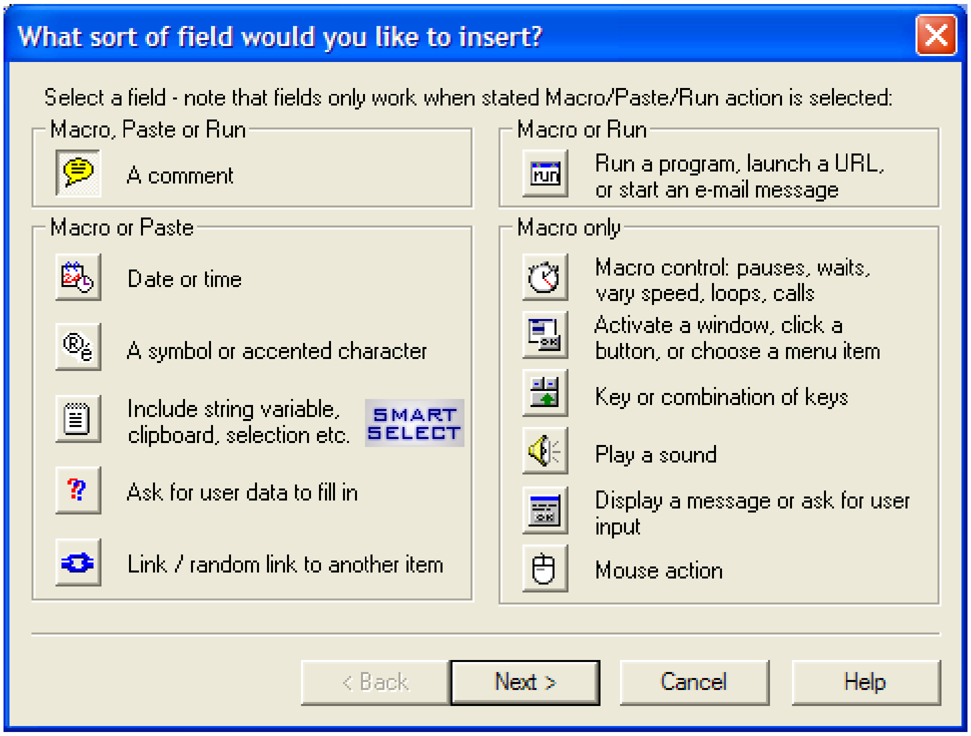
You can type fields manually, but the wizard helps to get the syntax correct. If the syntax is incorrect, the field will be treated as normal text, and typed or pasted in full. If a field fails when it is being run, that is, when the item it is in is being typed etc., the item will halt. For example, if you have a field: {Click "Help"} to click a help button, and none exists in the foreground window, the item will halt.

Fields are all enclosed in { and } characters. If you want a { to be included in an item as a normal character, you must type 2 together: {{. For example, "ascii art" may include a "flower" looking like: {@}. You must type it as {{@} in KeyText. Alternatively, you can change the field delimiters from {...} to [...] or <...> in global settings.

Note that if you add a [Return] after the closing brace of the following fields: Activate, Ask (but only if the no output modifier - is selected)**, Beep, Call, Click, Endloop, Input, Loop, Menu, Message, Mouse, New, Pause, Run, Speed, Wait, Wave - it will be ignored. This helps to lay out a series of fields more clearly. If you do want a [Return] to be typed by KeyText after such fields you can add a second [Return], or use {Return}. This feature can be turned off in the customize settings dialog.**

Certain fields can be "nested" inside other fields - Date, Include, Clipboard, Selection and variable (including Ask) fields. So, for example, you could have a Date field in a message box text, or have an Ask field ask for a file name as part of a Run field, etc.

Click buttons on the image below, or the list below that, to find out more about the fields; the appropriate actions are indicated.



What sort of field would you like to insert?

Select a field - note that fields only work when stated Macro/Paste/Run action is selected:

Macro, Paste or Run
A comment
Macro or Run
Run a program, launch a URL, or start an e-mail message
Macro or Paste
Date or time
A symbol or accented character
Include string variable, clipboard, selection etc.
SMART SELECT
Ask for user data to fill in
Link / random link to another item
Macro only
Macro control: pauses, waits, vary speed, loops, calls
Activate a window, click a button, or choose a menu item
Key or combination of keys
Play a sound
Display a message or ask for user input
Mouse action
< Back
Next >
Cancel
Help

For further information click here for Field syntax reference.

2.11.2 Comment

Type any comment text in the text window and press Finish. The text will be enclosed in {* and *} characters, and be ignored by KeyText when typing, pasting or running a text item.

2.11.3 Date or time

The wizard gives the opportunity to select what date should be used. You can choose "Current date", which refers to the date on which the text item is used - it is updated each time. You can also choose a date *n* days before or after the current date, and have the option to count only working days and ignore weekends (for example, if the current day is a Wednesday, adding 5 working days will give the following Wednesday).

Or you can choose a date *n* months before or after the current date, and can select whether the result is the first or last day of the month, or the same day of the month as the current date. You can also use this option to give the first or last day of the *current* month: choose Current date +0 months, and click "First/Last day of month".

Your date selection will result in a code being added to the date field; once you have decided on the date, press Next to select the format you require.

Your pre-defined date and time formats are indicated using, as an example, the date and time you have just selected. You can select one of these and press Finish, or if you want to define a custom date/time field, click Custom Date / Time, and press Next.

If you want to change your pre-defined formats, go to Control Panel and select Regional Settings.

Note that if you want to manipulate dates, integer variables allow the output of numbers as, for example, "Monday", "August", etc. - and in your own language.

2.11.3.1 Insert custom date or time

You can define a custom date or time field here by clicking the appropriate buttons to assemble the format you want. The "code" which generates the format is shown in the text window, and a sample is shown alongside it. Note that you can use ordinals in dates: for example, 1st, 2nd, 3rd, etc.

Remember to select leading or no leading zero as required. Press Finish when the date is as you require.

If the button for AM/PM is blank, this means an AM/PM indicator has not been set on your system. Go to Control Panel and select Regional Settings to set this.

The date code uses standard Windows date/time codes, with the addition of D, for 1st, 2nd, 3rd etc.; by using these codes, KeyText ensures that the date or time is given in your own language. See Field syntax reference for further details.

When the text item is typed or pasted, the date field is updated to give the current date or time.

2.11.3.2 Date codes

If you have selected a date which is not the current date, a code is added to the date field so that it can be updated correctly each time the item containing the field is used. The wizard will generate this code for you, but for reference it is given as follows:

Date[**+***n*[***code***]]
+ can be + or -, indicating forwards or backwards in time. ***n*** is a number from 0 to 99999, indicating the number of days or months to move from the current date.

If code is omitted, days are assumed: forward or back ***n*** days. Otherwise code can be:

d	days, same as if omitted
r	working days only: counts Monday to Friday but ignores weekends
m	months: forward or back n months, using same day of month as current date
mf	months as above, but giving the first day of the resulting month
ml	months as above, but giving the last day of the resulting month

n should only be 0 (zero) if **mf** or **ml** are given as codes, which will give the first or last day of the current month. Otherwise 0 simply indicates the current date and is not required.

2.11.4 Symbols / accented characters

Select the symbol or accented character you require, and click Finish.

2.11.5 Include string variable, clipboard, selection etc

Include Clipboard, Selection, Text File, String Variable or Environment Variable

SMART SELECT The first two options on the wizard dialog allow you to include the contents of the clipboard in an item, or to include any text currently selected in the foreground application. The latter gives KeyText its Smart Select feature.

These **Clipboard** and **Selection** fields will work with selected text - and also with text files copied from or selected in Windows Explorer. They are filled in before the item starts to run, and can be used as part of another field.

Note that in both cases the clipboard is used; the Clipboard field uses the existing text contents of the clipboard, while Selection *copies the selected text to the clipboard first*.

The 3rd option is to **include the contents of a text file**. If the text you want KeyText to use is in a text file, and you want to save disk space by *not* having a copy of it in KeyText, use the Include field to include its path and filename. When the item is typed or pasted, the specified text file will be "expanded" automatically by KeyText. You can have more than one Include field in a text item, if you wish. Use the Browse button to help locate the file you want.

The 4th option is to include the **result of an Ask field or string variable**. Enter the name of the string variable, or Ask field (the text between the first set of quotes - which also goes in the title bar of the Ask dialog).

The 5th option is to include an **Environment Variable**. Environment variables are system variables which running programs can use for configuration purposes; for example SystemRoot, TEMP, USERNAME. If you use the wizard to include one in a script, KeyText will advise you if the chosen variable does not exist. When run, KeyText will include the current value of the selected variable. Note that environment variables in file paths in KeyText’s Run field - for example, %ProgramFiles% - are expanded automatically.

Note that if KeyText finds any valid KeyText fields in the selection, clipboard text or included file, they will be processed as if they were part of a KeyText item - with the exception of further Clipboard, Selection or Include fields.

For each of the above, except for Environment Variable, checkboxes at the bottom give the possibility of manipulating the resulting text.

Select the first check box at the bottom of the dialog to have any trailing carriage returns or line feeds removed. This automatically adds the **X** to the end of Clipboard, Selection, Include or the Ask field, as mentioned above.

The second check box - Encode text for URL - encodes the text found on the clipboard, selection or text file so that it is suitable for inclusion in a URL. The Insert field wizard automatically adds an % to the end of Clipboard, Selection or Include to indicate this option.
An example is in the following use of a "field within a field":

> {Run "http://www.google.com/search?hl=en&q={Selection %}"}
> If the characters:
> "encode URL"
> are selected (including the double quotes) in the foreground application and this macro is run, the following URL will be launched
> http://www.google.com/search?hl=en&q=%22encode%20URL%22
> and the appropriate Google page looked up.

The third check box: do a regular expression replace allows the text on the clipboard, selection etc. to be manipulated using the logic provided by regular expressions. The supported regular expression syntax is given here. In simple terms, parts of the search expression tagged with \(and \) can be substituted into the replacement expression using \1, \2 etc., by order, and further manipulation, such as converting to upper case, can be done.

Bitmaps (Macro action only)
If you want to try it, select "All files" in the browse dialog and select a bitmap instead of a text file. When the item is "typed", the bitmap will be pasted in at the appropriate point in the text, but will not be formatted in any way.

This feature is not fully supported, and is included solely for the convenience of users who may wish to experiment with it.

2.11.6 Ask for user data

If you have a text item which includes repeated variable information, such as a customer name or product code, while the rest of the text is standard, string variables can help - and one of the ways of setting a string variable is the Ask field. Ask fields are processed before the item is run or pasted; you specify the title for the dialog and the question to appear on it, and when it is run the dialog offers a text box for you to fill in the information.

The alternative method of setting a string variable, for example {Product = Input "Product" "Enter product name"}, is displayed when the item reaches it rather than at the beginning, and is not suitable for items which are set to Action:Paste.

The text you enter is filled in *at the location of the Ask field* (unless you select the option “Don’t output result at Ask field”).

The name you give to the dialog is significant, because it acts as the string variable name, and can be used to repeat the information. For example, a KeyText item might look like this:

> Product: {Ask "Product" "Enter product title"}

> Thank you for ordering {Product}. Your order will be dispatched within 3 working days. Meanwhile, if you have any questions regarding {Product}, telephone…

This item could be set to Action: Macro or Action: Paste. In each case, when it is run you would be asked for the product title, and this would be filled in in *three* places: at the actual Ask field, and again on each use of {Product}. The filled-in text would then be typed or pasted automatically.

Note that if the name ends in a @, the result is stored as a global variable and is available to all items; the @ is not displayed in the Ask title.

You can optionally supply a password character which will appear instead of the actual text when the Ask field is run; typically an asterisk * can be specified to appear for each character entered.

You can choose whether hitting the Return key (when the Ask dialog appears) adds a new line to the Ask text, or is similar to clicking OK. The default is new line. You can also select whether the static text displayed on the Ask dialog is left-aligned or centered.

Also, you can choose to have the output URL encoded, so that, for example, a SPACE character types as %20. {Ask %"Search" "Enter search text"} would, if you responded Red Balloon, give Red%20Balloon. Remembering that the name of the Ask can be used to repeat the response, with this example {Search} would output Red Balloon, and {Search %} would output Red%20Balloon.

2.11.7 Link / random link

You can link text items together, which can be useful if several items all have the same text - or your name - at the end. Rather than have the duplicated text stored in several items, store it in one, and link the others to it.

KeyText also has a very similar Call field where control is returned to the item with the call field. With the link field control does not return, and therefore note that any text *after* a link field will be ignored by KeyText.

You can also select multiple items to link to, one of which will be chosen and linked to each time the item is used.

You can choose whether the items are chosen sequentially or at random. The first means that on each use the next item in order is used, looping back to the beginning after the last one is used. Random chooses a link at random from those selected. This feature is useful for those who like to include random quotations at the end of their e-mail signatures, etc.

You can specify "LinkSelect", which means that when KeyText reaches that point in typing or pasting an item, it will produce a dialog asking you to choose which item you want to link to. The dialog will contain a "radio" button and the menu text for each of the items you have chosen by making a multiple selection here.

Note that you can link / random link to *any* item, including those set with "Run" action.

2.11.7.1 Link/Call select details

Once you have selected a LinkSelect or CallSelect field and pressed Next, this wizard page lets you specify a title for the dialog and text to go on the dialog. The text appears above the radio buttons which give you the opportunity, when KeyText reaches that point in a text item, to choose what to do next - which item to call or link to.

2.11.8 Run program etc

To specify a program to run, click the appropriate button and press Next.

To start a new file, specify the required extension and click Finish. KeyText will start the application associated with the selected file extension. To find out what filetypes can be specified, run Windows Explorer, select Options from the View menu, and click the "File Types" tab.

To include a link to a page on the Internet, or to start a new e-mail message, click the appropriate button and press Next.

Note that if there is text in your item *following* a run or new field, KeyText tries to be as intelligent as possible in determining when to continue running the item. If it does start continuing too soon, consider adding a pause or a wait immediately after the run field.

2.11.8.1 Run program

This is one of the fields which can be inserted in a text item after selecting Run a program, launch a URL, or start an e-mail message in the Insert field wizard.

If you first want to check if the program is running, enter any part of the title of its window in the top text box; if this is found, the window is brought to the front and the specified .exe etc. is not run. This is useful if you want to avoid having two copies of a program running at the same time; for example, if you want to run calculator - but if it is running already just bring it to the front - enter calculator in the box at the top, and the location of the .exe below. If you want to specify an exact match on the window title, check "Match whole of title".

Windows 2000, XP, Vista and 7 users only: There is a "feature" since Windows 2000 which inhibits a program's ability to bring another window to the front. If you have asked KeyText to check for an existing window first, but in practice you see the desired window's button in the taskbar flashing, with the window remaining in the background, redo the Run field - this time selecting the "Force activate" option. Or edit the Run field manually to change the ^ to a * (asterisk).

Enter the name of the program to run, using the Browse button to locate it, if necessary. You can also specify a document here, or any file which has a file type registered with your Windows system; KeyText will run the associated application, loading the file you specify. To find out what filetypes can be specified, run Windows Explorer, select Options from the View menu, and click the "File Types" tab.

If you want to include command line parameters - startup information which would be included in a Run command line - enter them in the Parameters box (exe only). If this is a long filename, possibly including spaces, you should enclose it in double quotes.

Some applications need to be run with a default directory specified - normally the directory the application is located in. If you run an application using KeyText and it gives a message indicating it cannot find some of its files, you need to specify a default directory; consult the application's documentation for further information.

If you want KeyText to carry out further instructions after the Run field, then consider adding a pause or a wait immediately after the run field.

2.11.8.2 Go to Internet URL

This is one of the fields which can be inserted in a text item after selecting Run a program, launch a URL, or start an e-mail message in the Insert field wizard.

Enter the address of the Web page you want to go to. When the text item is run, your browser will be started (if it is not already running), and will go to the specified web page. Your normal procedure for going online will automatically occur; for example, if you use dial-up networking you may be asked if you want to dial now. In this case you could add {Wait 10.0 "Dial-up Connection"}{Click "Connect"} after the Run field (or whatever title is in your dial-up networking window).

See the tutorial, scheduler, and the wait field for ways in which further actions in your browser can be automated.

2.11.8.3 Start an e-mail message

This is one of the fields which can be inserted in a text item after selecting Run a program, launch a URL, or start an e-mail message in the Insert field wizard.

If your system has a default e-mail program registered, you can set KeyText to start a new e-mail message every time a text item is run. Enter, for example: mailto:someone@somewhere.com. If supported, when the resulting field is started, either with Action: Run or Action: Macro, your e-mail program should launch with a new message started, and the recipient typed in.

2.11.9 Macro control

1) Pause. Use the up/down arrows to select the length in seconds of the pause you require (up to 1000000 seconds). When KeyText gets to this point in your text item, it will enter Pause mode - indicated by an animated icon in the tray.

If a time of 0 is specified, KeyText enters an indefinite pause - it waits for you to either right-click the KeyText icon in the tray to resume, or left-click it to abort. This is the same as a pause invoked manually by right-clicking the KeyText icon in the tray, or pressing the Pause button, while an item is being typed.

If a time is specified, KeyText will wait that length of time, then continue. During that pause, if you right-click the icon in the tray KeyText will go into indefinite pause mode as above - right-click again to resume. Again, a left-click aborts.

Note that pressing the Pause or Esc keys have the same effect as a right or left click on the icon respectively.

It is possible to edit a Pause field so that the number of seconds is a variable, or the result of an Ask field; for example {Pause {Ask "Pause" "Enter seconds to pause for"}}. In these special cases, a response or variable value of 0 will indicate that no pause should occur.

2) Wait. Selecting this will enable the Next button on the wizard, taking you to the Wait wizard page. See here for information on Wait fields.

3) If clipboard / selection / window title / exe name / string variable (or Ask result) = or contains. Selecting this will enable the Next button on the wizard, taking you to the appropriate wizard page. See here for more information.

4) If pixel color. Selecting this will enable the Next button on the wizard, taking you to the Pixel color wizard page. See here for more information.

5) Change macro speed. If it is necessary to vary the speed at which KeyText types, use this option. This may be useful, for example, where older systems benefit from a slower typing speed, and the whole - or part - of a particular text item should be typed at a speed different from the default.

The default speed is that set in the Options dialog. The Speed field here gives settings from 1 to 10, where 1 is slowest and 10 fastest.

6) Restore default speed. Select this option to have KeyText return to default speed (inserts a {Speed} or {Speed 0} field).

7) Begin a loop to be done *n* times. KeyText can repeat all or part of a KeyText item as many times as you want, up to 99999999! Specify the number of times to run the text or fields which follow; if 0 is given it will loop continually until you press Esc or left-click the KeyText icon in the tray. Use "End a loop" to mark the end of the section to loop, although it is not necessary if the loop finishes at the end of an item.

Note that loops may be "nested" up to 5 deep, but you must make sure that each Loop… EndLoop pair is enclosed within another Loop… EndLoop pair. For example,
{Loop 2}abc{Loop 3}123{EndLoop}def{EndLoop}
will produce: abc123123123defabc123123123def

It is possible to edit a Loop field so that the number of times is a variable, or the result of an Ask field; for example {Loop {Ask "Loop" "Enter number of times to loop"}}. In this special case, a response or variable value of 0 will indicate that no loop should occur - and KeyText will jump to after {Endloop} - or abort the item if that is not found.

8) End a loop. Marks the end of a loop. Note that if Endloop is missing, KeyText assumes that the end of the item is the end of the loop.

9) Call another text item is similar to Link, except that after the called item is finished control is returned to the item which had the call field - which then continues. If you select this, click Next to get to the Call wizard.

10) Goto label. A label is specified as follows:
{:myLabel}
{Goto myLabel} will jump to it - and can be set up with this wizard option. Goto can be useful in conjunction with "if" logic, but care should be taken with jumping out of, or into, a loop or an "if then/ else/endif" section.

11) Cancel current item can be useful when if or goto logic is used.

12) Ignore returns/tabs/blanks on/off. For returns (hidden return characters at the end of lines) this allows you to override the "Return characters in Action:Macro items" setting in global settings. Ignore blanks will skip Returns, tabs and SPACE characters. If On is chosen, all Return, Tab or blank characters, from that point on in the macro, are ignored and not typed. If Off is chosen, all Return, Tab or blank characters, from that point on in the macro, are processed - and typed.

13) Exit from KeyText. Insert this in an item, and when this point is reached in "typing" it, KeyText will exit.

2.11.9.1 Wait

The default behavior ("Found" and "Cancel" radio buttons selected) is: when KeyText gets to Wait field in a text item it pauses and waits for a **window** whose title includes the specified text, to appear in the foreground. In this case the number of seconds specified (up to 1000000) indicate how long to wait before aborting.

If a time of 0 is specified, KeyText will continue waiting until the window appears, or you press Esc or left-click the icon. As with Pause, pressing Pause or f12, or right-clicking the icon, will force KeyText to continue, even if the window has not been found. A very short time, say 0.1 second, can be used if you simply want to check if the window exists.

It is not necessary to give the whole title of the window; enough to distinguish it from other windows is all that is required. If, however, you wish to match only the exact title then select the "Match whole title" checkbox (note that upper/lower case is not significant).

If necessary, you can specify additional static/fixed text in the wizard, so that KeyText can better identify the correct window. For example, if a waited for window has the same title as its parent window, additional identifying text may help to distinguish it. Note that this should be static text only; that is, not text on a web page or an edit window.

To monitor the foreground window for the required title, select "Check foreground window". If, however, you wish to monitor all windows, select "Check all windows". The latter option may be useful if you want to establish if the specified window exists already, rather than waiting for it to appear. **Note that if the window is found and is not in the foreground, KeyText will not bring it to the front automatically (use an Activate field following the Wait if you want that).**

A Wait field can wait for specified **Status bar text**. Note that this will only work if the foreground application uses the Microsoft common control "msctls_statusbar32". It can, for example, be used with Internet Explorer 6 (or earlier) to wait for "Done" in the status bar - but the same will not work with Mozilla Firefox or Internet Explorer 7, which use their own status bar control.

KeyText can also wait for a specified **file** to exist. Check the radio button "file whose name and path is" to use this feature. Wildcards * and ? can be used. When the wait begins it checks for a file matching the given path and name, and again every 15 seconds (more frequently if the wait time is less than 600 seconds), and continues the item when the file is found. This may be useful in a network environment where you want KeyText to wait until another process is completed before continuing; if the other process creates a dummy file to signal its completion, KeyText can use that to trigger the rest of its macro. If the time specified is 0, KeyText will keep waiting until the file is created, or the script is canceled by pressing Esc or left-clicking the KeyText icon.

The logic of the Wait field may be changed from the above by use of the appropriate radio buttons. You can specify whether the item should continue when or if the specified window / status bar text / file is found, or if it is *not* found. And you can have four alternative actions occur if the opposite is the case. This gives the following eight possibilities:
If found continue, else cancel (default)
If found continue, else start another item
If found continue, else call another item (then continue)
If found continue, else go to a label (for example {:goHere})
If not found continue, else cancel
If not found continue, else start another item
If not found continue, else call another item (then continue)
If not found continue, else go to a label (for example {:goHere})

See the wait field syntax for how the Wait field will appear once constructed by the wizard, or see If/then/else based on existence of a window in the tutorial for an example of how to use Wait.

Note that it is possible to edit a Wait field so that the number of seconds is a variable, or the result of an Ask field; for example {Wait {Ask "Wait" "Enter seconds to wait for"} "Window title"}. In these special cases, a response or variable value of 0 will indicate that no wait should occur and that the check for the window title should be immediate.

2.11.9.2 If clipboard / selection / window title / exe name / string variable (Ask result) = or contains

If clipboard / selection / window title / exe name / string variable (Ask result) = or contains

KeyText can examine text on the clipboard, any text selected in the foreground window, the foreground window selection or title, the launch exe (process name) of the foreground window, or a string variable (Ask response). You can check if the text equals or contains your specified text, or is empty (except for exe name), or is like a specified regular expression - and specify an action to take if there is - or is not - a match. *Note that the feature to examine the current text selection uses the clipboard*.

When examining Clipboard or selection text you can specify "Ignore trailing Returns on clipboard", which can be useful when examining the text contents of a spreadsheet cell on the clipboard - which often includes a trailing Return.

Two of these fields can be used in an Action:Paste field - and only at the beginning:
If exe equals or contains the specified text, or if title equals or contains the specified text - cancel. This is so that you can exclude an action:paste item with Trigger text from pasting in certain applications. For example, if an item (macro or paste) begins with {If exe contains "Word" cancel} then that item will abort if the foreground application is Word.

See "If" fields for more information, including details of the actions which can be taken when true.

2.11.9.3 If pixel color

KeyText provides a "dropper" on this wizard page which you can drag to the pixel you want.

This allows you to set the co-ordinates to test, and the required color - the dropper detects the color of each pixel as it passes over it. For convenience you can select "Hide wizard while dragging dropper".

Alternatively you can use the Color chooser to pick the required color.

The color is specified as 3 numbers between 0 and 255, for red, green and blue. This ranges from 0 0 0 which is black to 255 255 255 which is white.

If tolerance is 0, then an exact match on these values is required; specify a tolerance of up to 99 to allow different shades, or similar colors, to match.

You can choose to work with either screen co-ordinates (top left is 0,0) or co-ordinates relative to the window under the target when it is released. The "relative to window" option gives ability to detect the color of a pixel in a window - even if it appears on different positions on the screen. As long as the same window (the one over which you released the target) is at the front, and the point to check is consistently placed on the window.

See "If" fields for more information, including details of the actions which can be taken when true.

2.11.9.4 Call another text item

The Call field is very similar to the Link field. The difference is that whereas Link passes control to another item without returning, Call does return, and text or fields after the Call field continue. See Link / random link to another text item for more information on the Link field.

Note that while the Call field can only be used in an item set to Action: Macro, it *can* call an item set to Action: Paste. The text is pasted, and control returned. Also note that Calls can only go one level deep; the item *called* cannot Call another item - any Call field will be ignored.

You can also select multiple items to call, one of which will be chosen and called each time the item is used.

You can choose whether the items are chosen sequentially or at random. The first means that on each use the next item in order is used, going back to the beginning after the last one is used. Random chooses an item at random from those given.

You can specify "CallSelect", which means that when KeyText reaches that point in typing or pasting an item, it will produce a dialog asking you to choose which item you want to call. The dialog will contain a "radio" button and the menu text for each of the items you have chosen by making a multiple selection here.

2.11.10 Activate window, click button or choose menu item

There are four possibilities here. In the first three, note that case is not significant, and that the full text is required in 2 or 3; *part* of the text required can be given only in the case of 1, activating an existing window. Underlined characters in 2 and 3 (caused by the inclusion of an & character) are ignored.

1) Activate (bring to the front) an existing window. You can specify part or all of the window title you want to bring to the front. Case is ignored. KeyText will bring to the front the first window it finds which matches the text specified. For example, specifying Notepad would bring a window with the title "Untitled - Notepad" to the front. If you check "Match whole of title", then an exact match is required, and specifying Notepad would *not* bring "Untitled - Notepad" to the front.

The default is to find window 1 with the title - that is, the matching window nearest the front. If you want a window with the same title which is underneath that one to be activated, select: find window 2 with title.

If the window is not found, the KeyText item will stop. However, you may wish to run the appropriate program (or any program) instead. If so, check the option "If not found, run a program instead". The "Finish" button will then change to "Next" - click it to go to the Run dialog where you can specify the program to run. This is useful if you want to avoid having two copies of a program running at the same time; for example, if you want to activate calculator - or run it if it is not found - enter calculator in the activate box, check the "If not found..." option, and choose "Next" to specify the location of the .exe. Note that you can also do this by going directly to the Run program dialog.

Windows 2000/XP/Vista/7 users only: There is a "feature" since Windows 2000 which inhibits a program's ability to activate (bring to the foreground) another window. In many cases KeyText's Activate field works correctly, but if you see the desired window's button in the taskbar flashing, with the window remaining in the background, redo the Activate field - this time selecting the "Force activate" option at the bottom of the dialog. Or edit the Activate field manually to make it {Activate! "…" }.

2) Click a button. Here you must specify the full text of the button; don't worry about any underlined characters - they are ignored, and case is also ignored. KeyText will not always be able to find the button; some applications "paint" the text onto the surface of the button, rather than send the text to the button directly. If the text is "painted" on, KeyText will not find it. You could try observing how many Tabs it takes to bring the focus to the required button, and putting that number of {Tab} fields in the KeyText item, followed by {Return}.

Note that if the button appears in a consistent position, either in relation to the foreground window or the screen, a Mouse action may be used instead of Click.

3) Select a **menu** item. In the box, specify the menu item you want to select, separating menus and submenus from menu items using the | character. For example, to select the Options... item from a Tools menu, enter Tools|Options... or to select the Clipart... item from the Picture submenu of an Insert menu enter Insert|Picture|Clipart... . KeyText will not always be able to find the menu item; some applications "paint" the text onto the surface of the menu, rather than send the text to the menu directly. KeyText itself works like this, to produce its graphic menus! If the text is "painted" on, KeyText will not find it. You could try {Alt T}O in the first example above, or {Alt I}PC in the second case, assuming that the letters are the ones underlined.

2.11.11 Shift/key combination

This is one of the fields which can be inserted in a text item using the Insert field wizard. You can also access this wizard page directly from the "Item" menu of the Edit text items window.

Note that this works differently from the KeyText Keypad; it is used to place a field in your text specifying one combination of keys to be pressed at that point. Note that normally you would simply type or paste text directly into the Edit text items window; this wizard is normally used for shift combinations, or for keys such as "Page Down", numeric keypad 3, etc.

You can specify the number of times the key or combination of keys should be pressed; the default is 1 time.

Once you have "pressed" the key combination you want, you can either click "Finish", and the field will be added to your macro - or select "Add" to add it to the text box to the right of the Add button. The second method means you can add several key combinations, which will all be added to your macro on clicking "Finish".

The Clear button will clear any selected keys.

The section Field Syntax reference contains a list of how the key combinations will appear in your KeyText item. You can type them directly into your item if you do not wish to use the wizard.

Note that there is a key with the Windows logo, and an Application key, as seen on the Microsoft Natural Keyboard, and many other keyboards; KeyText can simulate these keys even if your keyboard does not have them. The Windows logo key can be used as a shift key; for example, with the letter R it would normally open the Windows Run dialog. Or it can be selected on its own, in which case its effect in a KeyText item is to open the Start menu; {Ctrl Esc} can be used with the same result. The Application key, which looks like a tiny context menu, is the one which provides quick and easy access to pop-up menus and help topics.

New in KeyText 3 is the ability to "hold down" and "release" a key. Note that this cannot be specified using the wizard, but can be edited manually afterwards. For example, {Shift+}abc{Shift-}def would type ABCdef. Note that key repetition does not occur if a key is "held down" by KeyText.

The use of key combinations opens up lots of possibilities, depending on the application in the foreground at the time. For example, if your text item includes words you want to have in **bold** type - and your word processor supports this type style, press Ctrl and B, then select Finish. The field {Ctrl B} will be inserted in your text, and when typed by KeyText, will toggle bold on or off.

Shift/key combinations can be used for navigation, menu selection, button pushing, etc. If you want a KeyText text item to carry out a certain sequence of keystrokes, observe what keystrokes are needed manually, then "program" them into the KeyText item.

For example, if a dialog requires you to enter some information in two different boxes, you may be able to do this simply by specifying the first text, then inserting a {Tab} using the Insert Field Wizard, then the second text followed by {Return}. In fact, you could just press [Tab] and [Return] in the text item to include the correct key presses, but it is easier to understand what is happening if the keys are specified in full.

Note that you can also specify button presses, menu selection, etc. using the Activate a window, click a button or choose a menu item dialog.

It is recommended that you do not use the {Alt Tab} etc. combinations in KeyText. It will work to change the active window to the next one, but specifying two together - because in effect the Alt key is released in between - will bring you back to the application you started with! It is recommended that you use an Activate field instead, to specify the title of the window you want to switch to.

Also note that when Ctrl-Alt Delete (the combination used to shut down your computer or, in Windows NT, to log on) is pressed on your keyboard, a special predefined signal is sent to Windows - which *cannot* be simulated by KeyText.

2.11.12 Play a sound

Locate any .wav file using the Browse button, and KeyText will play it when it gets to that point in typing a text item. A sound file "chime.wav" is included with KeyText for you to use if you wish.

Or select Beep to play the system default sound.

Playing a sound can be useful to accompany a display a message field - specify the sound field immediately before the message one. This combination can give an effective reminder or alarm function.

You could even select a suitable .wav and have it played hourly as a chime, using the KeyText scheduler; see also the KeyText tutorial.

2.11.13 Display a message / User input

When run, the **Input** field asks you for data *when it reaches that point in the item*. In this respect it differs from the Ask field, which asks you for data in the processing stage before the item is run. When you press OK, the data you have entered into an Input field is then typed. If you press Cancel, the KeyText item will abort.

You can optionally supply a password character which will appear instead of the actual text when the Input field is run; typically an asterisk * can be specified to appear for each character entered.

Alternatively, your response to an Input field can be stored as an integer or string variable for use later in the item - or, if it is a global string variable (has a @ at the end of the name), it can be used by any item.

The **Message** field displays a message as a text item runs. It could in fact be the only element of a text item and be used in conjunction with the scheduler to give a reminder or alarm function. Place a sound field before it for added impact.

If a Message field is scheduled, you will probably want to avoid having any text in the item after the message field. For example, you may have set a reminder for 2pm daily; at that time the message box is displayed and you click OK to dismiss the message. Then *any text or fields following the message will run*. It is uncertain what application you will have in the foreground to receive these keystrokes, etc.

Because of this possibility, the check box on this window is checked to say, "Delete any item text following the message field".

Uncheck this box if you have text, etc. in the item which you do want to have following the message - for example, a message in the middle of a text item indicating that it has reached a certain point and needs to know whether to continue.

The message will be displayed with an OK button. If the message is not the last part of a text item, the message will also have a Cancel button; selecting OK will dismiss the message and continue the item; Cancel will dismiss the message and abort the item. You can select whether the message displayed on the dialog is left-aligned or centered.

If the item with the message field is scheduled for a certain day or time, it will also display an "Edit/ Repeat..." button. Press this, and you can choose to either repeat the item after a specified number of minutes (1-240), or edit the text or schedule for the item.

While the input request or message box is displayed, KeyText enters pause mode, as indicated by the animated icon in the tray. Other KeyText functions - including scheduled items - are disabled until the message is dismissed. Note that this means only one message can ever be displayed at a time (by one instance of KeyText). If a scheduled event occurs while a message is displayed, it will not be actioned until that message is dismissed.

There is an option to automatically close the input or message box after a certain number of seconds; use this if, for example, you want to display a prompt which will automatically disappear after displaying for, say, 30 seconds. You can choose how the box closes. "Close and continue item" is the equivalent of clicking OK, and after the specified number of seconds the item will type any entered text if it is an input box, and continue if it is not at the end. "Close and abort" is the equivalent of clicking Cancel, and after the specified number of seconds any entered text in an input box will *not* be typed, and the item will abort, even if it is not at the end.

When Input is asked for you can optionally supply a password character which will appear instead of the actual text when the Input field is run; typically an asterisk * can be specified to appear for each character entered. You can also choose whether hitting the Return key (when the Input dialog appears) adds a new line to the Input text, or is similar to clicking OK. The default is new line.

2.11.14 Mouse actions

The mouse action fields allow you to specify a left click, double-click or drag action, a right click or drag, or a middle button click on the mouse. There is also the ability to move the pointer without a click, and the ability to restore the pointer to its position before the item was started.

KeyText makes it easy to set the co-ordinates for the action. Move the mouse over the target ⌖ on the wizard, hold down the left button, drag to where you want the mouse action to happen, and release. You will see the co-ordinates change as you move the mouse on the screen. For convenience you can select "Hide wizard while dragging target".

If you are specifying a drag (drag and drop / drag and release) action, then a second target is displayed allowing you to choose where the mouse button is released.

You can choose to work with either screen co-ordinates (top left is 0,0) or co-ordinates relative to the window under the target when it is released.

This "relative to window" option gives KeyText the unique ability to mouse-click a button even if it appears on different positions on the screen. As long as the same window (the one over which you released the target) is at the front, and the point to click is consistently placed on the window, KeyText will find it!

Relative co-ordinates are useful therefore if the location on screen of the target window is unpredictable; if the point to click is consistent in relation to the target window, then relative co-ordinates should hit the right spot - even if it is, for example, a link on a browser page. On the other hand, if a certain dialog always appears center screen, then screen co-ordinates should be used.

Note that with a drag operation it is permissible to have one set of co-ordinates relative, and the other screen.

As an alternative to specifying co-ordinates, you can select "Don't move mouse pointer". This causes the specified mouse action to take place wherever the pointer is at the time the KeyText item is run. Say, for example, that you frequently save pictures from your browser by right-clicking them and selecting "Save Picture As…". You could use the Mouse wizard to give a KeyText item like:
{Mouse R,-,-}s
and set a hotkey for it. From then on, moving the mouse pointer over a picture and pressing the hotkey will open the right-click menu and press "s" - the underlined letter which will open the save dialog.

You can set KeyText to simulate Shift, Ctrl or Shift and Ctrl, being held down at the same time as the mouse action takes place.

If you have your mouse buttons swapped for left-handed use, you should select the action as if the buttons were not swapped. For example, if you want KeyText to press the conventional right button to open a shortcut menu, specify right-click in KeyText - even though it would be the left button on your left-handed mouse.

2.11.15 If fields

If logic is available based on several different parameters, in three groupings:

Text
KeyText can examine text on the clipboard, selected text, the foreground window title, the launch exe (process name) of the foreground window, or a string variable (or "Ask" field result). You can check if the text equals or contains your specified text, matches a regular expression, or is empty (except for exe name) - and specify an action to take if there is - or is not - a match. The If clipboard text (etc) wizard helps you to set it up. See also If field syntax.

Pixel color
KeyText can examine the color of the pixel at a specified co-ordinate to check if it matches the color you have specified (within the tolerance you have also specified) and specify an action to take if there is - or is not - a match. The If pixel color wizard helps you to set it up. See also If field (pixel) syntax.

Integer variables
KeyText has local and global integer variables available for "if" logic. "If" fields based on variables must be entered manually; the Variables page will help you to set it up. See also If field (variables) syntax.

If actions
There are 5 possible if actions, which will be done if the result of your "if" question is true.

Note that if the result is false then the macro will continue immediately after the "If" field except when the action is continue - when it will jump to the next {Else if...}, {Else} or {Endif}.

Continue
This will result in an "If" field which ends with "then". An example will best illustrate how this works:

```
{If clipboard contains "green" then}
...do green functionality...
{Else if clipboard contains "blue" then}
...do blue functionality...
{Else}
...do something else...
{Endif}
and continue here.
```

The else and endif fields must be entered manually. For an {Else if...} field you can use the Insert field wizard to enter the {If...} field in the normal way - then type in the Else manually. Note that if these are not specified, and the "If" result is false, KeyText will jump to the end of the item - as if canceled.

Cancel
The item is canceled.

Start another item
Starts - links to - another item, which when using the wizards can be selected on clicking Next>.

Call another item
Calls another item, which when using the wizards can be selected on clicking Next>. After the called item has run, control returns to immediately after the "If" field.

Goto label
The item jumps to the specified label, which must be present in the item - for example:

{If exe not contains "Notepad" goto notNotepad}...do Notepad specific stuff ...{:notNotepad} and continue.

See also If field syntax.

2.11.16 Integer variables

Each item (with Action Macro or Paste) can make use of up-to 26 integer variables a, b, c... through to z, which can hold values in the range -999,999,999 to 999,999,999. Their value defaults to zero when an item starts, and is preserved in any item which is called or linked to by that item (that is, the scope is the current item and any items it calls or links to). They can also be associated with global variables, preserving their value from one item or session to another.

See below for a quick description of how to use these variables; see also Variables syntax for details of the syntax used, and also If field (variables) syntax.

The # character is used to denote a field which makes use of variables.

Assigning integer variables
They can be assigned values as in these examples:

```
{#a=1}
{#b=a}      (b will also equal 1)
```

{#c=Input "Input dialog title" "Enter value for c"}
There are four assignments that set variables to the current mouse co-ordinates. The following two set the variables to the screen co-ordinates of the mouse pointer:
{#x=MouseX}
{#y=MouseY}
While the following two set the variables to the current co-ordinates *relative to the window under the mouse pointer*, where 0,0 is the top left of the window area just under the title bar.
{#x=MouseX#}
{#y=MouseY#}
There is also a special assignment to assist in date manipulation:
{#d=DaysInMonth 2,2016} would set variable d to 29. Or use {#d=DaysInMonth m,y} if variables m and y are the month and year.
Note that the Input method above can be set up in the Insert Field Wizard display a message or ask for user input section.

Arithmetic
Arithmetic (integer only) can be performed as in these examples:
{#a+=1}
{#a*=b}
The following are available: +=, -=, *=, /= and %= (mod or remainder)
In all cases the result of the calculation is assigned to the left-hand operand. So if x currently is 3 and y is 4, after {#x*=y} x will equal 12.

Outputting integer variables
To output the variable - either "typed" or pasted in your text, or as part of another field - use:
{#a}
Or, for example, {Loop {#x}} to start a loop to be performed x times.
Width and leading zero can also be specified. For example:
{#y "10"} will output y with leading spaces up to a width of 10.
{#y "010"} will output y with leading zeros up to a width of 10.

Special formats are available to convert integers into date information. For example, if variable m is 2:
{#m "MMM"} will give Feb and
{#m "MMMM"} will give February
(or the month name in your own language). Similarly "ddd" and "dddd" will give the weekday name associated with the number, where 0 is Sunday, and D will give 1st, 2nd, 3rd etc. "tt" will give AM if the variable is less than 12, else PM. See date codes for more information.

Global integer variables
Local variables a-z can be associated with global variables whose values are stored and can be accessed by any item from any KeyText data (.ktt) file.
For example:
{#n@"Counter"}
The first time this is used, a global variable with the name Counter will be set up, with the current value of n.
Once the global variable exists, the above field would set n to the value stored in Counter.
When a macro with an association to a global variable ends, the global variable *is updated with the current value of the local variable*.
For example, say you want a macro to generate a filename which includes a counter which increments on each use.
{#a@"FileCount"}data{#a "04"}.txt{#a+=1}
On first use this would type data0000.txt. The next time it is used it would type data0001.txt, and so on.

Managing global integer variables

You can manage global variables through the Manage variables option in the KeyText Organizer. This allows you to view - and change - the current value, and also optionally set a reset value and specify that the variable should be reset to this:
When KeyText starts
Or every day at a given time
Or every Monday (etc)
Or on a set day each month.
If, in the above example, you wanted a filename which showed the month, and then a counter which was reset to 1 on the 1st day of each month, you could use:
{#a@"MonthCount"}{Date "yyMM"}{#a "04"}.txt{#a+=1}
and set MonthCount to reset to 1 on the 1st. In April 2006 this would generate 06040001.txt, 06040002.txt and so on.

Date / time / random integer variables

A variable can be associated with part of a date or time.

For example:
{#s@second}
would associate variable s with the current seconds (system clock). Each time {#s} is used it would output the current seconds from the time.

Options are: year, month, dayofweek, dayofyear, day, hour12, hour, minute, second.

Note that if, further down the item, you assigned a new value to that variable, for example {#s=10}, it would break the association with seconds - and s would hold the value 10.

A variable can be set to produce random numbers. For example:
{#r@rand 100}
After this, {#r} would output a random number between 0 and 99 each time it is used. For example:
{#r} {#r} {#r}
might produce 57 23 81.
{#r@rand b} would output random numbers between 0 and b minus 1.

Again, note that if you assign a new value to a random number variable, it breaks the association with random numbers. So, for example,
{#r@rand 10}{#r} {#r} {#r}{#r+=1} - {#r} {#r} {#r}
might produce 2 9 4 - 7 7 7
In other words, the {#r+=1} has generated a new random number for r, added one to it, and turned it into a normal variable.

“If” logic based on variables must be programmed manually

For example:
{if #a=0 goto label}
{if #a>b then}
Available logical operators are:
= not = (or !=) >= <= < >

goto, then, call, link and cancel are available. See If field (variables) syntax.

2.11.17 String variables

KeyText allows the use of string variables which, in conjunction with regular expressions, gives powerful string manipulation abilities. See also String Variables in the field syntax section.

String variable names can be a combination of letters, numbers, space and other characters, but *not* ' " = %or {} (or whichever field delimiters you are using). A minimum of 2 characters is required (and leading/trailing spaces are ignored). They should not start with a KeyText reserved words such as Pause, Loop, Shift, Input etc.

The @ symbol can only be used at the end of the variable name, and denotes that the string variable is global. That is - it can be used by any item, and from any instance of KeyText. Global variables are stored in a special encrypted file called KeyText.gsv, and you can move it to any location you like - use the File location button on the Manage variables dialog to set it. [Note that although encrypted, its contents are not password protected, and could be accessed by anyone running KeyText.]

A password can be set to prevent the global variables from being changed. The intention of this is to allow network users to share a single gsv file, while restricting the ability to change it. For example, say a company has a featured product of the day, which is set every morning. A user with password access could set a global variable ProductOfTheDay@ to the featured product name, set it to prevent changes, and then every user could have access to it - and it could be referenced in a number of different items.

Note that the scope of a *local* (as opposed to global) string variable is the current item, *and* any items directly called or linked-to by that item.

The string contents assigned to a string variable can be up to 2048 characters long.

Assigning variables
They can be assigned values as in these examples:
{Full name = "John Smith"}
{Featured product@ = "KeyText"}
{Color required = Input "Color" "Please enter required color"}
{Ask -"Sex" "Enter M of F"} (in this case the string variable name is Sex)
{Ask -"Phone number@" "Enter phone number"} (this sets a global string variable, but note that the @ will not appear in the Ask dialog title bar)

The Input method above can be set up in the Insert Field Wizard display a message or ask for user input section.

Note the difference between Ask and Input. When an item is started, KeyText scans for Ask questions - and asks them at the start. With Input the question is asked when it is reached in the item. Only the Ask field can be used in an item set to Action:Paste.

Note that field nesting can give interesting possibilities. For example:
{String two = "{String One}"} will make String two have the same content as String one.
{Start clipboard = "{Clipboard}"} will give a variable "Start clipboard" whose contents will be the text contents of the clipboard.
{Clip1 = "{Clipboard "\(.\).*" "\L\1"}"} will give a variable "Clip1" whose contents will be the first character on the clipboard, in lower case.
The following example associates integer variable w with the day of the week (0-6), and by "outputting" that into quotes, the string variable Weekday will contain Sunday, Monday etc. (in your own language).
{#w@dayofweek}{Weekday = "{#w "dddd"}"}

Outputting string variables

Note that the Insert Field Wizard helps you to set up the output of a string variable; choose the option "Include string variable, clipboard, selection etc".
To output the variable - either "typed" or pasted in your text, or as part of another field - wrap it in field delimiters.
Using the above examples:
{Full name} would output John Smith.
{Featured product@} would output KeyText.
{Featured product@ ".*\(.\{4\}\)" "\1"} would output Text (last 4 characters of KeyText).
{Weekday ".*" "\U&"} would output SUNDAY, MONDAY etc, depending on the actual day.

This last two examples show the use of regular expressions. The output could also be URL encoded:
{Full name %} would output John%20Smith.

Global string variables

By adding the @ to the end of the string variable name, its value is stored for future use by other items - or other KeyText data files. If it is changed by an item (or by Manage Variables), its new value will be used by all items referencing it.

Managing global string variables

You can manage global string variables through the Manage variables button in the KeyText Organizer, and selecting the Strings option. This allows you to add or delete variables, to edit their contents, to determine where they are stored and if they can be changed.

2.12 KeyText Settings

2.12.1 Settings dialog

This dialog has 5 tabs.

The first two are specific to the current .kt3 file (KeyText file where your text is stored):
Options
Password

The other three are global - settings which apply whichever .kt3 file is being used:
Global
Customize
User Information

2.12.2 Options

To go to Options, right-click the KeyText icon and select Settings; the Options dialog is the first tab.

The settings in Options apply only to the current .kt3 file - noted in the title bar of the dialog. This means that you can have different Options settings for different .kt3 files.

1) Choose the KeyText icon you prefer. This option is particularly useful if more than one KeyText is running at a time (see multiple instances) so that the different instances are easily distinguished. The option "None" is only available if a keyboard shortcut has been set (see below). Use this option with care: KeyText will disappear from your desktop, but will still be running with hotkeys active. See using KeyText with no icon in the tray for further information.

2) A keyboard shortcut (hotkey) to open the left-click menu can be switched on and selected here. To

select the shortcut, click the Hotkey box and then press the combination of keys you require. Note that if your keyboard has a Windows logo key it can be included in a hotkey combination (although note that a number of Windows logo key combinations are reserved for Windows use, and you may not be able to set them). Also note that if you choose a combination which is not allowed (e.g., Shift-A), Ctrl will be automatically added. If KeyText finds that the Hotkey is in use by another application, you will be asked to select a different one. You should avoid Hotkeys that may conflict with important functions of your word-processor etc.; combinations such as Ctrl-C (normally used for copying selected text to the clipboard) are not recommended! See also how to get text out.
Note that you can choose in global settings to keep this hotkey active even when another .kt3 file is loaded; this provides an easy way of switching from one set of text items to another.
See Hotkeys for more information.

3) Default action. Choose what action to take if a text item is selected which has Action: default specified. It can be Macro or Paste; the third Action: "Run" can only be selected in the edit window.

4) You can specify a text item which will be started automatically whenever the current .kt3 file is opened. Set this to "None" if you do not require an auto-start item.

5) You can set the first item to appear on the "right-click anywhere" menu. Your form-filling texts could therefore be set to start at, for example, "Name", so that your name would be at the top followed by items A, B and so on, or 4A, and that item would appear top of the list in the new menu. Note that the right-click anywhere menu does not show empty items or items with a schedule.

6) The speed at which an item set to Action: Macro is run. Note that you can vary the speed within a text item by using the speed field.

7) Switch "Auto-add clipboard text" on and off, and choose the item which will receive the accumulated texts (every time text is copied to the clipboard).

The last three settings relate to how text is added to an item using Auto-add, or the left-click menu popups: "Copy selected text to>", and "Add selected text to beginning (end) of>".

8) Choose whether added text goes to the beginning or end of the chosen item.

9) Specify character(s) to be used to separate the pieces of text added.

10) If Auto-add is on when you exit KeyText, if "Keep Auto-add on for next session" is checked, it is switched on again the next time you use the current .kt3 file.

2.12.3 Password

To go to Password, right-click the KeyText icon and select Settings; the Password dialog is the second tab.

The settings in Password apply only to the current .kt3 file - noted in the title bar of the dialog. This means that you can have different Password settings for different .kt3 files.

You can set up a password to protect any personal information stored in KeyText. **Please note that such information is stored entirely at your own risk, and MJMSoft Design cannot be held responsible for any loss or damage should such information be "hacked" or disclosed in any way.**

Click Set Password to define a password for the current .kt3 file or, if one has already been defined, click Change Password to change it.

Five levels of password security are offered; click the checkbox to enable them, then select:

1) Once in each session of KeyText Within one KeyText session, the password will only be asked for the first time the .kt3 file is opened. If you switch to another .kt3 file and then switch back, the password will not be asked for again. Note that if you exit KeyText and run it again the password will be required. Note also that if the .kt3 file is shared on a network and another computer alters it, the password will be required again.

2) Every time it is opened Requires the password every time the .kt3 file is opened.

3) Every time it is opened and after 1 hour of non-use additionally requests the password if the .kt3 file items have not been used or changed for one hour.

4) On selecting Edit text items, Settings, or Organizer This level allows you to use the items in the current .kt3 file, but not to view them or change settings. Note that this is the only password level which does not add the "Password lock now" option to the right-click menu. It is therefore ideal where you want someone else to be able to use the KeyText data (including the running of scheduled items), but not view or alter it.

5) Every time it is opened and on selecting Edit text items, Settings, or Organizer This is the same as level 4, except that the password is also required on opening the .kt3 file, and "Password lock now" does appear on the right-click menu. This means that the user will require knowledge of the password to use KeyText, which is not the case with level 4.

Password levels 4 and 5 can be used with the check box below them: "Hide scheduled items in left-click menu" (this check box also appears in the Customize dialog with global effect, unless the setting on this dialog overrides it). If this is selected, the scheduled item cannot be accessed from the left-click menu, but will continue to operate if the scheduled time arrives or Window appears - or if a hotkey has been set for the item and is pressed. If this level and option is selected, it offers added security for scheduled items such as password automations, without the inconvenience of level 6 below which requires a password every time KeyText is used.

6) Every time it is used requests the password on each use. This highest level of security is not generally recommended, unless you use KeyText to store a number of passwords - you then only need to remember the KeyText password to access them.

2.12.4 Global

To go to Global Settings, right-click the KeyText icon and select Settings; the Global dialog is the third tab.

The settings in Global apply "globally", unlike Options and Password, which apply only to the current .kt3 file.

1) Startup. To have KeyText start automatically every time Windows starts, check this option. Uncheck it to remove KeyText from the Startup list. Note that the startup setting applies to the current user. See What happens when KeyText starts for more information.

2) Welcome box; check this option if you do not wish the welcome screen to appear when you start KeyText.

The next option is only available in the full (unlocked) version of KeyText.

3) Whether to allow multiple instances of KeyText to run at the same time. If you use multiple instances, consider using the extra KeyTexts option. If this is set at, for example, 2, every time you run KeyText, it will open with the .kt3 file you used last, and start two additional instances with the previous 2 .kt3 files. If this is set, you can override it by holding down the SHIFT key when KeyText starts (doing this also suspends the scheduler).

4) Note that you can select a hotkey in options to open the left-click menu. In Global settings you can choose to keep such hotkeys active even when another .kt3 file is loaded; this provides an easy way of switching from one set of text items to another.

5) The .kt3 file saves all changes automatically, and keeps the previous version with the file extension . bak. You can switch off this backup option here - no further .bak files will be generated.

6) Typically, text items stored in KeyText are relatively small, and a limit is set to stop them becoming too big - a problem if auto-add is on and forgotten about. You can increase or change the maximum size of an item here.

7) Fields within text are normally delimited by braces {...}. If these are characters you use frequently, you can change the delimiters to [...] or <...>. Note that if you change this setting it applies to all .kt3 files, and you may need to edit them if they contain fields made using the previous setting.

8) Method to type accented characters in Action:Macro items. Two methods are available to type accented characters and symbols. The first simulates Alt 0nnn on the numeric keypad; this Windows standard should work in most applications. The second is for KeyText to "ask" Windows what key combination is available to type the character; if Windows specifies one, that combination is simulated (otherwise the Alt 0nnn method is used).
The first method is now recommended, mainly because of scenarios like this: KeyText asks Windows how to type é, and (depending on language settings) Windows may respond Ctrl-Alt e. However, if this combination is used in Microsoft Word, it produces a € character (depending on language settings). Because of this inconsistency, Alt 0nnn is recommended.

9) Return characters in Action: Macro items (that is, the hidden returns which have been typed to move to a new line). The first option is "Type all return characters", where, under Action: Macro, all [Return]s are sent as if typed at the keyboard.
The second option is "Ignore all return characters". In this case use {Return} or {Enter} to specify all new lines or returns. Or {Return 2} for two returns, and so on.
The third, and default, setting is "Ignore return which follows certain fields"; a [Return] character following immediately after the closing brace } of certain fields is ignored when "typed". This helps to lay out fields better. The fields concerned are: Activate, Ask (but only if no output selected), Beep, Call, Click, Endloop, If/Else/Endif, IgnoreTabs, Input, Loop, Menu, Message, Mouse, New, Pause, Run, Speed, Wait, and Wave. If you do want a [Return] typed after these fields, add a second one (only the first is ignored) or use {Return}.

10) If Caps Lock is on when an Action: Macro item started, it can be turned off - otherwise case would be reversed when the characters were typed. The default is to turn it back on after an item is finished, if it was on before. This setting can be changed here with the options, if Caps Lock is on, to leave it on, turn it off, or turn it off and switch it back on after the item is finished.

11) The "right-click anywhere" setting can be set here, having been initially set on installation. If on, the options are to have a right mouse-click, with Shift, Ctrl, or Shift and Ctrl held down, open the "right-click anywhere" menu. If on, it can be temporarily suspended by selecting "Suspend right-click anywhere" on the right-click menu. Note that if you have more than one KeyText running at the same time, only the first one can be set to display the "right-click anywhere" menu; although if you suspend the setting, it can then be turned on in another instance of KeyText.

2.12.5 Customize

To go to Customize Settings, right-click the KeyText icon and select Settings; the Customize dialog is the fourth tab.

The settings in Customize apply "globally", unlike Options and Password, which apply only to the current .kt3 file.

1) You can choose to switch syntax highlighting in the Edit text items window on or off.

2) To make it easier to switch between editing and testing an item, a hotkey can be set which will have the following effect:
If no KeyText dialog or "Edit text items" window is open, opens the "Edit text items" window at the last item to be edited or run.
If a KeyText dialog or "Edit text items window" is open but not in the foreground, brings it to the front.
If the "Edit text items" window is open and at the front, saves and closes it (presses OK).

3) KeyText has a menu style which includes graphics - particularly useful in indicating the expected action which will occur if a left-click item is selected. You can choose to have standard Windows menus here.

4) Hide scheduled items in left-click menu. Select this to remove from the left-click menu any items which have a schedule set. Such items may not need to be accessed directly from the menu, and hiding them can make non-scheduled items easier to locate. This option applies globally; it also appears on the password dialog for the current .kt3 file, and if set there, overrides the setting on this dialog. So that if the option is unchecked here (don't hide), and you open a .kt3 file with the appropriate password level and hide scheduled items set, they will be hidden.

5) Always display left-click menu in one column. By default, the KeyText left-click menu will split into 2 columns once it has sufficient items, avoiding the danger of the menu running off the top or bottom of the screen. However, a small number of Windows 98 users report that the 2-column menu becomes misplaced - a problem linked to specific display drivers. This option was added as a workaround for users who experience this problem.

6) The left-click menu has 4 popup submenus which offer shortcuts to certain KeyText functions. These can be omitted, but note that KeyText functions for automatically adding text to an item will then be unavailable.

7) Select this to add a "Right-click menu" item to the left-click menu. This enables keyboard-only users to access the right-click menu (press left-click hotkey first) without using a mouse.

8) Select an item here, and an extra option will be added to the left-click menu to add text directly to that item. This simply means one less keypress or mouse movement compared to using the "Add selected text to >" popup menu. Uncheck if you do not require this menu item to be present.

9) These options tell KeyText to react differently if a KeyText item is *right*-clicked in the right-click anywhere, or in the left-click menu. The choices are 1) no difference (right-click is the same as left-click), 2) do the alternative action (an item set to Action: Macro is pasted, and an item set to Action: Paste is treated as a macro), 3) add a Tab, 4) add a Return, or 5) open the Edit text item window for the item.

10) KeyText icon - left double-click action. By default, a left double-click with the mouse on the KeyText icon in the tray opens (or closes) the KeyText Keypad. Alternatively you can set it to start text item A - for convenience, you can store your most frequently used text there. However, if you have a slow double-click setting, you may find that the action starts accidentally - especially if you use a second left click to dismiss the left-click menu. Select "None" to disable the double-click feature.

11) Auto-add audio reminder. Auto-add gives the ability to automatically assemble texts copied to the clipboard into one text item. However, it is easy to forget it is switched on. The KeyText icon give a visual clue, and you can choose to have a "stapler" sound play every time KeyText auto-adds text.

12) KeyText displays warning / reminder messages when for example, you copy selected text to an item which already has text, or have auto-add left on for a length of time. If you have chosen "Don't remind me again" on these warnings, you can reset them here so that they appear again.

2.12.6 User Information

To go to User Information, right-click the KeyText icon and select Settings; the User Information dialog is the fifth tab.

The settings in User Information apply "globally", unlike Options and Password that apply only to the current .kt3 file.

If you entered your name when you first ran KeyText, you will see it displayed here, and can set or change the hotkey to it. The name given will appear at the top of the left-click menu, to give a convenient way of having it easily available all the time.

If you purchase KeyText, enter the code you are given under "Unlock code:". Note that the name *must* match exactly that supplied with your code, or the registration will fail.

If you wish your name to appear differently, please contact MJMSoft Design. For example, you may have specified only the initial letter of your first name when you purchased, but would like your first name to appear in full. Contact support@mjmsoft.com for help.

2.13 Hotkeys

KeyText supports hotkeys in four different ways.

1) Left-click menu hotkey. You can specify a hotkey in Options which will open the left-click menu. This shows you the available items for the .kt3 file; click one with the mouse or press the underlined letter to select it. This hotkey is specific to the current .kt3 file, but note that you can choose in global settings to keep this hotkey active even when another .kt3 file is loaded; this provides an easy way of switching from one set of text items to another. This hotkey can also be specified in a command line, over-riding any hotkey already set for the file being opened.

2) Hotkey to individual text items. In the Edit text items window you can specify a hotkey to that item. Note that if you press a hotkey to start an item, it will not start to the macro, paste etc. until you release the hotkey.

3) User name hotkey. In the User Information dialog you can specify a hotkey to your name, as specified when you first started KeyText, or purchased it.

4) Edit hotkey. In Customize you can set up a hotkey to make it easier to switch between editing and testing an item.

Note that the Windows logo key (if you have one) can be included in a hotkey combination (although note that a number of Windows logo key combinations are reserved for Windows use, and you may not be able to set them).

If you want a hotkey for a single keypress, you must use either the function keys f1, f2 etc, or the number keypad (but only if [Num Lock] is selected). In other cases KeyText requires that Ctrl or Alt be part of the combination, and will add Ctrl automatically if neither is specified.

If a hotkey is specified but does not appear to be active, check that it is not duplicated by another application, or another instance of KeyText.

2.14 Multiple "Instances" of KeyText

Each time you run KeyText an "instance" of it sends an icon to the tray. If you turn on the multiple instance feature in the global settings, you can run it again and a second instance sends an icon to the tray - and so on. Try using alternative KeyText icons to distinguish between them; see Options.

If you use multiple instances, consider using the extra KeyTexts option in the global settings. If this is set at, for example, 2, every time you run KeyText, it will open with the .kt3 file you used last, and start two additional instances with the previous 2 .kt3 files. If this is set, you can override it by holding down the SHIFT key when KeyText starts (doing this also suspends the scheduler).

See the tutorial for further information.

2.15 "No icon" option

In Options you can choose to have no KeyText icon in the tray. This is only available if you have a valid hotkey set in Options - you can then choose "None" under KeyText icon style. All hotkeys continue to function, giving KeyText functionality with no space taken up on your desktop.

Use this option with care, because there will be no indication that KeyText is running! *Remember the hotkey you set*, because this is the only way of bringing up the KeyText menus.

If you do forget and therefore cannot get access to your texts, next time you run it hold down the SHIFT key. This will force KeyText to start with an icon showing, and you can then check the hotkey on the Settings/Options dialog.

Like other hotkey and icon settings, this choice applies only to the current .kt3 file - but you can have more than one file with no icon associated.

You may find that the .kt3 will sometimes run *with* an icon. This indicates that the hotkey for that .kt3 is already being used by another program, or another instance of KeyText. If the hotkey cannot be set, then invisible operation is useless!

See Hotkeys for more information.

2.16 KeyText Keypad

For convenience, KeyText offers a Keypad which will stay on top of all other windows until closed. To activate it, right-click the KeyText icon and select KeyText Keypad, or left double-click the KeyText icon in the tray (assuming the Keypad option is set in Settings / Customize).

The Keypad lets you "type" using your mouse or other pointing device.

The Keypad window has two tabs. One shows a miniature keyboard, and lets you "type" with your mouse - a feature asked for by users who do not always have a keyboard attached to their (portable) PC. The other shows the standard symbols and accented characters, and provides an easy way to insert these into an application.

The Keypad does its best to keep track of the place you are typing to, but if the window you are typing to loses keyboard focus, that is, it is no longer active and the caret is not present or static, simply click the window with your mouse at the point where you want text to go, and continue using the Keypad.

Please note the following:

On pressing the Keypad shift key, shifted characters are shown where appropriate.

The Keypad is based on a standard US keyboard, and will not reflect regional differences.

The Caps Lock button activates Caps Lock *only* for the Keypad itself, not for any physical keyboard present.

Combinations such as Alt-Tab and Ctrl-Alt-Delete are not recommended, nor are using Alt-letter combinations to open menus. Change windows, shut down, select menu items etc. with your mouse in the normal way.

The Windows logo key can act as a shift key, and will stay "down" until another key is pressed. On an actual keyboard it can also be used as a normal key, and will bring up the Start menu; in KeyText a double-click on the Windows logo key will simulate this - but note that a subsequent click on a Keypad key will immediately dismiss the Start menu - so this feature is of limited use.

The Application key (looks like a menu-selection button) normally produces the right-click context menu for the active application. Note that because the mouse is over the Keypad when selecting the Application key, the menu may not appear at the expected location.

The Keypad may not operate correctly in combo boxes; see known limitations.

Part III

3 KeyText Organizer

3.1 Introduction

The KeyText organizer window gives access to the KeyText **scheduler** which can "run" an item at a specified time or regular interval. Because the text in an item set to "macro" or "run" can include commands (fields), you can run programs at specified times or intervals, display reminders or alarms with your choice of sound, even have an hourly chime. These are all set up through the Set Schedule window.

If your scheduled item includes a Message field, when the message is displayed it includes a "Repeat" button, so that you can choose to have it repeat after a certain number of minutes (1 to 240), or change the schedule.

As well as having items which run at a certain time, you can have items scheduled to start whenever a certain window appears. See Windows automation for more information.

Note that the "paste" action is not recommended for an item scheduled for a certain time - nor is an item which begins with text, and is set to "macro". This is because KeyText will assume you want to use the current foreground window when the scheduled item is actioned - it may start typing or pasting at an inconvenient moment!

Note that a KeyText data file (.kt3) with scheduled events in it must be open with KeyText running. If the .kt3 file is not open at the time of a scheduled item, it will be missed - and no warning is given later to indicate a missed scheduled item. It is recommended that you have KeyText running all the time, if necessary using multiple instances to make sure that .kt3 files with schedules set are open.

To open the Scheduler, right-click the KeyText icon and select Organizer.

Only one scheduled event can run at a time. For example, assume you two text items, both set up as reminders with message boxes. One is set for 9:00am every day, the other for 9:10am. The first reminder will appear at 9:00am, and "runs" until the OK or Cancel button is clicked. If this doesn't happen until *after* 9:10am, the second reminder will not appear on time; instead it will wait until the OK or Cancel button on the first has been clicked.

Note also that if a KeyText dialog is displayed, or the Edit text items window; KeyText will wait until all dialogs are closed before processing any scheduled events.

You can suspend all scheduled events by selecting "Suspend scheduled items" on the right-click menu (only visible if there are scheduled items set), checking the box at the top of the Organizer window or, to start KeyText with the scheduler suspended, hold down the SHIFT key while KeyText starts. This also prevents automatic extra KeyTexts starting, if this option has been set. See what happens when KeyText starts.

Important: If the Scheduler is suspended and a timed event passes, it will be missed. For example, if you have an item set to run at 9:00 on the last Friday of each month, and at that time KeyText is either not running or the Scheduler is suspended, the item will not run that month. For an item set with a Minutes schedule - to run every *n* minutes, if the Scheduler is suspended at that time, it will count another *n* minutes and try again then.

On the Edit text items window if an item has a schedule set, the schedule button on the toolbar has a green surround. If it has a schedule set, but the scheduler is suspended, the surround shows red.

Scheduled items can be set to minutes, hourly, daily, weekly, monthly, yearly, once, or On Window. See the KeyText Scheduler to find out more.

3.2 Organizer Window

The KeyText Organizer gives access to KeyText's Scheduler features. For general information, see the KeyText scheduler. Note that changes made here are effective immediately - there is therefore a Close button, rather than OK and Cancel.

You can change the order of KeyText items - the up and down arrow buttons move the highlighted item up or down. Or press Alt and the up and down cursor keys to do the same.

Note that as you move an item up or down the list, the other items all move up to fill the space; their location therefore changes on the left-click menu or Edit text items window. If you are used to selecting an item by opening the left-click menu and pressing the letter associated with it, this may change.

An alternative is to use the swap button ⇕. This lets you swap the locations of two items, while all the others stay in the same position. Select the first item and click the swap button; then select the item to swap it with and click the swap button again: the two items will change places. You can cancel the swap operation by clicking the swap instruction button which opens at the bottom of the dialog. You can also press Alt S to select the swap button.

The main window on this dialog lists your KeyText items, indicating the action to be carried out if one is selected, any hotkey, and whether a schedule is set for that item. Select the required item before selecting the Add, Edit or Remove buttons.

See Set Schedule to find out how to add or edit a scheduled item. You can suspend all scheduled items by clicking the check box at the top left of the dialog. You can also stop KeyText from monitoring the keyboard for trigger text by clicking the check box on the top right.

Click **Add schedule...** to add a schedule to the selected item.

Click **Edit schedule...** to edit an existing schedule on the selected item.

Click **Remove schedule** to remove a schedule from the selected item.

Click **Timer settings...** to change the timer settings for all scheduled items.

Click **Edit text item** to go to the Edit window for the selected item - or double-click the item.

Click **Clear text item** to delete the contents of an item. This clears the whole item - text, menu text, hotkey and any schedule associated with it.

Click **Group names** to set names for some or all of the A-Z sets of items. The name of a group (for example, Personal, Accounts, Work) will be displayed in the title bar of the Edit text items window, and in the left-click menu just below your name. Additionally, the left-click menu item "Go to group..." will display a list including group names, allowing you to jump straight to the required set of items.

Click **Copy list** to copy to the clipboard a list of all the KeyText items - their location (A,B,C etc), their menu text (or first part of the item contents if no menu text is set) and a hotkey, if one is set. After clicking Copy list you are also given the option of including the full text of each item. You can then paste the list into a Word Processor, format it as you require, and print it out for use as backup or reference. If you use a number of hotkeys it can be particularly useful to have a list of them to hand.

Click **Desktop Shortcut** to make a desktop shortcut to the highlighted item.

Click **Manage Variables** to manage global string and integer variables.

3.3 Windows automation

This relates to Windows automation; for general information, see the KeyText scheduler.

You can set KeyText to look out for a specified window, identified by its title and, if necessary, some static text in the window to identify it more accurately. Whenever a matching window comes to the front, KeyText will start the text item associated with the schedule.

A text item can include special fields to click a button, select a menu item and more, so this feature gives the possibility of automating many of your common Windows tasks.

For example, a password request dialog can be completely automated so that in future, with the .kt3 file running in KeyText, the password box will be filled in automatically whenever it appears - no intervention required! If the password dialog title is not enough to identify it accurately, find some unique identifying text within the window and specify that in the Advanced dialog. See Tutorial for step-by-step examples.

If the "Match whole of title" box is checked, then you must give the full title on the caption bar of the window you want to look for. If unchecked you can specify any part of the title. Case is not significant.

You also need to specify what KeyText should do once it has found the required window. If "wait for a new window with the same title..." is checked, then once KeyText has found a matching window and started the associated item, it will not respond to that particular window again, but wait for a new window with the same title.

Alternatively, specify a period to wait before checking again. In this case, if the same window is still in the foreground or is brought to the foreground after the specified delay, the item will be started again. It is possible for KeyText to enter a loop if it repeatedly finds the same window. To get out of the loop, right-click the KeyText icon, and click "Suspend scheduled items".

See also the wait field, which can be used to tell KeyText to pause and wait for a certain window to appear - while an item is running.

3.4 Set schedule window

For general information, see the KeyText scheduler.

To set an On Window schedule, select the radio button which says: "Start text item ? whenever this window title appears."

To set an item to start at a specific time, select "Start text item ?:" and select from the combo box: by minutes, hourly, daily, weekly, monthly, yearly, or once. Boxes appropriate to your choice will appear below the combo box - select the time/interval/date you want and press OK.

Press the "Now" button to set the appropriate boxes to the current date or time.

The "Minutes" and "Seconds" options work differently from other timed schedules. The others will start at a set time or times, whereas a Minutes or Seconds schedule it starts at set intervals - the number of minutes or seconds you set in the Scheduler. You can choose whether the item will start automatically - immediately after the schedule is set or after startup, or "by user" - it will first run when you select the item manually, and only after that run at regular intervals. Note that it will time its next run to the given number of minutes or seconds *after the item starts*, so that the item should run at approximately the scheduled intervals. Note that if an item takes, for example, 20 seconds to run and it is scheduled to start every 15 seconds, it will start approximately one second after each time it finishes.

With an Hourly schedule you can choose how many minutes past the hour, and which hours the item will start on. This gives the possibility of starting an item, say, at 9:00am, 12noon and 5:00pm.

With a Daily schedule you can choose which days - for example, Monday to Friday only.

With a Weekly schedule you can set the day of the week - and which weeks in the month. For example, on the 1st and 3rd Tuesday of a month, or the last Friday of a month. The "Last" option could be either the 4th or the 5th occurrence, depending on how many times that weekday occurs in the month.

With a Monthly schedule you can choose which months the item will run on. For example, the 1st of every 2nd month. When specifying the date, note that the drop-down list includes Last day, Last day - 1… Last day - 7, so that it is easy to set an item to run, for example, on the 3rd last day of every month.

Using this feature you can schedule programs to start at specified times or intervals, display messages or reminders at specified dates or times, even set an hourly chime using a suitable .wav file.

Note that if a KeyText dialog is open at the time specified, the scheduled event will be delayed until the dialog is closed. Also note that KeyText must be running at the time of a scheduled event for it to be started.

3.5 Advanced On Window settings

The first option is "Windows to check". Normally a new window or dialog would come to the front, and therefore checking the Foreground window only will find it.

Checking "All windows" allows all windows to be monitored, not just the foreground window. This means that if an application running in the background generates a window or message which does not come to the front - for example, a "You have new email" message - KeyText can be set to look out for it and perform whatever actions you have in the associated item.

There are two important points to note when using "All windows".

1) **If the window is found and is not in the foreground, KeyText will not bring it to the front automatically**. You should have an Activate field in the associated item if this is what you want. For example, if KeyText spots a "You have new email" message in the background, you would need {Activate "Title of email program"} at the beginning of the item if you wanted to bring it to the front.

2) **Note that if "All windows" is checked in any "On Window" schedule, KeyText continually monitors all windows on your system, which uses more processor time than the "Foreground window" option**. It is recommended that if using this feature you click the "Timer settings…" button on the Organizer window and reduce the frequency of checking to, say, 1 or 2 times every second.

If the window title is not sufficient to accurately identify the window/dialog you want, enter any fixed (static) text in the window which will identify it better. For example, different applications may produce a dialog with the title: "Enter password". So that KeyText knows which is the correct one, look for some text in the window which is unique and always there, and specify it in the Advanced window; the correct dialog may have static text such as "Password for ABC Systems" - specifying "ABC" in KeyText will identify it. A maximum of 20 characters may be specified. But note that some applications produce non-standard dialog boxes - in which case KeyText may not be able to find text specified here, and therefore not react to that dialog.

Some applications display a window title which changes after loading is complete. For example Notepad may display "Untitled - Notepad" briefly before changing to the name of a text file being opened. This may result in KeyText starting unintentionally. Specify a time in tenths of a second; after that gap, KeyText will double-check the title and, if it has changed, not start the associated item.

You can restrict the number of times a scheduled item is started. Choose from 1 to 10 times, and specify the period in which it applies. For example, if an item clicks a "Redial" button, you may wish to restrict it to 5 clicks within any 10 minute period. The period can be from 10 seconds to 8 hours, or choose "session" to restrict the number of times the item is started while the current .kt3 file is open (without time limit).

3.6 Manage global variables

Variables can be used in both Action: Macro and Paste items.

Integer variables
26 integer variables are available in each item, local to that item, and named a,b,c....x,y,z. They default to zero, but their value is carried into any called or linked-to items (that is, the scope is the current item and any items it calls or links to). They can be associated with global variables which are retained from one session to another, are available to all KeyText files, and can be maintained in the Manage global variables dialog which is accessed from the KeyText Organizer.

Local variables are associated with global variables by using a field similar to this:
{#v@"Counter"}

When an item with a field like this is run, KeyText looks up the global variable and if it is already set, the variable v is given that value. If it is not already set, a new variable "Counter" is created and given the current value of v (normally zero, unless v has been given a value earlier in the item). When the item finishes, the new value of v is stored in the variable "Counter".

The current value of any global variable can be viewed or changed using the Manage Global Variables dialog - which also allows you to set up new global variables.

Additionally, you may wish to reset automatically global variables at set times. For example, you might have a variable MonthlySaleCounter which you want to reset to 1 on the first day of each month.

The Manage Global Variables dialog allows you to set a variable to reset to a specified value:
- Every time KeyText starts
- At a set time every day
- On a set day of the week
- On a set day of each month.

String variables
String variables can be "global", as indicated by the @ symbol at the end of the variable name. Global string variables can be accessed from any item - and from any instance of KeyText, allowing sharing of common text strings between KeyText data files - and even between users.

In the Manage Global Variables dialog you can add, delete or edit the contents of global string variables. You can also set the file location where they are stored so that, for example, a shared "KeyText.gsv" file could be located on a network.

With this in mind, it is possible to Prevent changes to global string variables - to reduce the ability of multiple users to make unwanted changes. A password is asked for the first time you select "Prevent changes…", and thereafter it is required to make changes to the changes setting.

It is possible, after selecting "Allow changes…" to specify that they are allowed for this session only (in which case other users without password access will still see it as locked to changes; or specify that they are allowed permanently - in which case all users can make changes, until the password is again used to prevent changes.

3.7 Desktop shortcuts

In the Organizer dialog you will find a button "Create desktop shortcut", which adds an icon to your desktop which, when double-clicked, will run the item. It helps you take advantage of one of KeyText's command line features.

Bear in mind that the item should not be set to Action: Paste, or with straight text set to Action: Macro; in both cases, because you have just double-clicked a desktop icon, KeyText will not know where to direct the paste or type operation. The item should begin with a field such as Run, Activate, Mouse, etc.

Once the shortcut has been double-clicked, one of the following scenarios will take place:

If the specified .kt3 file is already running, the item will be started, and KeyText will continue running.

If no KeyText is running, it will run, start the item, then exit.

If KeyText is running but with a different .kt3 file, it will momentarily run a second instance of KeyText, start the item, then exit the second instance.

If the specified .kt3 file is already running but a dialog or edit window is open, KeyText will momentarily run a second instance of KeyText, start the item, then exit the second instance.

If KeyText is running with an item in progress - say in Pause mode - KeyText will report that it is busy; this is the only situation in which the item will not run.

3.8 Timer settings

Timer for checking for a specified window
Use the up/down arrows to select the number of times per second you want KeyText to check. In general, selecting a higher number means that KeyText will respond more quickly when the window appears, but it also means that your system is doing more work - checking more often.

Note that this timer will only run if the open .kt3 file has a schedule set to "On Window".

Timer for checking for a specified date / time.
This timer setting works differently from the above, and specifies the number of seconds between checks. In general, a lower number means that KeyText will be more accurate in starting the associated item. A higher number may mean, for example, that it could start up to 20 seconds late - but your system is doing less work, and your system clock is probably not that accurate anyway!

Note that this timer will only run if the open .kt3 file has a schedule set to a specified date / time.

Part IV

4 Reference

4.1 Field syntax

KeyText has a range of fields to perform various actions, normally in text items set to Action: Macro. KeyText checks text between { and } characters (or other delimiters if set) for one of the fields described below. If not found, it will check to see if it is a key or key combination such as {Delete}, {Ctrl-Shift Z} etc. If it is, KeyText checks for a number following it, and if present repeats the key(s) the specified number of times; for example, {Tab} presses the [Tab] key once; {Tab 10} presses it 10 times. If the field has still not been recognized, KeyText will treat the field as ordinary text. If, when an item is run, a field command fails: for example, {Click "OK"} when there is no OK button, the item will stop at that point.

In considering the Actions specified below it should be noted that an item set to Action: Run is treated like a batch file, with each line either as a program to run, or a Run/New field. If parameters etc. are required, the Run field should be used. Action: Paste treats the item as text to be pasted as a whole, while Action: Macro acts more like a script, typing simple characters and performing commands specified in fields in a sequential manner.

All the fields* can be set up with the help of the Insert Field Wizard, and therefore the information below is for reference. The wizard should normally be used.
*Except for some of the integer/string variable fields.

Notes on quoted text in fields
Many fields have parameters which are text wrapped in double quotes. For example {Message "Title" "Body text"}. Please follow these guidelines:

- If the text includes a double quote character, wrap it in single quotes instead (if using the Insert field wizard, this will happen automatically).
- You can nest many fields, for example {Message "First 3 characters of clipboard" "{Clipboard "\(.\{3\}\).*" "\1"}"} or {Pause {#a}}.
- If you want a left brace, use {{. For example, {Message "Left brace" "The left brace character looks like {{"}.

The following field may be used with Action: Macro, Paste or Run

*** ***
Used to insert a comment which will be ignored. For example, {*This text will not appear when typing or pasting this item*}.

The following fields may be used with Action: Macro or Paste

Ask[***char***] [**L**][**^**][**-**][**%**]***name***[@] ***question***
Ask fields ask for user data which is then filled in before the item starts or is pasted. ***name*** appears in the title of the dialog which requests the data, and ***question*** appears in the body of the dialog, above a text box where you can type the data to be filled in. ***name*** is significant because the result is stored in a variable of that name. For example, {Ask "First name" "Enter customer's first name"} would be filled in with your response to the question, as would subsequent uses of {First name}.
If ***name*** is followed with @ the Ask result is stored as a global variable called ***name***. Note that the @ is not displayed in the dialog.
If a single character ***char*** is specified, this character will appear instead of actual characters typed when the Ask field is run. For example Ask*... will display asterisks, as in a password entry box.
If present, **L** indicates that ***question*** should be left-aligned on the Ask dialog; otherwise it is centered.
If present, **^** indicates that the Return key will "OK" the Ask dialog; otherwise the Return key will start a new line

in the response text.
The **-** character indicates that your response to the Ask field should not be output at that point in your macro; this may be useful if your only use for the response is in {If...} fields.
The **%** will encode the text for a URL; characters not allowed in a URL are encoded - for example a SPACE character becomes %20. Note that **-** and **%** cannot be used together, as if no output is required there is nothing to URL encode. Also note that URL encoding applies to this output of the Ask result. For example, if you have an item with: {Ask %"First names" "Enter customer's first names"} and enter John James, this would be output as John%20James. Subsequent use of {First names} would result in John James; use of {First names%} would result in John%20James.

name [**%**][[**d**]***expression replacement***]

As stated above, use of the ***name*** from an Ask field (same as a string variable) will repeat the response to the Ask field; if **%** is used it will be URL encoded (eg, SPACE character becomes %20).
The optional expression replacement allows a regular expression to be defined (within single or double quotes), with a corresponding replacement expression (within single or double quotes). This gives the possibility of extracting and manipulating text from the ask field result, ready for output. The **d**, if present, will display a regular expression debug dialog showing error and other regular expression information. See also String variables below.

Clipboard [**X**][**%**][[**d**]***expression replacement***]

Inserts the contents of the clipboard. There must either be text on the clipboard, or a text file - for example, a .txt file copied to the clipboard using Windows Explorer. If the **X** is included, as in {Clipboard X}, any trailing carriage returns or line feeds are removed. This is useful, for example in copying a spreadsheet cell, because Excel, in copying a cell, adds a carriage return and line feed to the text format of the cell. If there is no text on the clipboard, the Clipboard field is ignored.
The **%** indicates that the text will be encoded for a URL; characters not allowed in a URL are encoded - for example a SPACE character becomes %20.
The optional ***expression replacement*** allows a regular expression to be defined (within single or double quotes), with a corresponding replacement expression (within single or double quotes). This gives the possibility of extracting and manipulating text from the clipboard ready for output. The **d**, if present, will display a regular expression debug dialog showing error and other regular expression information.
Note that Clipboard fields are filled in with the clipboard contents at that point in the item, unlike earlier versions of KeyText which filled it in with the start clipboard contents. (For that behaviour, use {StartClipboard}).
Note also that the Clipboard field can be used inside other fields. For example if a KeyText item has {Run "{Clipboard}"} and the text Notepad is on the clipboard, Notepad will run.
The Clipboard field should not be used in the same KeyText item as a Selection field, because on detecting a Selection field KeyText will copy the current selection to the clipboard, thus changing its contents before the Clipboard field is filled in.

Date[**+*n***[***code***]] [**Short** / **Long** / ***User***]

Inserts the current date and/or time using the specified format. **Short** and **Long** refer to the pre-defined Windows date formats set up in Control Panel, Regional Settings, Date. If not specified, long is assumed. Or you can define a ***User*** date, enclosed in single or double quotes, and using the following format letters:

y, M, d, h, H, m, s or t	Numbers with no leading zeros for single-digit years, Months, days, hours (12), Hours (24), minutes, seconds, or A/P
D	Similar to d above, but gives day of month as an ordinal: 1st, 2nd, 3rd etc.
yy, MM, dd, hh, HH, mm, ss or tt	Numbers with leading zeros for single-digit years, months, days, hours (12-hour clock), Hours (24-hour clock), minutes, seconds, or AM/PM
MMM or ddd	Three-letter abbreviations for the Month or day
yyyy, MMMM or dddd	Full name of the year, Month or day

For example, {Date Short} - or just {Short} - will give a date like 10/18/06 (depending on your Windows default

date format), while {Date "dddd MMMM d, yyyy"} will give a date in the form Wednesday October 18, 2006.
You can generate a date which is ***n*** days or months before or after the current date. The date field will be updated correctly each time the item containing the field is used. The wizard will generate this code for you, but for reference it is given as follows:
+ can be + or -, indicating forwards or backwards in time. ***n*** is a number from 0 to 99999, indicating the number of days or months to move from the current date.
If ***code*** is omitted, days are assumed: forward or back ***n*** days. Otherwise ***code*** can be:

d	days, same as if omitted
r	working days only: counts Monday to Friday but ignores weekends
m	months: forward or back n months, using same day of month as current date
mf	months as above, but giving the first day of the resulting month
ml	months as above, but giving the last day of the resulting month

n should only be 0 (zero) if mf or ml are given as codes, which will give the first or last day of the current month. Otherwise 0 simply indicates the current date and is not required.
For example, {Date+10r} will give the date 10 working days forward from the current date. {Date-1ml} will give the last day of the previous month.

EnvVar *name*

Inserts the Environment Variable name (for example SystemRoot, TEMP, USERNAME). If entering this field using the Insert Field Wizard, a warning is given if the named Environment Variable does not exist.

If title/exe *operator text* cancel

Normally "If" fields are reserved for Action:Macro items, but there is a special use of the If field at the beginning of Action:Paste items which allows you to check the window title or exe (or process) name of the foreground window, and cancel.
operator can be one of the following:

=	equals: looks for an exact match with ***text***
not =	(or !=) - not equals: must not equal ***text***
contains	contains ***text***
not contains	(or !contains): does not contain ***text***
like	matches the given regular expression
not like	does not match the given regular expression

A sample use of this is: you have an Action:Paste item which pastes a paragraph of text whenever you type the Trigger text abc - but you don't want this to happen in Word. If you have this at the beginning:
{If title contains "Word" cancel}
then the item will not paste when you type abc. You can specify more than one such field, separated by spaces or returns.

Include [X][%]*file* [[d]*expression replacement*]

Inserts the mentioned ***file***, being a file with full path given enclosed in single or double quotes. The file can be a .bmp file, in which case the unformatted bitmap image will be inserted at that point (applies to Action: Macro only), but normally indicates a text file, the contents of which are automatically included in the text item. For example, {Include "C:\Docs\message.txt"} will place the contents of message.txt in the outputted text. If Include **X** is used, any carriage returns or line feeds at the end of the text file are removed. The **%** indicates that the text will be encoded for a URL; characters not allowed in a URL are encoded - for example a SPACE character becomes %20.
The optional ***expression replacement*** allows a regular expression to be defined (within single or double quotes), with a corresponding replacement expression (within single or double quotes). This gives the possibility of extracting and manipulating text from the file ready for output. The **d**, if present, will display a regular expression debug dialog showing error and other regular expression information.

Link *A*

Instructs KeyText to finish the current item and immediately process item ***A***. For example, if you have signature text in item A, and various e-mail body texts in B to M, instead of including the signature in each of B to M, you could finish each with {Link A}.

LinkRand *ABC...*

Instructs KeyText to finish the current item and immediately process another item, chosen randomly from the list ***ABC...***. For example, if you have signature text in item S, and various quotations in A, B, C and D, you could end your signature with {LinkRand ABCD} and each time it is used a quotation will be added at random from the list A, B, C and D.

LinkSelect *ABC... title body*

Instructs KeyText to finish the current item and process another item, chosen by the user from the list ***ABC...***. When this field is reached, with either Action: Macro or Action: Paste, KeyText displays a dialog with the specified ***title*** and ***body***, with a series of radio buttons showing the menu text of items ABC.... Choose one and press OK to link to it, or Cancel to abort. ***title*** and ***body*** should both be enclosed in single or double quotes. For example, {LinkSelect XYZ "Select date" "Please choose a date from the following list:"} would produce a dialog with title "Select date", message "Please choose...", and under that 3 radio buttons showing the menu text for items X, Y or Z. In this example, they could offer dates 7 working days away, 10 working days, or last day of current month. Ordinary text could precede the LinkSelect field, the appropriate date could be chosen and entered, and each of X, Y and Z could then Link to another item for continuation text.

LinkSeq *ABC...*

Instructs KeyText to finish the current item and immediately process another item, chosen sequentially from the list ***ABC...***. For example, if you have signature text in item S, and various quotations in A, B, C and D, you could end your signature with {LinkSeq ABCD}. The first time item S is used, it will add quotation A. The second time B, then C, D, A, B and so on.

Time

Inserts the current time using you pre-defined Windows time format as specified in Control Panel, Regional Settings, Time. You can specify different formats by using the **Date** field - see above.

Selection [X][%][[d]expression replacement]

This field is the key to KeyText's SMART SELECT feature. Before the item is started, KeyText copies any text (or text file) selected in the foreground application, and uses it to fill in any Selection fields. If the **X** is included, as in {Selection X}, any trailing carriage returns or line feeds are removed. This is useful, for example in copying a spreadsheet cell, because Excel, in copying a cell, adds a carriage return and line feed to the text format of the cell.
The **%** indicates that the text will be encoded for a URL; characters not allowed in a URL are encoded - for example a SPACE character becomes %20.
The optional ***expression replacement*** allows a regular expression to be defined (within single or double quotes), with a corresponding replacement expression (within single or double quotes). This gives the possibility of extracting and manipulating text from the selection ready for output. The **d**, if present, will display a regular expression debug dialog showing error and other regular expression information.

The following fields may be used with Action: Macro or Run

New *extension*

Starts a new file with the specified ***extension***. KeyText will start the application associated with the selected file extension with a blank document, page, etc. To find out what filetypes can be specified, run Windows Explorer, select Options from the View menu, and click the "File Types" tab. For example, {New "DOC"} will run the application associated with the .doc extension, with a blank page ready.

Run [^/*[!]*title*] *program* [*parameters directory*]

Runs ***program***, being a program with full path given. It may also be a document, or any file which has a file type

registered with your Windows system; KeyText will run the associated application, loading the file you specify. ***Parameters*** may be given if you want to include command line parameters - startup information which would be included in a Run command line (exe only). Some applications need to be run with a default ***directory*** specified - normally the directory the application is located in. If you run an application using KeyText and it gives a message indicating it cannot find some of its files, you need to specify a default directory; consult the application's documentation for further information. If you first want to check if the program is running, use the ***title*** argument to specify any part of its window; if the window is found, it is activated (brought to the front) and the specified .exe etc. is not run. All arguments should be enclosed in single or double quotes. For example, {Run "Notepad.exe"} will run Notepad; {Run ^"Notepad" "Notepad.exe"} will run Notepad, unless an existing window has Notepad in its title - in which case it will be brought to the front. {Run "Notepad.exe" "C:\Personal\MyFile.txt" ""} will run Notepad with MyFile.txt loaded, and {Run "C:\Personal\MyFile.txt"} will run the program associated with the .txt extension, with MyFile.txt loaded. {Run "C:\MyProg\MyProg.exe" "" "C:\MyProg"} will run MyProg.exe with C:\MyProg as the default directory.
The **!** (exclamation mark) option indicates that a whole match should be done on title when checking if the program is already running. For example {Run ^!"My Program" "C:\Personal\MyProg.exe"} would run MyProg.exe unless a window existed with a title of "My Program".
The ***** (asterisk) option is an alternative to the **^** which specifies a window title to check for first. It applies to users of Windows 2000/XP/Vista/7, who may find problems with a Windows feature which inhibits a program's ability to bring another window to the front; instead the desired window's taskbar button flashes. If required, the ***** implements KeyText's "Force Activate" feature, which may help to resolve this problem, and bring the desired window to the front. The example above would become {Run *"Notepad" "Notepad.exe"}.
[Note that in KeyText 2 "!" was used to indicate the force activate feature, but for consistency with the Wait field, "!" now indicates that a "match whole of title" is required. To ensure backwards compatibility the old format of {Run "!Notepad" "Notepad.exe"} (where the ! is after the opening quote) will indicate force activate.]

Run URL
URL can be a www address with the http:// prefix, or an e-mail address with the mailto: prefix. When the former is run in a KeyText item, your default browser will launch and take you to the specified http:// address. If the latter, your default e-mail program will be run (if not running already) and a new e-mail started addressed to the specified address. For example, {Run "http://www.mjmsoft.com"} will take you to the MJMSoft Design web site, and {Run "mailto:support@mjmsoft.com"} will start an e-mail to MJMSoft Design.

The following fields may only be used with Action: Macro

Activate [*n*] [*****][**!**]***title***

The ***title*** argument is whole or part of the text in the title bar of the application you want to activate, enclosed in single or double quotes.
Causes KeyText to activate (bring to the front) the specified window. If not found, the KeyText item will halt. For example, {Activate "Notepad"} will bring Notepad to the front, if running.
If ***n*** is specified (must be >0), KeyText will activate the ***n***th window matching the title specified. For example, {Activate 2 "Notepad"} will bring the second-down Notepad window to the front; note that if a second Notepad window is not found, KeyText will halt.
The **!** (exclamation mark) option indicates that a whole match should be done on title when checking if the program is already running. For example {Activate !"My Program"} would only activate a window which had "My Program" as the whole title.
The ***** (asterisk) option is applies to users of Windows 2000/XP/Vista/7, who may find problems with a Windows feature which inhibits a program's ability to bring another window to the front; instead the desired window's taskbar button flashes. If required, the * implements KeyText's "Force Activate" feature, which may help to resolve this problem, and bring the desired window to the front. The example above would become {Activate *"Notepad"}.
[Note that in KeyText 2 "!" was used to indicate the force activate feature, but for consistency with the Wait field, "!" now indicates that a "match whole of title" is required. To ensure backwards compatibility the old format of {Activate! "Notepad"} (where the ! is immediately after Activate) will indicate force activate.]

Beep

Plays the system default beep sound.

Call *A*

Instructs KeyText to process item ***A***, then return to the original item which had the Call field and continue.

CallRand *ABC...*

Instructs process another item, chosen randomly from the list ***ABC...*** and then return to the original item which had the Call field and continue.

CallSelect *ABC... title body*

Instructs KeyText to process another item, chosen by the user from the list ***ABC...*** and then return to the original item which had the Call field and continue. When this field is reached, with either Action: Macro or Action: Paste, KeyText displays a dialog with the specified ***title*** and ***body***, with a series of radio buttons showing the menu text of items ***ABC...***. Choose one and press OK to link to it, or Cancel to abort. title and body should both be enclosed in single or double quotes.

CallSeq *ABC...*

Instructs KeyText to process another item, chosen sequentially from the list ***ABC...*** and then return to the original item which had the Call field and continue.

Cancel

Aborts the current item.

Click *button*

The button argument is the whole text on the button to be clicked, enclosed in single or double quotes. Causes KeyText to click the button. If not found, the KeyText item will halt. For example, {Click "OK"} will click the OK button (even if one of the characters appears underlined on the button). Note that this will only work for a "standard" Windows button - but not, for example, in a button within a browser window.

[**Win** / **-Ctrl** / **-Alt** / **-Shift**] *key*

Presses the specified key combination. For example, {Ctrl-Alt D}, {Ctrl-Shift End}, {Return}, {Page Down}. **Win** refers to the Windows logo key seen on many keyboards.

Else, Else if and Endif

Used in conditional "if" structures where "continue" has been specified in the wizard and therefore the field includes "then". For example:
{If clipboard contains "female" then}She
{Else if clipboard contains "male" then}H e
{Else}It
{Endif}

EndLoop

Marks the end of a section which is looped; not required at the end of an item, as EndLoop is assumed there.

Exit

Exits from KeyText at that point in the item.

Goto *label*

Goes to the specified label, which must be in the form {:label}. Goto fields which jump out of or into Loop or If…then…Else…Endif structures should be avoided.

[**Else**]**If clipboard/selection/title/exe/{*string variable or ask title*}/"{EnvVar *envvar*}"** *operator text action*

This field allows you to check text on the clipboard, the selected text, window title or exe (or process) name of

the foreground window, the value of a string variable (or Ask field result), or an Environment variable. Note that ***envvar***, which is the environment variable name, must be enclosed in single or double quotes - *and* the **EnvVar** field itself must be enclosed in quotes; for example {If "{EnvVar "systemdrive"}" = "C:" then}.
operator can be one of the following:

=	equals: looks for an exact match with ***text***
not =	(or !=) - not equals: must not equal **text**
contains	contains **text**
not contains	(or !contains): does not contain **text**
like	matches the **text**, which should be a regular expression

action can be one of the following:

then	continues item if true, else jumps to next Else, Else if or Endif
cancel	cancels item, else (if false) continues item
link *a*	starts item ***a***, else (if false) continues item
call *a*	starts item ***a*** (and returns to after the if), else (if false) continues item
goto *label*	goes to ***label***, else (if false) continues item

Note that if the action is **then** the item will continue if the result of the "if" is true. For all other actions the item will continue if the result of the "if" is false. Otherwise - if true - it will do the required action: cancel, link, call or goto label.

[**Else**]**If pixel color** [#]***nx,ny operator nr,ng,nb,nt action***

Tests the pixel color at co-ordinate ***nx***,***ny*** where 0,0 is top left. The **#** symbol, if present, indicates that the co-ordinate specified is relative to the current foreground application - otherwise they are screen co-ordinates.
nr, ***ng***, and ***nb*** are the red green and blue values to be matched (or not matched), and must be between 0 and 255.
nt indicates the tolerance required (0 to 99). If tolerance is zero then an exact match is required. [The tolerance is based on the Euclidean distance between the specified and actual red/green/blue values.]
operator can be **=** or **not =**.
action is the same as in the "If clipboard..." section above.

[**Else**]**If** #***v operator n action***

"If" operation on integer variables. ***v*** should be a variable - represented by a single character in the range a-z. ***n*** can be another a-z variable, or a number (positive or negative).
operator can be one of the following:
=
not = (or **!=**)
<
<=
>
>=
action is the same as in the "If clipboard..." section above.
For example: {If #a>100 goto lastSection} or
{If #x=y then}...do x=y processing
{Endif}

IgnoreBlanks [**On** / **Off**]

If **On** is specified (or on/off are both omitted), all return, tab and space characters from that point in the item are ignored and not typed (ie, all white space outside fields).
If **Off** is specified, all returns, tabs and spaces from that point in the item are processed - they are typed.

IgnoreReturns [**On** / **Off**]

This overrides the "Return characters in Action:Macro items" setting in global settings. If **On** is specified (or on/off are both omitted), all return characters from that point in the item are ignored and not typed. Note that this refers to the hidden return characters at the end of lines; {Return} and {Enter} can still be used.
If **Off** is specified, all return characters from that point in the item are processed - they are typed.

IgnoreTabs [On / Off]

If **On** is specified (or on/off are both omitted), all tab characters from that point in the item are ignored and not typed. Note that this refers to tabs generated by pressing the Tab key; {Tab} can still be used.
If **Off** is specified, all tab characters from that point in the item are processed - they are typed.

Input[*char*] [*seconds*[x]] [L][^]*title question*

Displays a message box with the specified ***title*** and ***body***, both being enclosed in single or double quotes. KeyText will display the message box and wait for you to press OK before typing any text you have entered and continuing with the rest of the text, or press Cancel to abort. It differs from the Ask field in that the data is asked for *when it reaches that point in the item*, rather than in the processing stage before the item is run. Use the Input field if there is some variable data in an item which is not repeated, or if you have a loop and want to enter manually one piece of data on each loop.
If a single character ***char*** is specified, this character will appear instead of actual characters typed when the Input field is run. For example Input*… will display asterisks, as in a password entry box.
If ***seconds*** is specified then the input box will automatically close after that number of seconds. KeyText will then type any entered text and the item will continue as if the OK button had been clicked - unless **x** is also specified, in which case any entered text will not be typed and the item will abort as if Cancel had been clicked.
If present, **L** indicates that question should be left-aligned on the Ask dialog; otherwise it is centered.
If present, **^** indicates that the Return key will "OK" the Ask dialog; otherwise the Return key will start a new line in the response text.

Loop [*n*]

Specify the number of times ***n*** (where ***n***>=0 and ***n***<=99999999) to run the text or fields which follow, up to an EndLoop marker, or the end of the item if no EndLoop is found. If 0 is given, or no number specified, it will loop continually until you press Esc or left-click the KeyText icon in the tray. Note that if a variable is used in place of ***n*** then if the variable is zero this indicates that the whole loop should be skipped; for example {Loop {#a}}…{Endloop} would be skipped if variable a was zero.
Note that loops may be "nested" up to 5 deep, but you must make sure that each Loop… EndLoop pair is enclosed within another Loop… EndLoop pair.

Menu *menu1|menu2[|menu3...]*

The ***menu1*** argument is the top-level menu, and ***menu2*** is the item to click - or ***menu2*** is a submenu and ***menu3*** is the item to click, etc., all enclosed in single or double quotes.
Causes KeyText to select the specified menu item. For example, {Menu "Data|Sort by|Date"} will select the Date item on the Sort by submenu of a Data menu.

Message [*seconds*[x]] [L]*title body*

Displays a message box with the specified ***title*** and ***body***, both being enclosed in single or double quotes. KeyText will display the message box and wait for you to press OK before continuing with the rest of the text item (if any). If the item with the message field is scheduled for a certain day or time, it will also display an "Edit/Repeat..." button. Press this, and you can choose to either repeat the item after a specified number of minutes (1-240), or edit the text or schedule for the item. While the message is displayed, KeyText enters pause mode, as indicated by the animated icon in the tray. Other KeyText functions - including scheduled items - are disabled until the message is dismissed. Note that this means only one message can ever be displayed at a time (by one instance of KeyText). If a scheduled event occurs while a message is displayed, it will not be actioned until that message is dismissed. For example, {Message "Alarm" "It is now {Time}"} will produce a box with the current time displayed; such a message could be combined with a Wave field in a scheduled item to produce an alarm-clock function.
If ***seconds*** is specified then the message box will automatically close after that number of seconds. The item will then continue as if the OK button had been clicked - unless **x** is also specified, in which case the item will abort as if Cancel had been clicked.
If present, **L** indicates that body should be left-aligned on the Ask dialog; otherwise it is centered.

Mouse *action*[[shift],[#]*nx*,*ny*[,[#]*ntx*,*nty*]]

Action can be one of the following:
L - left click

D - left double-click
L~ - left drag
R - right click
R~ - right drag
M - middle button click
N - no click (just move the pointer)
Restore - restore pointer to its position before the item was started (takes no further parameters)
Shift (if present) can be one of the following, and causes the stated shift key to be "held down" while the mouse action takes place:
S - shift key
C - ctrl key
SC - shift and ctrl key together
The **#** symbol, if present, indicates that the co-ordinates ***nx*** and ***ny*** are relative to the current foreground application; the co-ordinates in this case may be negative if the point to be clicked is in the title or menu area of the window, or on the left border. If # is not present the co-ordinates are screen co-ordinates, with 0,0 as top left.
nx and ***ny*** may be **-**, as in {Mouse D,-,-}, which indicates that the mouse pointer should not be moved before the double-click, or other action, is made.
If a drag action (drag and drop) is selected, ***ntx*** and ***nty*** give the point at which the button should be released.
Relative co-ordinates are useful if the location on screen of the target window is unpredictable; if the point to click is consistent in relation to the target window, then relative co-ordinates should hit the right spot. On the other hand, if a certain dialog always appears centre screen, then screen co-ordinates should be used.
Note that with a drag operation it is permissible to have one set of co-ordinates relative, and the other screen.

Pause [*seconds*]

When KeyText reaches this point in an item, it will pause for the specified number of ***seconds***, being a number from 0 to 1000000 which can include one digit after the decimal point. A pause state is indicated by an animated icon in the tray. If the argument is zero or omitted KeyText enters an indefinite pause - it waits for you to either right-click the KeyText icon in the tray to resume, or left-click it to abort. This is the same as a pause invoked manually by right-clicking the KeyText icon in the tray, or pressing the Pause or f12 key, while an item is being typed. If a time is specified, KeyText will wait that length of time, then continue. During that pause, if you right-click the icon in the tray KeyText will go into indefinite pause mode as above - right-click again to resume. Again, a left-click aborts. Note that pressing the Pause, f12 or Esc keys have the same effect as a right or left click on the icon respectively. For example {Pause 2.5} will pause for 2.5 seconds.

Speed [*n*]

Changes the speed of simulated typing for all or part of an item set to Action: Macro. If ***n*** is 0 (zero) or omitted, speed is set to the current default as set in the Options dialog. Otherwise choose a number from 1 to 10, corresponding to the slider position in the same dialog, where 1 is slowest and 10 is fastest.

Wait [*seconds*] [-][$][*][!]*title* [*static*] [if...]

title is all or part of the title of a window (enclosed in single or double quotes) which KeyText will wait for before continuing; if it has not appeared within ***seconds*** seconds (0.1 - 1000000), the item will cancel. If ***** is specified, then KeyText will check all windows for title, otherwise only the active or foreground window is monitored. If **!** is specified, KeyText will look for an exact match between ***title*** and the window title (but not case specific); otherwise it will match any window which includes the text in ***title***.
Optionally ***static*** text may be specified to further identify the window; note that this must be fixed text, as in a dialog label or button, and not dynamic text as in an edit or browser window.
If **$** is specified, then instead of waiting for a window with ***title*** KeyText will wait for a file to exist or be created whose path and filename is ***title***. Wildcards can be used. For example {Wait 0.0 $"c:\test.txt"} will wait indefinitely (0 seconds puts no time limit on the wait) until it detects c:\test.txt, when it will continue; to cancel press Esc or left-click the KeyText icon. {Wait 600.0 $"c:\data\*.doc"} will wait up to 10 minutes for any file with the extension .doc in the path c:\data.
If **-** (hyphen) is specified then KeyText will wait for the specified ***title*** in the status bar of the foreground window. Note that this will only work if the foreground application uses the standard windows “status bar” common

control. It can therefore be used with Internet Explorer to wait for “Done”, as in {Wait 60.0 -"Done"}. This would continue as soon as “Done” appeared in the status bar - but cancel if not found within 60 seconds.
When KeyText reaches a Wait field, it will enter a pause state.
If there is no **if…** condition specified, or there is one of the following four conditions (note that the first is the same as having no 0D6AF878C) then KeyText will continue if the specified window appears, but if *not found* it will cancel, link to (start) another item, call another item or goto a label, and then continue.
if not found cancel
if not found link A
if not found call A
if not found goto label
If one of the following four **if…** conditions is used, then KeyText will continue if the specified window *does not* appear, but if *found* it will cancel, link to (start) another item, call another item or goto a label, and then continue.
if found cancel
if found link A
if found call A
if found goto label
Note that while in the wait state, if you right-click the KeyText icon in the tray or press the Pause or f12 key - the KeyText item continues. If you left-click the KeyText icon in the tray or press the Esc key - the KeyText item cancels.

Wave *wavfile*

Plays the specified ***wavfile***, being a file with the .wav extension.

The following key combination fields may also be used with Action: Macro

{Application} {Back} {Cancel} {Caps Lock} {Clear} {Delete} {Down} {End} {Esc} {F1} {F2} {F3} {F4} {F5} {F6} {F7} {F8} {F9} {F10} {F11} {F12} {F13} {F14} {F15} {F16} {Home} {Insert} {Left} {Num *} {Num +} {Num -} {Num .} {Num /} {Num 0} {Num 1} {Num 2} {Num 3} {Num 4} {Num 5} {Num 6} {Num 7} {Num 8} {Num 9} {Num Lock} {Page Down} {Page Up} {Pause} {Print Screen} {Return} {Right} {Scroll Lock} {Space} {Tab} {Up} {Shift} {Alt} {Ctrl} {Win}

Note that the Application key is the key which, when pressed, normally brings up an application's pop-up menus and help topics; the Win key is the Windows logo key.

The above can be combined with the shift keys Shift, Ctrl, Alt and Win. For example, {Ctrl A}, {Ctrl-Shift F1}. Additionally, a number may be placed at the end of the field to indicate how many times the character (combination) should be repeated. For example: {Shift Tab 6}.

Integer Variables

Variables can be used in both Action: Macro and Paste items. 26 integer variables are available in each item, local to that item, and named a,b,c….x,y,z. They can be associated with global variables which are retained from one session to another, and available to all KeyText files. Default value of a variable is 0.

#*a*=*n*

This assigns the number ***n*** to integer ***a***, where n is a number in the range -999,999,999 to 999,999,999, or another variable. For example, after:
{#a=10}{#b=a} both variables a and b will equal 10.
Note that field nesting enables string variables, clipboard etc to be used for the ***n***. For example:
{seven="7"}{#a={seven}}{#a}
will output 7.

#*a*=DaysInMonth *m*,*y*

Sets variable ***a*** to the number of days in month ***m*** of year ***y***, where ***m*** and ***y*** are variables or numbers.

#*a*=Input *title question*
This produces a dialog box into which you can type a number in the range -999,999,999 to 999,999,999, which will be assigned to variable ***a***. The dialog will have ***title*** in the title bar and ***question*** as the text. For example: {#i=Input "Input dialog title" "Enter value for i"} sets i to the value you input.

#*a*=MouseX[#]
#*a*=MouseY[#]
Sets variable a to the current x or y screen co-ordinates of the mouse cursor, where 0,0 is the top left of the screen.
If the modifier **#** is added, co-ordinates are relative to the window under the mouse cursor, where 0,0 is the top left of the window area immediately below the title bar.

#*a*+=n
#*a*-=n
#*a=n**
#*a*/=n
#*a*%=n
These fields add, subtract, multiply, divide, or calculate the remainder (mod) using variable ***a*** and variable or number ***n*** and assign the result to variable ***a***. For example, if variable e has the value 16 and f has the value 4: After {#e+=f}, e will equal 20. After {#e-=10} e will equal 6. After {#e*=f} e will equal 64. After {#e/=8} e will equal 2, and after {#e%=12} e will equal 4.

#a [**"*n*/0*n*"**]
This field outputs (types) the value of variable ***a***. If ***n*** is specified, this indicates the minimum width of the output text; if **0** is specified before it, leading zeros will be added, otherwise spaces. For example, if variable x has a value of 100, {#x} will type 100, {#x "5"} will type   100 (2 spaces before the 1), and {#x "05"} will type 00100.
Note that this field can be embedded inside other fields. For example if variable x is 100, then {Pause {#x}} will start a 100 second pause, or {Loop {#x}} will begin a loop to be run 100 times.

#a dateformat
This allows the integer ***a*** to be output as part of a date. ***Dateformat*** should be surrounded in single or double quotes, and from the following list:

y, M, d, h, H, m, s or t	Numbers with no leading zeros for single-digit years, Months, days, hours (12), Hours (24), minutes, seconds, or A/P
D	Similar to d above, but gives day of month as an ordinal: 1st, 2nd, 3rd etc.
yy, MM, dd, hh, HH, mm, ss or tt	Numbers with leading zeros for single-digit years, months, days, hours (12-hour clock), Hours (24-hour clock), minutes, seconds, or AM/PM
MMM or ddd	Three-letter abbreviations for the Month or day
yyyy, MMMM or dddd	Full name of the year, Month or day

For example if variable d is 6, {#d "dddd"} will output Saturday (or the equivalent in your own language).
Note that if d is 13, 20 etc, it will also output Saturday. DaysInMonth (above) can also be used in date manipulation .

#a@param
This field associates variable ***a*** with a time parameter which can be one of the following;
year, **month**, **dayofweek** (0-6 for Sunday-Saturday), **dayofyear**, **day**, **hour12**, **hour**, **minute**, **second**.
From that point on in an item, that variable will output the current specified time value.
Note that if such a variable is subsequently assigned another value, it will no longer return the time value when used in that item - it will become an ordinary variable.

For example, after {#d@day} variable d will give the current day of the month.

#*a*@rand *n*
This field will cause variable a to output random numbers in the range 0 to n-1. For example:
{#r@rand 10}{Loop 10}{#r "3"}{Endloop}
might produce: 4 5 9 6 7 0 2 1 3 6
Note that if such a variable is subsequently assigned another value, it will no longer return random numbers when used in that item - it will become an ordinary variable.

#a@*name*
This field will associate variable ***a*** with global variable name (which should be wrapped in single or double quotes). If the global variable does not exist it is created and assigned the current value of ***a***. If it does exist, ***a*** is assigned the current value of the global variable name.
Note that when an item with such a field finishes, the global variable is updated to the new value of ***a***.
For example, if MyCounter (see Manage global variables) is currently 99, after an item with:
{#c@"MyCounter"}{#c+=1}
MyCounter will equal 100.

Note that "if" fields using integer variables can only be used in Action:Macro items; see above.

String Variables

String variable names can be a combination of letters, numbers, space and other characters, but not ' " = %or {} (or whichever field delimiters you are using). A minimum of 2 characters is required (and trailing spaces are ignored). They should not start with KeyText reserved words such as Pause, Loop, Shift, Input etc. Global string variables can be managed in the Manage variables section of the KeyText Organizer.

String[@]=***value***
Assigns ***value***, which should be wrapped in single or double quotes, to the string variable ***String***. If @ is added, the variable becomes global, and its current value can be accessed from any item. Note that field nesting allows, for example:
{Clip1="{Clipboard "^\(.\).*" "\u\1"}"}
which would set the variable Clip1 to the first text character on the clipboard, in upper case.

String**[**@**]=Input**[***char***] [**L**][**^**]***title question***
This produces a dialog box into which you can type a new value for the string variable ***String***. If the @ is included, the variable becomes global, and can be accessed from any item.
If a single character ***char*** is specified, this character will appear instead of actual characters typed when the Input dialog is displayed.
If present, **L** indicates that question should be left-aligned on the Ask dialog; otherwise it is centered.
If present, **^** indicates that the Return key will "OK" the input dialog; otherwise the Return key will start a new line in the response text.
Note: this is similar to the Ask field above. However, the Input dialog is displayed when it is reached in the item, whereas the Ask dialog is displayed when the item starts. The Input method is not suitable for an item set to Action:Paste.

Ask[***char***] [**L**][**^**][**-**][**%**]***name***[@] ***question***
See Ask field above for more information.

String [**%**][[**d**]***expression replacement***]
This outputs the current value of string variable ***String***. If **%** is used it will be URL encoded (eg, SPACE character becomes %20).
The optional ***expression replacement*** allows a regular expression to be defined (within single or double quotes), with a corresponding replacement expression (within single or double quotes). This gives the possibility of extracting and manipulating text from the ask field result, ready for output. The **d**, if present, will display a regular expression debug dialog showing error and other regular expression information.

Output of the string variable is normally typed or pasted, but it can be "nested" in another field; for example:
{Message "Output" "Variable Name currently contains {Name}"}

Note that "if" fields using string variables can only be used in Action:Macro items; see above.

4.2 Regular Expressions

Regular Expressions can be used to match text on the clipboard, current selection, string variable etc, as set up in the Insert field wizard page: If clipboard / selection / window title / exe name / string variable (Ask result) = or contains.

KeyText also supports replacement expressions, based on text in the current selection, clipboard, string variable, etc. See below.

The following Regular Expression operators are available.

Operator:	**Description:**
.	Any single character. Example: h.t matches hat, hit, hot and hut.
[ ]	Any one of the characters in the brackets, or any of a range of characters separated by a hyphen (-), or a character class operator (see below). To match a hyphen (-) it must be either the first or last character within the brackets. Examples: h[aeiou][a-z] matches hat, hip, hit, hop, and hut; [A-Za-z] matches any single letter; x[0-9-] matches x0, x1, …, x9, x-.
[^]	Any characters except for those after the caret "^". Example: h[^u]t matches hat, hit, and hot, but not hut.
^	Anchors the search to the start of the text (Selection, Clipboard etc). See note below.
$	Anchors the search to the end of the text (Selection, Clipboard etc). See note below.
\<	The start of a word.
\>	The end of a word.
\n	New line character. *Cannot be followed by '?', '+', '*' or {}.*
\t	The tab character.
\q	The single quote character (which cannot be used within a regular expression if the expression is delimited by single quotes).
\d	The double quote character (which cannot be used within a regular expression if the expression is delimited by double quotes).
\xdd	"dd" is the two-digit hexadecimal code for any character.
\(\)	Groups characters together as an expression, to change the scope of other operators. Examples: a\(big \)? dog matches a dog and a big dog; p\(rin|oin\)t matches print and point. Also gives a tagged expression to use in a replacement expression, numbered according to its order in the regular expression.
\{count\}	Matches the specified number of the preceding character. Example: ho\{2\}p matches hoop, but not hop. *Only supported after normal characters, and operators '.' and ']'.*
\{min,\}	Matches at least the specified number of the preceding character. Example: ho\{1,\}p matches hop and hoop, but not hp. *Only supported after normal characters, and operators '.' and ']'.*
\{min,max\}	Matches between min and max of the preceding character. Example: ho\{1,2\}p matches hop and hoop, but not hp or hooop. *Only supported after normal characters, and operators '.' and ']'.*
*	Matches zero or more of the preceding characters or expressions. Example: ho*p matches hp, hop and hoop.

Operator:	Description:
?	Matches zero or one of the preceding characters or expressions. Example: ho?p matches hp, and hop, but not hoop.
+	Matches one or more of the preceding characters or expressions. Example: ho+p matches hop, and hoop, but not hp.
|	Matches either the expression to its left or its right. Example: hop|hoop matches hop, or hoop.
\	"Escapes" the special meaning of the above expressions, so that they can be matched as literal characters. Hence, to match a literal "\", you must use "\\". Example: \< matches the start of a word, but \\< matches "\<".

Note that the start and end anchor characters ^ and $ are required if you want to match the whole of the text contents. For example, the regular expression h[aio]t would give a match on photograph, whereas ^h[aio]t$ would only give a match if the text was hat, hit or hot.

Similarly, if only literal characters are used in the regular expression:
red
this would match the exact text: red, and also match: the red ball
Using the anchor characters ^ and $ for beginning and end allow you to specify an exact match;
^ball$ would only give a match with: ball.

KeyText supports the following expressions within square brackets; note that this means double square brackets are required. For example:
[[:alpha:]]+ will match one or more letters. [[:upper:][:digit:]] will match a number or upper case letter.

Expression:	Description:
[:alpha:]	Any letter.
[:lower:]	Any lower case letter.
[:upper:]	Any upper case letter.
[:alnum:]	Any digit or letter.
[:digit:]	Any digit.
[:xdigit:]	Any hexadecimal digit (0-9, a-f or A-F).
[:blank:]	Space or tab.
[:punct:]	Anything that is not a control or alphanumeric character.
[:word:]	Letters, hyphens and apostrophes.

Replacement Expressions
The fields {Clipboard}, {Selection}, {Include...} and string variables (Ask field results) can optionally include a regular and replacement expression, to allow manipulation of the source text. This can be set up in the Include string variable, clipboard, selection etc wizard.

For example: {Selection "\(^[[:word:]]+\) \([[:word:]]+$\)" "First name is \1\nSurname is \U\2"}
would, if John Smith was selected, output:
First name is John
Surname is SMITH

If a replacement expression does not give the result you expect, try regular expression **debug** mode, as set by the **d** in this example:
{Selection d"\(^[[:word:]]+\) \([[:word:]]+$\)" "First name is \1\nSurname is \U\2"}

When KeyText reaches the field, it will display a dialog showing debug/error information.

Here are the expressions which can be used in the replacement expression; everything else is output as it is.

Expression:	**Definition:**
&	Substitute the text matching the entire search pattern.
\0 to \9	Substitute the text matching tagged expression 0 through 9. \0 is equivalent to &.
\n	Include a newline.
\t	Include a tab.
\xdd	"dd" is the two-digit hexadecimal code for any character.
\u	Force the next substituted character to be in upper case.
\l	Force the next substituted character to be in lower case.
\U	Force all subsequent substituted characters to be in upper case.
\L	Force all subsequent substituted characters to be in lower case.
\E or \e	Turns off previous \U or \L.

4.3 Command Line

If you run KeyText from a command line or in a batch file, you can specify which .kt3 file to open and/or a hotkey to use for a keyboard shortcut - or an item to start automatically.

If you simply specify the exe file, the last used .kt3 file will be opened, as in:
c:\KeyText\KeyText
In most cases this would open the file KeyText.kt3.

Or you can give the .kt3 filename and let Windows file associations do the work, as in:
c:\KeyText\Personal.kt3
which would run Personal.kt3.

If using the first method above, you can optionally add parameters to start a specific item within a .kt3 file. For example:
c:\KeyText\KeyText /s2A KeyText.kt3
will automatically launch item 2A in KeyText.kt3.

The similar:
c:\KeyText\KeyText /x2A KeyText.kt3
will do the same - and then exit from KeyText when 2A is finished.

The full syntax to use is:
KeyText [***/cdata***] [***filename***] [***#password***]

Where **c** is s, x or h.

s runs KeyText with filename, runs the item specified as ***data*** (for example, /s2A) and continues running.

x does the same, but KeyText exits after the item is run (unless filename was already running), in which case it continues to run.

h sets a hotkey to open the left-click menu of filename, in which case ***data*** must be from 2 to 5 characters. The first gives the character to press, then a combination of the letters c (Ctrl), a (Alt), s (Shift) and w (Windows logo key). The hotkey rules apply; if no shift keys are given or "s" alone is given, Ctrl is added.

The h parameter is available mainly to give backwards compatibility for TrayText users; KeyText will normally use the hotkey specified in the Options dialog; this is specific to each KeyText data file (.kt3).

If a ***filename*** is specified, this overrides the default .kt3 file loaded. If the .kt3 file is password protected, the ***password*** can be specified in the command line.

For example:
c:\KeyText\KeyText /hpsc personal.kt3
will run KeyText from the KeyText folder, make Ctrl-Shift-P the hotkey, and load personal.kt3. The case of the characters given is not significant.
c:\KeyText\KeyText.exe #abc123
will run the default (last used) .kt3 file, which is protected with the password abc123.

A command line can also be used to enter KeyText's unlock code, as follows:
KeyText /u *name code*
Where ***name*** is the name, wrapped in single or double quotes, and ***code*** is the supplied unlock code - again wrapped in single or double quotes.

Part V

5 Technical Support

5.1 Answers to Common Questions

Q. Why does the speed of "typing" sometimes vary?
A. You may notice the speed at which text is being typed slow down or speed up; this is probably due to your system or word-processor being busy doing other things. KeyText can only type the text as fast as your system can deal with it, and this may vary even within one "type" session. At higher typing speeds KeyText pauses regularly to let the receiving program catch up with displaying the text; this may give a stop-start effect, but is normal behavior.

Q. When Auto-add is on, why is text sometimes added by KeyText when I haven't copied text to the clipboard?
A. Some programs copy text to the clipboard without telling you! KeyText has no way of detecting whether text has been added to the clipboard "manually" by you, or "on the fly" by another program; in both cases KeyText will add copied text to the Auto-add item. It has been noted that Microsoft® Word 97 sometimes does this.

Q. Why do I get unexpected results if the open brace character { is in my text item?
A. KeyText assumes that the open brace starts a special field - an instruction to click a button, display a message, etc. If you want the character itself, enter 2 together - {{; when typing etc., KeyText will give one. This does not apply to the close brace, which is treated normally. If you use the { character often, consider changing the field delimiters to [...] or <...> in global settings.
You may also notice strange things if you have one character enclosed in braces. For example, {+} when typed will produce = (the equals sign). This is because KeyText interprets {+} as a field containing a combination of keys - and the keys are treated as their unshifted equivalent: =. If you use the Insert Field Wizard, it would produce {Shift =} if you wanted the + sign, that is, the shift key required, followed by the unshifted character. Of course you would normally simply have the + sign in your text. If what you want is {+} then you must enter: {{+}. The same effect will be noticed if you have for example, {Y}; it will appear as y - there being no shift key mentioned, KeyText assumes the unshifted character is required.

Q. I have set up a reminder message - why does it not appear at the scheduled time?
A. This may be because you do not have "Macro" set as the action for the scheduled item. Fields like "Message", "Sound" and others will only work if the action is "Macro" - or if it is "Default" and the default action is "Macro". When you use the Insert Field Wizard, note carefully what actions can be used with the different fields available.

Q. My text contains fields; why does it sometimes stop unexpectedly?
A. Stopping unexpectedly may indicate that KeyText has not been able to find what you wanted; for example, if you have a field {Click "OK"} and KeyText cannot find a button with the text OK in the foreground window, KeyText cannot proceed and will therefore stop. If there is an OK button, it may be that the program concerned has "painted" the text on the button rather than sent it to the button as ordinary text. This can apply to menu items also, and KeyText cannot find text painted in this way. Look at any underlined characters on the button or menu - you may be able to get KeyText to type that to click it; or experiment with the [Tab] and [Return] keys.

Q. Why does (e) appear against some of the items in the left-click menu popups?
A. This is short for (empty) - indicating that a particular item has no text.

Q. Why do the popup menus for the KeyText left-click menu sometimes appear at the far left of the screen?
A. This may happen if you are using a screen resolution of 640 x 480, and indicates that Windows has

found that there is not enough room to display the popups in the normal position to the right. A workaround would be to identify which menu items are widest, and to specify a shorter menu text for them; this is because the width of the whole menu is related to the width of its widest item.

Q. I use Windows 98; why is the left-click menu misplaced after I enter text in item N?
A. If sufficient items are used, KeyText divides the left-click menu into 2 columns for convenience. Unfortunately there is a problem experienced by a very small number of Windows 98 users relating to certain display drivers which causes the menu to be misplaced. If you are one of very few users who experience this, go to customize settings and select "Always display left-click menu in one column" to fix it.

Q. I use KeyText to run a program, then type some characters. Why does it start typing too soon?
A. KeyText cannot tell when the program being run is ready to receive keyboard input, though it does its best to work this out. If it starts typing too soon, add a pause or a wait immediately after the "Run" field.

Q. Sometimes no text appears using "Macro", even though the KeyText icon indicates it is "typing" - what is wrong?
A. This may happen if keyboard focus changes to the Windows desktop or taskbar, possibly as a result of the mouse being used while KeyText is typing. Don't worry if this happens - simply press Esc to abort typing.

Q. In the "Insert custom date or time" dialog (Insert Field Wizard), why is the button beside AM/PM blank? Or can I have it with small letters?
A. This means that an AM/PM indicator has not been set on your system. Go to Control Panel and select Regional Settings; click the time tab and set it to what you require.
This also applies if KeyText displays, for example, AM and PM, and you would rather have am and pm.

Q. Why do hotkeys sometimes not appear on the menus, or appear grayed in the Edit text items window or Options dialog?
A. This indicates that the hotkey is already in use. For example, if you have 2 .kt3 files with the same left-click hotkey set, only one hotkey - the more recent - can be active and appear on the right-click menu. If you have more than one instance of KeyText running, subsequent instances cannot set a hotkey to an item if the same hotkey is already used by an earlier instance; in this case the hotkey will appear grayed in the Edit text items window. If the earlier instance is closed, the hotkey will become active for the later instance.

Q. Why do numbers appear when typing, which are not in the text item?
A. Probably because you are using a word-processor with automatic numbering of paragraphs. If your item has the number 1 at the beginning of a line, when a line break occurs the number 2 may be automatically inserted, etc. Either turn off this feature of your word-processor, or use Paste to insert the text item.

Q. Why does KeyText sometimes type characters when I don’t expect?
A. 1) You may have a scheduled item set. If you change a scheduled item, KeyText reminds you that a schedule is set; but if you ignore this and, for example, put ordinary text in a scheduled item, it is possible that when the scheduled event occurs KeyText will put the text in whatever application is running at the time.
2) If a scheduled item has a message field, it is recommended that no text follows that field in the item. If there is text, then when you click OK to dismiss the message it will be typed - which may not be your intention.

Q. When using the "Run" action, why does KeyText sometimes tell me a file is not found which I know exists?
A. This may happen if you have included a command-line parameter at the end of the line. If you want to specify a parameter or default directory, you must use a run field set up using the Insert Field wizard. This field can be used with the Macro or Run action.

Q. I have KeyText set to start every time Windows starts. Why does it not always load the same .kt3 (data) file?
A. When KeyText starts, it loads the last .kt3 file to be closed. If you work with more than one .kt3 file, this may vary depending on which you closed last. If you want to start a particular .kt3 file whenever Windows starts, put a shortcut in your Startup menu to the .kt3 file itself, rather than KeyText.exe. Instructions for this can be found in What happens when KeyText starts?

Q. I have specified text in Advanced schedule settings to help identify a dialog better. Why does KeyText not find it?
A. If the window title is not sufficient to accurately identify the window/dialog you want for an "On window" schedule, you can enter in the Advanced schedule settings any fixed (static) text in the window which will identify it better. If KeyText fails to find text which you know is there, it could be because the text is not static, and no other application - including KeyText - can gain access to it; only fixed static text should be specified. This may apply, for example, to text in a browser main window, to any editable window, or to a window where text is placed in a non-standard way.

Q. Why does KeyText not remember my settings?
A. This indicates that you have restricted permissions for your registry, where KeyText stores its settings. KeyText firstly tries to store its configuration information in HKEY_LOCAL_MACHINE, but if it cannot, it then tries HKEY_CURRENT_USER. If it is denied permission to write there also, it will not be able to store configuration information; this may happen, for example, if you are logged on as guest. Log on as Administrator - or contact your system administrator - to alter the security permissions on either of these registry areas.

Q. I have re-installed Windows (etc.), and no longer have my unlock code. KeyText has gone back to "Evaluation Version, and I can only see items A to U - are the rest lost?
A. No - not lost. Although you cannot see them, all your texts are retained. Send an e-mail to support@mjmsoft.com for a duplicate unlock code, enter it, and they will be restored.

5.2 Windows XP/Vista/7

The part of the taskbar previously known as the tray, or system tray, in earlier versions of Windows is called the Notification area in Windows XP and Vista. And new features help to keep this area less cluttered than in previous versions.

There are two side effects of this which affect KeyText users.

The first is that the "Shift icon" moves the icon *away* from the time rather than towards it as in previous versions of Windows.

The second is that the KeyText icon may disappear. This is because of the "Hide when inactive" feature; if Windows decides that an icon is inactive, it will hide it.

To prevent the KeyText icon disappearing in this way, or if has already been hidden, right click a blank area of the Taskbar and select Properties. Under Notification area, click Customize. Alternatively, Right click a blank area of the Notifications Area and select Customize Notifications. Select the KeyText icon, then click the down arrow next to it and select "Always show".

If running on **Windows 7**, on first use KeyText displays a dialog advising the user that the KeyText icon in the taskbar Notification Area cannot be shown automatically. The dialog contains a button that opens the Windows Notification Area Settings, and allows "Show icon and notifications" to be chosen for KeyText.

5.3 Known Limitations

Combo boxes and non-standard text boxes
KeyText does all it can to accurately identify the exact place at which typing or pasting should begin. However, clicking the KeyText icon does move the keyboard focus away from the active window. KeyText returns focus straight away, but in the case of combo boxes, when the focus is returned the whole contents of the combo box are selected - even if they weren't before. When KeyText starts to type, etc., it therefore deletes any (selected) text already there, rather than inserting it at the caret.

Certain non-standard text windows may show similar behavior. If you suspect KeyText is functioning incorrectly, try this. With the application in the foreground, place the caret at the required position, and then press a button on your Windows taskbar to change to another window. Then press the taskbar button to go back to the first program, and observe what has happened to the caret. Normally it goes back to the same place - in which case KeyText will function correctly. If the caret has disappeared, or all the text is now selected, it is possible that KeyText - including the Keypad - will have problems.

One way round this is to set a hotkey for the KeyText text item; pressing a hotkey does not change keyboard focus in the same way as clicking the KeyText icon.

This behavior may also affect copying or adding text to a KeyText item using the left-click menu popups.

Non-standard controls
Most applications construct dialogs, buttons, menu items etc. using standard Windows functions for sending text to the relevant dialog/button/menu. However, some applications "draw" or "paint" text directly onto the dialog etc. In such situations, KeyText may not be able to locate the button or menu item specified in a "Click" or "Menu" field. This may sometimes also apply to text specified in Advanced "On Window" Settings, and KeyText may therefore fail to respond to a dialog, even if text specified there appears on the dialog. This behavior has been observed in dialogs generated by Microsoft Office 97® applications. In most cases an alternative approach can be devised which gives the required result. "Microsoft" and "Office 97" are registered trademarks of the Microsoft Corporation.

MS-DOS window
When typing to an MS-DOS window, KeyText may sometimes "stick" - and reset itself after 10 seconds or so. Altering the typing speed in the Options dialog normally fixes this, but note that the problem may be to do with the MS-DOS program itself. KeyText has been tested extensively with the MS-DOS Editor, and functions correctly.

Repeated character after pause
When "typing", KeyText does all it can to track how many characters have been received by the destination program - so that, if KeyText is paused by pressing Pause or f12, or right-clicking the icon in the tray, it knows where to resume. However, you may occasionally find that on resuming (press Pause or right-click again), the character before the pause is repeated.

Index