

AutoKey

User's Manual

Table Of Contents

Introducing AutoKey	3	
How to Register		4
License Agreement	4	
Disclaimer	4	
Manual Conventions	5	
Technical Support	5	
Package Contents	5	
What You Need		5
Installation Instructions		6
Using AutoKey		7
The Main Window	7	
The Turn Off (or Turn On) Button	7	
The Help Button	7	
The About Button	7	
The Quit Button	7	
The Application List		7
The Add... Button	8	
The Configure...Button	8	
The Remove Button		8
Adding an Application		8
Removing an Application		9
Setting Up an Application		9
Setting Up a Menu Command		9
Setting Up a Menu Accelerator	10	
Entering the Interval		11
Special Conditions		11
Deferring Autokeys	11	
Title Screening		12
Other Streamlining	12	
The OK Button	12	
The Cancel Button		12
Turn It Loose!	12	
Using Help	13	
Usage Notes	14	
DOS Applications	14	
Screen Blankers		14
What if I'm in the middle of something and AutoKey tries to send a key?		14
But my program already has an auto-save!	14	
Where does AutoKey store its data?	15	
Future Enhancements		16
Glossary	17	
§		

Introducing AutoKey

Now you can LAUGH at power failures, snicker at UAE's. Or take the buffoon who kicked your power cord to coffee.

And even when USER ERROR rears its ugly head, potentially destroying four hours of work, you can breathe a sigh of relief...

Because you had the foresight to run AutoKey.

AutoKey protects the data in every Windows application you're working with, by instructing the other applications to save their data. Even if you're frequently switching between multiple applications, the data in ALL your Windows applications is protected.

To use AutoKey, you just set up the applications you wish to protect, minimize AutoKey, and forget it. At the pre-defined intervals, AutoKey examines the Windows desktop, and if any of the protected applications are active, sends the keystrokes you request to all the applications, just as if you typed the keys yourself. AutoKey even protects the data in multiple copies of a program.

You can even tell AutoKey to send non-saving keystrokes, causing an application to perform special functions at the intervals you set up. For example, you might want to get stock quotes automatically throughout the day, or download hourly weather information for your area.

Other features of AutoKey include:

- * Keystroke control via menu commands, or through keyboard accelerators, or both.
- * Execution of nested popup menu commands, up to five levels of nesting.
- * Time, keystroke or mouse click intervals.
- * The ability to "screen out" applications if they contain untitled works.
- * A handy notification of pending autokeys.
- * A defer keystroke, which allows you to intercept and skip a pending autokey. This keystroke can be unique for each application under AutoKey's control.

This section of the [AutoKey User's Manual](#) contains some introductory information, a description of the conventions used in this document and AutoKey itself, an inventory of the AutoKey package, a list of what you need to run AutoKey and installation instructions.

How to Register

To register your copy of AutoKey, simply print the file REGISTER.TXT using your favorite word processor or the DOS print command, fill it out, and send it and \$39.95 (US dollars) to:

Lynn Wallace
Simple Software
PO Box 1465
Salt Lake City, UT 84110-1465

Registered users receive the latest version of AutoKey (without the registration reminder), a printed user manual, free updates for a year after registration, and technical support for as long as you use AutoKey. (Technical support for non-registered users is available for installation and evaluation purposes only.)

License Agreement

Non-registered users are granted the use of AutoKey for a trial period of thirty days. If, after thirty days, you are still using AutoKey and have not registered your copy, you must either register your copy immediately or immediately cease to use AutoKey.

Non-registered commercial use of AutoKey is strictly prohibited.

A single registered copy can be used by one individual at any given time. Simultaneous use by more than one individual requires multiple registrations. Volume discounts and site licenses are available.

Permission is granted to copy and distribute AutoKey in archive form as long as the original archive remains unaltered, and as long as no fee is charged beyond the cost of the media.

Disclaimer

The sad state of legal affairs requires that in order for you to use AutoKey, you must accept the following disclaimer:

AutoKey is supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of AutoKey.

Manual Conventions

Phrases which appear in AutoKey windows are italicized when they're referenced in this document.

Most controls in AutoKey have keyboard mnemonics, which allow you to "click" each control with a keystroke sequence. Pressing Alt and the character underlined on the control's title will click the control. These mnemonics are generally not described in this manual.

Technical Support

If you are a registered user, you can get free answers or educated guesses to your questions (assuming they apply to AutoKey!) by contacting:

Lynn Wallace
Simple Software
PO Box 1465
Salt Lake City, UT 84110-1465
CompuServe: 70242,101

Sorry, no phone support is available at this time without first contact via mail or CompuServe.

Package Contents

The distribution archive of AutoKey contains the following files:

AUTOKEY.EXE	The AutoKey executable.
AUTOLIB.DLL	The run-time support Dynamic Link Library.
AUTOKEY.HLP	The help file.
REGISTER.TXT	The registration form.
README.TXT	The generic info file.
MANUAL.WRI	This manual.

What You Need

AutoKey requires an IBM PC or compatible with an 80286, 80386 or 80486 microprocessor, at least 1 MB of RAM (2 MB or more suggested) and Microsoft Windows 3.0 or newer. A mouse is recommended.

AutoKey doesn't support Windows on an 8086 machine, or Windows' real mode on any other machine. Real mode doesn't allow multi-tasking, so AutoKey offers nothing to real mode users.

In addition to the above physical items, you should have a working knowledge of Microsoft Windows. AutoKey looks and feels like any standard Windows application, and what works in most other applications works in AutoKey. This manual doesn't go into detail about standard Windows functions.

Installation Instructions

If you are installing AutoKey from a floppy disk:

Insert the floppy in the drive.
Choose File/Run from Program Manager.
Enter <drive>:SETUP.
Follow the instructions as they are displayed.

If you are installing AutoKey from a file archive:

Copy the archive file to the desired disk and directory.
Execute the archive file:
From DOS C:\AUTOKEY>AUTOKEY.EXE
From Program Manager File/Run C:\AUTOKEY\AUTOKEY.EXE

If desired, add AutoKey to a Program Manager group with:

File/New [Program Item] C:\AUTOKEY\AUTOKEY.EXE

or the actual drive and directory where you placed AutoKey.

At this point using both methods, you are ready to run AutoKey. Either use File/Run or double-click on its icon.

You will probably want to edit your WIN.INI file (in the Windows directory) to cause Windows to load AutoKey whenever Windows starts. Near the top of WIN.INI you should find something similar to:

```
[windows]
load= <other programs>
run= <other programs>
```

Modify the "run=" line as follows:

```
run= <other programs> C:\AUTOKEY\AUTOKEY.EXE
```

(Be sure to specify the correct disk and directory if you installed AutoKey in a different location on your disk.)

AutoKey normally runs iconized, with only the registration reminder window displayed for a short period. However, during its initial execution, or whenever you've deleted or renamed the \AUTOKEY\AUTOKEY.CFG file, AutoKey will open its Main Window when running, and the application list will contain only the pre-configured applications. So, once you've performed these setup actions, AutoKey should be doing its thing quite unobtrusively every time you use Windows.

Using AutoKey

Run AutoKey using your preferred Windows method. The first time you run it, you'll see the main window. Access on-line help via the Help button.

The Main Window

The main window includes a system menu box, a minimize box, a window frame, the *Turn Off/On* button, the *Help* button, the *About* button, the *Quit* button, the application list, the *Add...* button, the *Configure...* button and the *Remove* button.

The system menu box and minimize box behave as you expect. But the window frame is for display purposes only, to make the AutoKey window stand out from other items on the desktop. You can't use it to re-size the window. The remaining items in the main window are discussed below.

The Turn Off (or Turn On) Button

Clicking this button toggle's AutoKey's operation status off or on. AutoKey doesn't really do anything until you minimize it into an icon. Whenever AutoKey is displayed as a window, all you can do is configure it.

The Help Button

As you might expect, clicking this button provides you with help. Pressing F1 produces the same effect.

The About Button

Clicking this button creates a dialog which contains background information and instructions for registering your copy of AutoKey. (These instructions can also be found in Help.)

The Quit Button

Clicking this button exits AutoKey. Alt-F4 produces the same effect, as does selecting Close from the system menu.

The Application List

The application list contains all the applications AutoKey will affect. For each application, it shows the intervals between autokeys in time, keystrokes and mouse clicks. If the list contains more applications than can fit in its display, a vertical scroll bar is provided.

One of the applications in the list is highlighted. This is the currently selected application, and is the one affected by the *Configure...* and *Remove* buttons. You can select a different application by clicking in the list or by moving the highlight bar with

the arrow keys.

The Add... Button

Click this button to add a new application to AutoKey's control. You can add more than one copy of an application to the list, complete with its own unique parameters. Clicking this button creates the Add dialog box, described later under "Adding an Application". The new application is added to the application list and highlighted when you're done.

AutoKey can control up to 64 applications.

The Configure...Button

This button allows you to change the autokey, intervals and special conditions for an application. Clicking this button creates the Configure dialog box, described under "Setting Up an Application".

The Remove Button

This button removes the current selected application from the list. You can't retrieve it; to restore it, you must *Add...* and *Configure...* it again.

Adding an Application

Clicking the *Add...* button on AutoKey's main window causes the Add dialog to appear.

This dialog box allows you to enter the application name as it will appear in the application list, and either the application's window title or its EXE file name. In addition, there are pushbuttons to OK or Cancel the entries.

There's nothing special about the *Application Name* box, just fill it as you desire, up to thirty characters ("Word for Windows," "Windows Write" or "WordPerfect" for example). But the *Window title* and *EXE file name* both need a little explanation.

Since many applications change the title of their window depending on their state, you can't specify every possible case. So the Window title box allows you to use the * (asterisk) character as a wildcard. The * can only be used at the end of a title string. For example, AutoKey comes set up for Microsoft Word for Windows, and Word's window title has been entered as "Microsoft Word*". This means that any window containing "Microsoft Word" followed by any other characters (or no other characters) will fall under AutoKey's control.

You may decide to enter the application's EXE file name instead of its window title. This is the full name, including the disk and directory path, of the application's executable file as it is stored on your system. This field is set to all capital letters. AutoKey searches for the file when you click OK and notifies you if it can't find the file. You can then re-check the file's name and location and re-enter it.

It takes AutoKey slightly less time to find an application by its window title than its

EXE file name, so you're mildly encouraged to enter the application's window title instead of its file name. Though AutoKey shouldn't create any noticeable processing overhead, entering a window title will speed it up a bit.

Clicking *OK* causes AutoKey to collect the information you've entered and perform any checks on it. The Add dialog box disappears and the new application is entered into the application list with the following default parameters:

Autokey: the File/Save menu command.
Time interval: 10 minutes.
Keystroke interval : 2000 keystrokes.
Mouse click interval: 500 clicks.
5-second notification: off.
Defer key: none.
Window title screening: none.
Background autokeys: off.
Iconized autokeys: on.

Of course, only the interval values appear in the application list. To see the other default values, click the *Configure...* button on the main window.

Clicking the *Cancel* button closes the Add dialog box and causes AutoKey to forget any data you've entered.

Removing an Application

To remove the current application from the list, click the *Remove* button on the main window. To restore the application to the list, you must add it again.

Setting Up an Application

When you click the *Configure...* button on the main window, the Configure dialog box appears, allowing you to view or change the current application's parameters.

There's a lot going on here, so we'll split it up.

Setting Up a Menu Command

Clicking the *Menu Command* button creates a new dialog box.

The *Menu* box lists the name of the menu containing the command you wish AutoKey to invoke for the application, for example File, Edit, Options, etc. The *Submenu/command* boxes contain the names of menus within the previous menu, or the command within a menu.

You can use the * character as a wildcard at the end of the menu items if you wish. Don't worry about including underlines in the menu items you enter here.

AutoKey comes set up for Microsoft Word for Windows, Windows Write and WordPerfect for Windows with the File/Save menu command, so you can get an idea of how to use this dialog by looking at the menu commands for those applications. For a more complex example, let's say you have an application with the following menu structure

+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

|File   |Edit   |Screen   |
+-----+-----+-----+-----+
|       |       |Calculations |
|       |       |Text       +-----+-----+
|       |       |Graphics   |Update   |Curves  |
|       |       +-----+Scale   |Edges   |
|       |       |Change Colors|Rotate  |
|       |       +-----+Faces   |
|       |       |Sprites   |
|       |       +-----+
+-----+-----+-----+-----+

```

and you want AutoKey to execute the Screen/Graphics/Update/Sprites command at every interval. You would set up the Menu Command dialog box as follows:

Menu:	Screen
Submenu/command:	Graphics
Submenu/command:	Update
Submenu/command:	Sprites
Submenu/command:	(empty)

The *OK* and *Cancel* buttons cause AutoKey to accept or ignore your changes, respectively. The *Test* button is useful for making sure your menu command works the way you expect. If the target application is active, clicking this button sends the menu command to the application. You can observe the results in the application window, provided it's visible, and confirm that AutoKey does what is intended.

Setting Up a Menu Accelerator

Clicking on the *Menu Accelerator* button in the Configure dialog creates a new dialog box.

This dialog allows you to specify a menu accelerator keystroke, or any other keystroke, for an application. Menu accelerators are menu commands assigned to short, quick keystrokes, and the keystrokes appear at the right of the menu command to which they apply.

The *Accelerator Keystroke* box works differently from standard Windows edit boxes. Only a single keystroke is allowed to fill the box at one time, and the box contains the name of the key. All keys available on your keyboard are useable in this box except Shift, Ctrl, Alt, Tab, Esc and Enter. Tab, Esc and Enter perform their standard Windows dialog functions: Tab moves the input focus to the next control (the *Clear* button in this case); Esc cancels the dialog; and Enter OK's the dialog. The Shift, Ctrl and Alt keys are ignored in the *Accelerator Keystroke* box, but you can combine them with the keystroke using the check boxes in the dialog.

If AutoKey can't determine the name of a key on your keyboard, "Unmapped" appears in the *Accelerator Keystroke* box when you enter the key in the box. The key should still work as expected.

The *Clear* button allows you to clear the menu accelerator. This button is here

because the Delete and Backspace keys are intercepted by the *Accelerator Keystroke* box, and thus can't be used to clear it. Clicking on *Clear* clears the *Shift*, *Ctrl* and *Alt* check boxes as well as the *Accelerator Keystroke* box.

The *Modifiers* check boxes allow you to enter the Alt, Shift and Ctrl keys as modifiers to the accelerator keystroke. If the menu accelerator you wish AutoKey to send to the controlled application involves pressing one (or more) of these three keys at the same time the accelerator key is pressed, click the appropriate box(es).

A note about modified accelerator keystrokes: In order to send a Shift, Alt or Ctrl keystroke to another application, AutoKey must "fool" Windows into acting as if the Shift, Alt and/or Ctrl key is pressed. This means that other applications "see" these keys being "pressed" also. So if you're in the middle of typing a line of text in a text editor, and AutoKey sends a menu accelerator to another, inactive application, the editor may be fooled into acting as if you've pressed Shift, Alt or Ctrl, and your text may be affected. The bottom line: use modified menu accelerators sparingly, or don't use the keyboard heavily while AutoKey may send a modified menu accelerator. Or you may set up an application which uses a modified menu accelerator to only receive its autokey when it's active. You do this by checking the *Don't send autokey if window is in background* box in the Configuration dialog. Unmodified menu accelerators (keystrokes that don't involve the Shift, Alt or Ctrl keys) should cause no problems.

The *OK* and *Cancel* buttons cause AutoKey to accept or ignore your changes, respectively. The *Test* button is useful for making sure your changes will work. If the target application is active, clicking this button sends the autokey to the application. You can observe the results in the application window, provided it's visible, and confirm that the keystroke does what is intended.

Entering the Interval

The *Invoke Autokey* section of the Configure dialog box allows you to enter the intervals at which AutoKey will send the autokey(s) to the controlled application.

You specify intervals in minutes (1-999), keystrokes (100-9999) and mouse clicks (50-9999). You can cancel an interval check by clearing the appropriate check box.

Special Conditions

The *Special* section of the Configure dialog allows you to streamline AutoKey so it runs with a minimum of intrusion.

Deferring Autokeys:

Checking the *Beep before sending the autokey* box makes AutoKey sound a mellow beep on the PC's speaker five seconds before it sends the autokey to the application. When you check this box, AutoKey instantly sounds the beep to let you know what to expect. The notification beep happens only before the autokey for the active application. This is useful in conjunction with the defer key, described next.

Checking the *Allow me to defer it by typing ... within 5 seconds* box enables the edit

box containing the defer keystroke. You enter the key you want to use to abort a pending autokey in this box, which functions just like the *Accelerator Keystroke* box described earlier: a single keystroke is displayed in the box, and all keystrokes except Esc, Tab, Enter, Shift, Ctrl and Alt are valid.

The *Modifiers* check boxes let you combine the Shift, Ctrl and Alt keys with the defer keystroke. You can mix them as desired. For example, if you want to be able to abort an application's autokey with Shift-Ctrl-Alt-Z (four keys close together on the keyboard, and unlikely to be tied to a function in any application), you'd press the Z key in the keystroke box, and click the *Alt*, *Shift* and *Ctrl* check boxes next to *Modifiers*. The defer key section of the dialog would look like this:

```

                                     +-----+
Allow me to defer it by typing |Z      |
within 5 seconds.             +-----+

Modifiers:   X Alt      X Shift X Control
```

Then, when you hear the beep indicating that AutoKey is about to send a key to the application you're working in, you can press Shift-Ctrl-Alt-Z, and AutoKey will ignore the application for this interval.

The defer key is only useful when combined with the notify beep, otherwise you won't know when an autokey is on the way. So if you turn off the notify beep, AutoKey clears the defer key section of the Configure dialog.

You can only defer the autokey for the current active application. You can't defer autokeys to applications that are open but sitting idle on the desktop.

Title Screening:

The *Don't send autokey if window title contains ...* box allows you to enter part of a window title that AutoKey looks for before it sends a key to the application. If the text appears anywhere in the application's window title, AutoKey won't send the key. Don't use wildcards here, just enter the text you want AutoKey to look for in the application's title.

If you have had problems with inadvertently killing untitled works that you later wanted to use by mistakenly exiting a program without saving the work (just as the author of AutoKey does!), you may wish to ignore this feature. Thus, when AutoKey sends a save command to an application, you'll probably be prompted for the file name, which may disrupt your work but helps keep you "honest".

As examples of this feature, Microsoft Word uses "Document1", "Document2", etc., as its generic title of documents that haven't been named. So AutoKey comes pre-set for Microsoft Word with "Document" as the text to check for in Word's title. When an autokey is due for Word, and "Document" appears in Word's window title, AutoKey will skip Word for the current interval. AutoKey also comes pre-set to look for "Untitled" in Windows Write's window title, and "unmodified" in WordPerfect for Windows.

Other Streamlining

The remaining check boxes allow you to tell AutoKey when to send keys depending on the protected application's state.

Don't send autokey if window is in background applies to applications that are open but inactive. If you've started an application, but are currently working in another, and you've checked this box for the first application's autokey, the first application will be skipped for the current interval.

Send autokey even if iconized applies to applications that you've opened, but that you've minimized. If you've done work in an application but have minimized it to free up space on the Windows desktop, AutoKey won't send the key unless this box is checked for the application.

The OK Button

Clicking *OK* on the Configure dialog box causes AutoKey to save all the information you entered. Any changes in the intervals between autokeys will show up in the application list.

The Cancel Button

Clicking the *Cancel* button on the Configure dialog box causes AutoKey to forget all your changes. The parameters for the application you were modifying will be unchanged.

Turn It Loose!

When you're done setting up all the applications you want AutoKey to affect, turn AutoKey on by clicking the *Turn On* button, and minimize it. (If the button reads *Turn Off*, then AutoKey's ready to go - just minimize it.) Then continue with your other work.

Using Help

Help is available by clicking the *Help* button. Most of the information in this manual appears in Help. The first thing you see in an introduction, and you can then click *Index* for the Help Index. You can jump to associated topics by clicking on the highlighted text in the Help window, and you can view the help from start to finish using the *Browse* buttons. Dotted-underlined words in Help have definitions associated with them. Clicking the words displays the definition.

Usage Notes

DOS Applications

Does AutoKey work with DOS programs running under Windows? No, but it does work around them. DOS applications are a special case in Windows. Windows creates a "virtual 8086" for DOS programs, letting them pretend they're running on a dedicated PC. In doing so, it turns off a lot of its normal processing, including the standard keyboard and mouse interface. So AutoKey can't talk to DOS applications.

This also means that Windows can't let AutoKey know about keystrokes and mouse clicks that you're making inside a DOS program, so keystroke and mouse click intervals that you've set up for target applications don't take into account work you're doing in DOS programs.

If you're running a DOS program in a window, AutoKey runs normally as far as other Windows applications are concerned, except for the problem with keystrokes and mouse clicks mentioned above. Timed intervals work normally.

If you're running a DOS program full-screen, Windows' timers slow down or even stop, depending on the multi-tasking conditions you've set up for the DOS program. This has the following effect on AutoKey. If the DOS program isn't running Exclusive (meaning it isn't hogging the entire CPU), AutoKey runs slowly, depending on system load. If the DOS program is running Exclusive (meaning Windows shuts all of itself down except for its task-switching keys), Windows, including AutoKey, goes to sleep until you switch out of the DOS program using Alt-Tab, Alt-Esc, Ctrl-Esc, Alt-Space, or Alt-Enter.

Screen Blankers

So what if you leave your PC on overnight, with AutoKey running? Will it still send its autokeys, doing a lot of harmless but useless work? Yes, unless you have a good screen-blanker running at the same time.

Some screen blankers notify Windows when they are shutting down the display, and AutoKey watches for this. When a screen blanker takes over the display, AutoKey deactivates its timers, and reactivates them when the system wakes up.

What if I'm in the middle of something and AutoKey tries to send a key?

You don't want AutoKey to step on you by typing keys for you while you're in the middle of opening a file, for example. So before it sends a key to an application, AutoKey checks to see if the application is actually ready for it, and if it isn't, waits for five seconds before checking again. When you're finished with your operation, AutoKey sends the autokey to the application.

But my program already has an auto-save!

Ah, but does it protect you from yourself? Most programs with an auto-save feature save their work in a temporary file(s), and the temporary file is deleted when you exit the program. This is almost always all you need, and it even allows you to tinker with a work while preserving a reasonably complete version on disk.

However, if you've been around computers long enough, you've probably told such a program, WordPerfect for example, to "ExitNoSaveYesI'mSure" when you were sure that that was EXACTLY what you didn't want to do!

So, AutoKey is worth running against all such programs, though this is a matter of personal taste.

Where does AutoKey store its data?

When you change the application list, it copies the list to the AUTOKEY.CFG file in the AutoKey directory (usually C:\AUTOKEY). This data is in AutoKey's internal format, so you can't modify it by hand. If something goes wrong, and you believe the application data is bad, your only recourse is to delete the AUTOKEY.CFG file and re-enter the data.

It is suggested that you make a backup copy of the AUTOKEY.CFG file once you're comfortable with your setup.

Future Enhancements

This initial development of AutoKey has emphasized its underlying function, at the expense of some bells and whistles. As it is, it's a very useful program, but no program is ever complete.

Assuming sufficient support from you, the user, AutoKey will continue to grow, becoming more functional and user-friendly. Here are some of the things likely to be included in a future release.

- * AutoKey should "eat" the defer keystroke when it is typed, but it currently doesn't. (This is considered a pseudo-bug, but is the only known bug in this version.)
- * Much cleaner user interface. No more typing; AutoKey will learn what you want it to do by recording your actions.
- * Glitz. Need we say more? AutoKey's not particularly exciting to look at, is it?
- * AUTOKEY.INI. The binary data in the AUTOKEY.CFG file is subject to corruption or deletion, and currently a hard-copy is simply not possible. Also, hacker types may desire to configure AutoKey using a text editor, instead of the user interface.
- * Finer, more complete, application configuration. You should be able to edit the application name, window title or EXE file name without disturbing other parameters.
- * Shortcut keystroke macros. "Schiz" magically becomes "Schizophrenia," etc. The advantage of having AutoKey do this is it would happen in every application you use, without setting up a macro for each application.
- * Enhanced interval control. Instead of a number of minutes, autokeys could be dispatched every day, every hour, once a year, after certain keys have been typed, etc.
- * Expanded application control. Instead of simple keystrokes or menu commands, AutoKey could become a full-fledged macro recorder, and it could also execute other applications or terminate them at user-specified times.

Glossary

Auto-save program. An auto-save program helps you preserve your work by automatically saving your data to disk at certain intervals. It's not the same as a backup program, which periodically backs up existing files on your disk to other media. Backup programs don't protect data in memory, just that on disk.

autokey (vs. AutoKey). "autokey" refers to the menu command or menu accelerator which is sent to target applications. "AutoKey" is the name of the program.

Default application configuration. This is the setup parameters given to a new target application that you've just added to AutoKey. This setup should work well for common Windows applications. This is the default target application configuration:

Menu Command:	File / Save
Menu Accelerator:	None
Time interval:	10 minutes
Keystroke interval:	2000 keystrokes
Mouse click interval:	500 clicks
5-second notification:	off
Defer key:	none
Window title screening:	none
Background autokeys:	off
Iconized autokeys:	on

Defer keystroke. This is the keystroke you use to skip an impending autokey. AutoKey can beep five seconds before it sends the autokey to a target application, allowing you to defer it by pressing the key(s) in this sequence. This keystroke can include Shift, Ctrl and/or Alt. You define this keystroke in the Configure dialog box.

EXE file name. This is the name of a target application's executable file. AutoKey uses this to determine if a target application is active, and to send the autokey to the application. See also Window title.

Menu command. This is a command in the target application's menu which AutoKey invokes at the proper interval(s).

Menu accelerator. This is a shortcut keystroke in the target application's menu which AutoKey invokes at the proper interval(s). Menu accelerators usually appear to the right of a menu command in the target application's menu.

Mnemonic. In Windows, this word refers to accelerator keystrokes which apply to window controls instead of menu items. They function identically.

Nested menus. Some Windows applications have complex menus. To organize their menus, they may "nest" menus by associating another menu with a menu selection. Selecting such a menu item will create a new menu instead of invoking a command, and the new menu may contain more menus as well as commands. AutoKey can invoke commands in nested menus.

Target application. This term refers to another Windows application which AutoKey

can control. In general a target application can be any Windows application other than AutoKey. AutoKey target applications appear in the application list.

Window title. This is the title of a target application's window. AutoKey uses this to determine if a target application is active, and to send the autokey to the application. Because many Windows applications change their window titles, you can use the * character as a wildcard at the end of the window title. See also EXE file name.