

Stiletto 98g

Table of Contents

Overview.....	5
Stiletto License and Lack of Warranty.....	6
Contacts for Questions or Support.....	6
Installation and Removal.....	7
Configuring a Button.....	8
Special Labels for Buttons.....	9
Battery Status Display.....	9
Resource Usage Displays.....	10
Using Icons to Label Buttons.....	11
Displaying Button Commands with Flyover Help.....	12
Tray Icon Buttons.....	12
Using the Keyboard to Access Buttons.....	12
Configuring Stiletto.....	14
The Info Dialog.....	15
Undoing Configuration Changes.....	15
The Bar Dialog.....	17
Working with Invisible Bars.....	19
Positioning to Stiletto bar.....	20
Changing Number of Buttons Displayed.....	20
Changing the Size of Stiletto Buttons.....	20
Buttons Configuration.....	22
Special Configuration Options.....	23
Clipboard Enhancement.....	24
Window Control Dialog.....	27
Automatically Moving the Mouse Cursor to a Dialog Button.....	28
No Click Command Execution.....	28
Specifying Explorer View and Arrangement Settings.....	28
Tracking Recently Opened Explorer Windows.....	29
Automatically Running Commands when Windows Open.....	30
Menu Setup.....	31
Running Commands at Stiletto Startup.....	32
Omitting Icons from some Menus.....	32
Accessing a Menu via Mouse Click.....	33
Menu Contents Tab.....	34
Submenus.....	35
Program-Specific Menu Contents.....	36
Displaying a Menu Centered or Offset from the Mouse Cursor.....	37
Customizing Layout of Menus with Icons or Colors.....	37
Displaying a Menu as a Toolbar.....	37
Pinning (Permanently Displaying) a Menu.....	38
Time Setup.....	40
Alarm Details.....	41
Using Alarms to Close or Wait on Active Tasks.....	42
Suspending Alarms.....	43
Adding Reminder Message Alarms.....	43
Alarm Log.....	43
Stiletto Sounds.....	45

Paper/Saver Tab.....	46
Hot Keys and Mouse Action Commands.....	47
Program-Specific Hot Keys.....	48
Command Entry Controls.....	49
Entering Command Entry Controls Information.....	49
Running Multiple Commands.....	51
Wait Command.....	51
Stiletto Built-In Commands.....	54
Built-in Commands: Menu Structure.....	54
Built-in Commands: Active Task Manipulation.....	54
Built-in Commands: Messages, Alarms, and Timers.....	55
Built-in Commands: Stiletto Position and Size.....	56
Built-in Commands: Exit Windows or Stiletto.....	56
Built-In Commands: Execute Files, Documents, or Prog Mgr Commands.....	57
Built-in Commands: Screen Saver.....	58
Built-in Commands: WallPaper, Screen Saver, and Sound.....	59
Alphabetic List of Built-in Commands.....	60
Manipulating Windows of Running Programs.....	62
Specifying Action for Control Window Command.....	62
Specifying the WindowID for the Control Window Command.....	63
Switching to another Active Window.....	63
Closing an Active Window.....	64
Making a Window Not Always On Top.....	65
Making a Window "Always On Top".....	65
Minimizing a Window.....	65
Active Window Switching with Buttons.....	66
Omitting Windows and Words from Active Window Lists.....	68
Tiling Active Windows.....	68
Sending Key Strokes to Other Programs.....	69
Specifying the Window to Receive the Keys.....	69
Specifying the Keys to be Sent using Send Keys.....	70
Specifying the Keys to be Sent using Send Keys (Blank Separator Approach).....	71
Examples of Send Key Commands.....	72
Sending Keys to Programs When They Are Started.....	73
Send Mouse Clicks to the Active Window.....	74
Sending Mouse Clicks and Moves to the Active Window.....	74
Sending URLs to a Running Web Browser.....	75
Running a Command with a Randomly Selected File.....	75
KeyBoard Macros.....	77
Timers.....	79
Controlling Timers Externally.....	80
Controlling Timers by the Status of the Modem.....	80
Setting Timers and Associated Commands.....	81
Setting Timer Value and State.....	82
Timer Logs.....	83
Changing the Timer a Button is Displaying.....	83
Folder Contents Menu.....	85
Entering Work Directory Information for Folder Contents Command.....	86
Special Folders for Folder Contents Menu.....	87
Virtual Desktops.....	89

Copying Buttons on the Bar.....	93
Scripts: Launching or Closing All Commands on a Menu.....	93
Dynamically Changing Button Color.....	95
Dynamically Changing Button Text.....	96
Dynamically Changing Button Icon.....	96
Filling a Menu with All Active Windows.....	96
Scrolling with the Mouse.....	98
Moving Stiletto.....	99
Drag and Drop onto the Stiletto Button Bar.....	100
Drag and Drop Rename.....	101
Accessing Program Manager or Other Shell Groups.....	102
Cut and Paste Commands.....	102
Running Multiple Stiletto Button Bars.....	103
Stiletto Command Line.....	103
Restarting Stiletto with a new Configuration.....	105
Manually Reconfiguration of Stiletto.....	105
Automatically Reconfiguration based on Active Window.....	105
Showing and Moving the Bar Temporarily.....	107
Position Stiletto.....	108
Caps Lock and Scroll Lock.....	108
Hiding Windows.....	108
Working with Tray Icons from Other Programs.....	108
Training Stiletto to Recognize Tray Icons from Other Programs.....	109
Positioning Windows with Stiletto Exec.....	110
Automatic Minimization of Inactive Windows.....	111
Automatically Hiding Windows.....	112
Executing Dos Commands.....	113
Browse and Run Command.....	114
Tiny Type and Run Dialog.....	115
Copy Characters, Files, Date, Time to Clipboard.....	116
Inserting Text in a Log File.....	117
Using the Clipboard Contents as the Command Parameter.....	118
Changing Screen Display Resolution.....	119
System Resources Window.....	120
Show Size and Position of Windows as These Are Changed.....	120
Saving and Restoring Desktop Icon Positions.....	121
Features of NT and Win95 Versions of Stiletto.....	122
Advanced Configuration Dialog.....	122
Internal Configuration Options.....	124
Frequently Asked Questions.....	126

Overview

Stiletto is a tiny button bar task switcher/launcher and clock/calendar which lets you use any mouse button to run a command. **Stiletto** also includes menus, hot keys, alarms, task scheduling, sounds, wallpaper, and screen saver control: it handles your command launching and desktop needs using a single consistent but unobtrusive interface.

Stiletto incorporates these features:

- A small-footprint button bar launcher (e.g. fits over title bar of maximized window).
- Use of any mouse button to launch commands.
- Drag and drop files to start commands.
- User-configurable command menus with submenus.
- Create menus on the fly from file directory contents.
- Attach menu to button bar, desktop, title bar, or all three.
- Start commands from hot keys, mouse clicks/chords, tapping ctrl/alt/shift, or screen corners.
- Multiple button bars (through multiple instances and configurations).
- Floating button bar, choice of many resolution-independent standard positions, or position in caption of active window.
- Direct access to Start Menu, program manager or other desktop shell groups and their commands.
- Switch to or close any active window using a menu or button bar.
- Make any active task or launch any command as always on top.
- Text label, icon, clock, date, stopwatch timer, or resource display on any button.
- Adjustable button size.
- Built-in commands for drag and drop file rename, windows exit/restart, browsing and running files (with run history), moving the button bar, playing sounds, and others.
- Control of Caps Lock/Shift and Scroll Lock behaviour.
- Support for file associations.
- Reminder messages, regular chimes, and scheduled start/stop of commands.
- Wallpaper display and switcher/randomizer.
- Screen saver display and switcher/randomizer.
- Randomization and testing of system and application sounds.
- Random noise making by playing sounds sporadically as specified.
- Send key strokes to programs when launched, or to running programs.
- Flyover (balloon) help to display the commands for any button.

The Win95/NT 4 shell provides many useful features; **Stiletto** is intended to supplement it by providing quick, minimal-mouse click access to your most used commands while taking up little desktop space, and to provide utilities related to **Windows** start-up and time, all in one consistent package.

Stiletto License and Lack of Warranty

The **Stiletto** program, DLL, Help File, Word Document File, and readme file are all Copyright 1995-1998 by Bruce Switzer. All Rights Reserved.

The Stiletto icon was created by Jonas Hjortlund.

THIS SOFTWARE IS DISTRIBUTED "AS IS," WITHOUT WARRANTY AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH THIS PROGRAM MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM.

Stiletto may not be sold nor be used in any profit-oriented endeavor without the express written permission of the author with the exception that **Stiletto** may be distributed freely via media intended to make shareware available to the public for trial. All files, including the **Stiletto** program, DLL, help file, readme file, license file, and all others in the **Stiletto** zip file, must be included.

All trademarks used in this Help File are the property of their respective owners and are used for explanatory purposes only.

The jpeg conversion routines in this software are based in part on the work of the Independent JPEG Group.

Contacts for Questions or Support

Find out about the latest **Stiletto** version at
<http://www.inforamp.net/~crs2086/index.htm>

You can contact me via e-mail
bswitzer@kagi.com

Or you can write me at

Bruce Switzer
204 Duplex Avenue
Toronto, Ontario, Canada
M5P 2B2

Installation and Removal

Installation

For automatic installation, unzip stil98c.zip to a temporary folder and double click on setup.exe.

For manual installation:

If you have never used Stiletto 98:

To install Stiletto, first unzip the Stil98c.zip file into a temporary directory, then unzip stilinsc.zip into a fresh directory which will serve as your main Stiletto directory. You can delete the stilinsc.zip file and temporary directory when you are done.

If you are already using Stiletto 98:

To install, first make a backup copy of your Stiletto directory, then shut down any running Stiletto bars, and finally unzip stilinsc.zip into your current Stiletto directory, overwriting all files there. You can delete the stilinsc.zip file directory when you are done.

If you used Stiletto 16 bit:

To install Stiletto, first UnZip the Stil98c.zip file into a temporary directory, then unzip stilinsc.zip into a fresh directory which will serve as your main Stiletto directory, You can delete the stilinsc.zip file and temporary directory when you are done.

You can copy the stiletto.ini file from your 16 bit Stiletto directory to the main Stiletto directory to re-use your configuration.

Double click on stiletto.exe to start **Stiletto**. Double click on Stiletto.hlp for information.

If you find **Stiletto** useful, you will probably want to put in it your StartUp group so that **Stiletto** starts automatically with Windows. For Win95/NT 4, use the Info tab on the configuration dialog to do this; you activate the configuration dialog by ctrl+left click on the button bar.

If you used the 16 bit version of **Stiletto**, you can copy your Stiletto.ini files into the **Stiletto** 98 directory and access them from the 32 bit version.

Removal

To de-install **Stiletto**, remove it from your StartUp group and erase the directory containing **Stiletto**.

Configuring a Button

You use the Configure Button dialog to set the label of a button and the commands which are run when you click the button with the left, middle, or right mouse button.

(You can simulate the middle mouse button by holding down shift while pressing the left button or by pressing both buttons at once after selecting this option from the Bar dialog).

To set the button label, type a label of up to 39 characters into the label combo box or select a special label from the drop down box. If desired, choose an icon using the Icon Source drop down. You set text and icon position using the Buttons dialog

Set the commands for each mouse button using the command entry controls.

You can optionally enter help text to be displayed with flyover help.

You can change the size of an individual button by setting the Width Magnify percentage (for horizontal button bars) or Height Magnify percentage (for vertical button bars).

You can set this button's face and text colors by checking the Own Color checkbox and selecting the colors after pressing the text and face buttons. If you do not set both face and text, defaults are used (gray for face, black for text). Unchecking the box causes the button to revert to the bar colors. Unless you check the freeze color checkbox, you can also dynamically change the button color with the built-in Change Button Color command.

You can set the button text font by checking Own Font and setting the font with the Font button.

For Win NT4/95, you can specify that **Stiletto** buttons appear as tray icons.

The Bar dialog lets you set the bar and base button size, bar orientation, button and flyover color, default font for labels.

The button configuration dialog can be accessed by pressing and holding the button to be configured, by selecting the button to be configured from the Buttons dialog, or by dropping a file on the button with the Alt-key held down, or under Windows 95/NT 4, by right-dropping a file to a button with the bar tab item "Show menu for right drag-drop" checked..

If you have swapped the mouse buttons using Control Panel - Mouse, then **Stiletto** will try to automatically swap the right and left labels on the button dialog. Automatic label swapping does not work with all mouse drivers, so you can also do it manually with the Bar dialog.

Special Labels for Buttons

See button [configuration](#).

Using the drop down combo box in the top left of the button configuration dialog, you can:

- select a time or date display from the label drop down lists. (Use [Time Setup](#) tab if you want seconds on time display)
- select one of the resource [usage](#) displays from the drop down box.
- select [battery](#) status display from the drop down box.
- select a [timer](#) as a label and assign its [timer id](#).

If you select and time or date display from the drop down, you can also specify an offset in minutes to be added to the time before display. You can specify a negative or positive offset, but avoid the + sign for a positive offset.

Battery Status Display

For portable computers, you can display the status of the battery on a button or in the resources [windows](#).

The display consists of these three fields:

- percent of battery power remaining (255% means no information available)
- character + if battery charging, - if discharging, ? if unknown charging status
- AC** if ac connected, **DC** if battery power being used; ?? if unknown.

Resource Usage Displays

Stiletto can display free Windows resources as a button label by selecting the desired display from the Configure Button dialog. (Free GDI and User resource display are not available in Win NT).

(You can also use the Show System Resources built in command to display system resources).

These displays are available:

Free GDI Res	Displays the percentage of free GDI resources in the form NNg, where NN is the free percentage.
Free USER Res	Displays the percentage of free USER resources in the form Nnu.
Free Physical Memory	Displays free physical memory in Kilobytes.
Free Virt/Phys Memory	Displays free page file plus physical memory and free physical memory in Megabytes.
Free Virtual Memory	Displays free page file plus physical memory in Kilobytes.
Min of GDI/USER	Displays the minimum of GDI and USER in the form NNf.
GDI/USER	Displays both GDI and USER free resource in the form NNg/MM, where NN is the free GDI resources (shown by the g) and MM the free USER resources
Min and Mem (MB)	Displays the minimum of GDI and USER as well as the free physical memory in Megabytes in the form NN/MM.M, where NN is the minimum of free resources and MM.M is the free memory in megabytes (1024K).
Free Disk Space (MB)	Displays free disk space in MegaBytes.

GDI resources include device-context handles, brushes, pens, regions, fonts, and bitmaps. USER resources include window and menu handles and related structures.

Use the Time Setup dialog to set the rate at which **Stiletto** refreshes the resource displays. You can also use this dialog to set a level for **Stiletto** to automatically monitor USER and GDI resources.

The drive letter is normally shown on the free disk space display; put "ShowDriveLetter=0" under [General] in the stiletto.ini file to avoid display of the drive letter.

Using Icons to Label Buttons

Stiletto can display an icon on a button label; to access this feature select the desired display entry from the icon drop down box at the left of the Configure Button dialog.

These displays are available:

Icon from Left	Displays icon from left command.
Icon from Middle	Displays icon from middle command.
Icon from Right	Displays icon from right command.
Icon from File	Displays icon from file that you browse for or key in to edit box.

Controls for working with icons will appear when you select one of these entries.

If there is more than one icon in the file, **Stiletto** lets you select which icon you wish to display.

You can display both icons and text on label and set the percentage of the button space devoted to the icon using the icon size % edit box.

Stiletto normally selects the large icon and stretches/shrinks it to fix the size of the button face. If you prefer, you can select a fixed size icon, either large or small (Win95/NT4 only). This will usually yield a better looking icon.

If you change the command file or icon file, you may need to press the **Show Icon** button to see the new icons from the selected file.

Displaying Button Commands with Flyover Help

If you hold the mouse button over the **Stiletto** button bar without pressing a mouse button, **Stiletto** will display a small window showing the commands associated with each of the three mouse buttons for the underlying button. Each of the letters **L**, **M**, or **R** (Left, Middle, Right) appears beside the appropriate command.

After you press a button or move the mouse off the **Stiletto** bar, this window will disappear.

You can control whether **Stiletto** displays the help window with the Bar dialog from the Configuring Stiletto built-in command. You can also use this dialog to control whether resource usage is displayed on flyover help: **Stiletto** can optionally display free space for local hard disks; free physical kilobytes, used memory percentage, and GDI/User free percentage (Win 95 only); and timer value for the button and time/date.

Use the Configure Button dialog to set the text displayed with flyover help. If you do not specify any text, the commands themselves are displayed.

If you have swapped the mouse buttons using Control Panel - Mouse, then **Stiletto** will try to automatically swap the right and left labels on the button dialog. Automatic label swapping does not work with all mouse drivers, so you can also do it manually with the Bar dialog.

After you launch a command, flyover help is normally suspended until you move the mouse off the **Stiletto** button bar. This is to prevent problems with full screen **Dos** commands: if flyover help is not suspended, flyover help will appear unnecessarily after you activate a full screen **Dos** command and minimize the **Dos** window. But, if you never use full screen **Dos** and want flyover help not to be suspended, you can use an internal Stiletto ini option to accomplish this.

Tray Icon Buttons

Using the button dialog, you can specify that **Stiletto** display a button as a tray icon on the Win95/NT4 task bar. Left/middle/right mouse clicking on the tray icon runs the same commands as clicking on the corresponding **Stiletto** button.

The icon displayed in the tray is the one chosen for the button. If no icon is available, the **Stiletto** icon is used.

You can show as many buttons as you want as tray icons. The buttons need not be displayed on the button bar; for example, you could configure a button bar to show buttons 1-16 and to show 17-20 as tray icons.

In addition, there is an extra set of 10 buttons which can only be displayed as a tray icons. They are accessed from the Buttons dialog.

Using the Keyboard to Access Buttons

You can use the keyboard to access the commands on the **Stiletto** button bar.

First, you need a way to activate the bar from the keyboard. Set a hot key to the following command:

Command *Stiletto Exec

Parameter mouse stiletto
(put a number n after stiletto to start at the nth button)

When you activate the hot key, the mouse cursor will be moved to the bar and the bar will be ready to receive any of the following keystrokes:

L	activate left command of current button (you can also use Enter)
M	activate middle command of current button
R	activate right command of current button
left arrow	move to next button
right arrow	move to last button
end	move to last button
home	move to first button
up arrow	move to next row in multi-row bar
down arrow	move to previous row in multi-row bar
Esc	move mouse cursor to position it had before *stiletto exec mouse stiletto

Configuring Stiletto

You configure **Stiletto** buttons, menu contents, sounds, paper/saver, hot keys, and alarms with the Configure **Stiletto** set of tabbed dialogs.

You start this dialog by left-clicking anywhere on the **Stiletto** bar with the Ctrl key pressed, or through the Configure **Stiletto** built-in command. The command displays a set of tabbed dialogs as follows:

Info shows Windows information and resource usage; allows you to register **Stiletto** and to put **Stiletto** on your Start Menu.

Bar allows bar position, font, layout to be set.

Buttons sets button number, order, commands, active task tracking

Special sets functions which customize your Windows interface

Windows Control sets functions which customize your Windows interface

Menu Setup controls the format of the active window list, the mouse click used to start a menu by clicking the desktop, and the format of menus.

Menu Contents allows you to change the contents of menus.

Time Setup controls low resource warnings and the sounds associated with alarms and chimes.

Alarm Contents allows you to add or change alarms.

Sound controls the sounds played for windows and application events and allows you to specify how **Stiletto** should automatically change them.

Paper/Saver allows you to change the Windows wallpaper or screen saver and to set up automatic changes of either.

Key/Mouse allows you to assign commands to hot keys or mouse actions

Key Setup allows you to fine tune hot keys or mouse actions

You can switch to a new tab by left-clicking it or by pressing Ctrl+tab or Ctrl+shift+tab on the keyboard.

All the tabbed dialogs share one OK and Cancel button: If you press OK, all changes made on all tabbed dialogs are saved and **Stiletto** restarts with the new configuration. If you press Cancel, all changes made on any tabbed dialog are discarded. Note however that the Configure Button dialog has its own OK and Cancel; if you press OK on this dialog the changes are saved even if you press Cancel on a tabbed dialog. Several dialogs also have an apply button which allows you to preview changes to the bar appearance.

Stiletto normally positions the tabbed configuration dialog near the bar; if you prefer it to be centered on your screen, you can use the CenterConfig internal option. You can disable the configuration command with the Lock internal option.

The Info Dialog

The info dialog appears whenever you select the Configure Stiletto built-in command. It corresponds to the Info tab. Use the dialog to: insert or remove **Stiletto** short cuts, register **Stiletto**, undo configuration changes, save and restore desktop icons positions, create new button bars.

Configuration

For information on any control on the dialog, click the ? in the upper right and then click again over the control you are interested in.

Stiletto shortcut buttons: For Win95/NT4, you can create shortcuts for **Stiletto** on any of your Start Menu, your Programs Menu, your Accessories Group, or your Start Up Group (if you put a **Stiletto** in your Start Up Group, **Stiletto** will be automatically started when Windows starts). You can also remove all shortcuts. The buttons are disabled when the shortcut is already in place.

Undo Configuration button: You can undo the last set of configuration changes.

Registration Info Button: The title on the dialog indicates whether you have registered Stiletto. You can pay for **Stiletto** or enter the registration code through a dialog accessed with the button.

Icon save/restore buttons: There are buttons to save and restore desktop icon positions.

Create New Bar Button: There is a button to create a new bar. To run a bar after prompting for its configuration file, use

Command c:\your\path\stiletto.exe
Parameter ?

To run the bar corresponding to myconfig.ini, replace the ? by myconfig.ini.

Advanced: Accesses advanced configuration options.

Undo Configuration button: You can undo the last set of configuration changes.

Registration Info Button: The title on the dialog indicates whether you have registered Stiletto. You can pay for **Stiletto** or enter the registration code through a dialog accessed with the button.

Icon save/restore buttons: There are buttons to save and restore desktop icon positions.

Capture tray icon info: Used to help access tray icons from other programs

Info and Resource Lists: The left hand list box shows how long Windows has been running, what type of CPU you have, whether or not there is a math co-processor present, what mode Windows is running in, and the name of the configuration file used by this instance of **Stiletto**. The right hand list box shows the percent of GDI and user resources free (if the InfoResources internal option is used), the percent of free memory, and the number of free kilobytes on each hard disk drive. (For Win NT, free resources are not available and are shown as 99).

Use the time setup dialog to set a low resource warning alarm.

Undoing Configuration Changes

Stiletto lets you undo the last set of configuration changes you made using the "Undo" button on the Info dialog.

This button will be enabled when an undo configuration file is created and disabled when the undo configuration file is used. The undo file is created when the OK button is pressed from any configuration tab. The undo file will retract all changes made since the OK button was previously pressed on the config dialog.

Stiletto also keeps a backup file of your configuration file; this backup is created and changed each time you press OK. The backup file name is config.bki where "config" is the name of the backed up config file (eg stiletto).

The Bar Dialog

Purpose

The Bar dialog sets the bar position, color, visibility, base size. It also configures flyover help, lets you specify that right and left mouse buttons select the middle command, and lets you specify what happens when you right-drag/drop a file from explorer onto to bar.

You can configure a bar to be invisible.

Configuration

To access the Bar dialog, ctrl+left click the **Stiletto** bar and select the "Bar" tab from the Configuring Stiletto dialog.

For information on any control on the dialog, click the ? in the upper right and then click again over the control you are interested in. Press the Apply button at any time to preview the effect of any command which affects the bar look or size.

Checkboxes:

Bar always on top	Bar is set always on <u>top</u>
Use Windows button colors	Uncheck to set colors with buttons at right of dialog
Flat look for bar	Check to remove separators between buttons; gray check for heightened 3D effect
Number flyover help lines	Set number of lines of <u>flyover</u> help; use 0 to turn off flyover help: 2 lines shows Left, Right; 1 line shows left only.
Show resources on flyover help	Check to see resources, disk free, time, on <u>flyover</u> help.
Show menu for right drag/drop	Uncheck to have <u>right-drag/dropped</u> file from Explorer used on the right command; check to see menu to allow either configuration or running of a command
Swap left/right mouse labels	Check to have Stiletto swap labels related to right/left mouse buttons.
Click left/right mouse for middle	If checked, clicking right and left activates middle command
Hide bar after each command	Use to make Stiletto invisible after each command.
Hide bar at Stiletto start	Check to hide bar at Stiletto start (hide bar after each command also must be checked)
Hide bar if screensaver	Check to make Stiletto take extra steps to hide itself if a screen saver is running.
Hide bar if full screen	Stiletto hides itself if a full screen program is running.
Hide caption bar	Check to have Stiletto hide itself if you select a <u>caption</u> position and there is no active window.
Auto reconfigure	Advanced feature to automatically change bar configuration based on active <u>window</u> .

Drop Down boxes:

The drop down boxes "show bar when mouse bumps screen edge", "milliseconds mouse held at edge to show bar", and "delay for hide/move" are used to fine tune the configuration of invisible bars.

The flyover help delay drop-down controls how long the mouse cursor must be positioned over the bar before flyover help is displayed.

Click Desired Position

Use the controls to set the bar position. Click an unlabeled button to position the bar at that part of the screen. Or select a task bar or caption position.

You can also position **Stiletto** by clicking and dragging near the left hand side of the bar (top for vertical bars). (You can use the advanced edges option to disable this feature, if you want).

Button Sizes

Use the edit boxes under "Button Sizes" to set the base horizontal and vertical size of the button bar relative to the base size. (At 100%, **Stiletto** will be as high as a caption and button width will be based on caption icon width). Use the "Big Icon Size" and "Small Icon Size" to set buttons size just large enough to hold the corresponding icon.

Bar Color and Flyover Color

Use these buttons at the right of the dialog to set bar color and default font. Make sure you uncheck "Use Windows button colors" first.

Working with Invisible Bars

The bar configure dialog contains options for hiding **Stiletto**.

Check the "Hide Stiletto after command" to make **Stiletto** invisible after each command launched from **Stiletto**. Gray check to show a narrow strip where the mouse can be bumped at screen edge to show as set by the "Show Stiletto when mouse bumps screen edge" drop down box. You can change the strip size with the internal "MarkerSize" option.

Before using this option, make sure you have a way to make **Stiletto** visible again! You could use the mouse cursor option (see next paragraph) or attach the built-in command Show/Move **Stiletto** to a menu or to another Stiletto button bar which is running. If you forget to do this, edit the ini file and remove the autohide option from the [General] section.

Use the "Show Stiletto when mouse bumps screen edge" drop down to determine whether **Stiletto** should be made visible when the mouse cursor passes over the nearest screen edge. **Stiletto** will be shown if invisible or uncovered if hidden by another window. For vertical **Stiletto**, the nearer vertical screen edge is used; for a horizontal button bar, the nearer horizontal screen edge is used. For this option, if "Hide Stiletto after command" is checked, **Stiletto** will become invisible again when you move the mouse cursor off the button bar, even if no command is executed.

You can control how long the mouse has to be held at the edge with the "held at edge" drop down; you can also control how long **Stiletto** stays visible with the "autohide delay" drop down.

You can indicate that bars positioned in the caption are to be hidden if no visible window is active

Assign the command

Command *Show/Move Bar

Parameter *toggle

to a hot key to use the hot key to reverse bar visibility.

You can hide a **Stiletto** bar from another program by executing the command

```
c:\yourpath\stiletto.exe config.ini hide
```

where config.ini is the name of the configuration file of the executing **Stiletto** bar. You can show a hidden bar with

```
c:\yourpath\stiletto.exe config.ini show
```

You can animate the disappearance of the bar with the advanced configuration dialog. You can also specify that the bar should only be visible if a specified window is the foreground window.

Positioning to Stiletto bar

The bar configure dialog contains options for positioning **Stiletto**.

Use the buttons in the middle of the dialog to set the position for **Stiletto** on your desktop. The buttons with no label place **Stiletto** at a fixed position on the screen. For help on the individual positions, click on the ? in the upper left of the dialog then click on a position of interest. (You can also position **Stiletto** manually using the **Stiletto** Position built-in command.)

There is also a check box to indicate whether **Stiletto** is to float always on top of all other windows.

Note for task bar positions: the first time you set a task bar position on an autohide task bar, first disable auto hide, then set Stiletto to a task bar position, then re-enable autohide. You may also need to disable autohide and restart Stiletto.

Changing Number of Buttons Displayed

You can change the number of displayed buttons on a **Stiletto** button bar by executing the built-in command Bar Size n, where n is the new number of buttons to be displayed and is placed in the command parameter edit box. For example, Bar Size with a command parameter of 12 sets the number of displayed buttons to 12. The command may be assigned to a button or a menu.

This command has exactly the same effect as using the Buttons dialog.

If you omit the number "n", **Stiletto** will prompt for it using the command parameters dialog.

You can also make **Stiletto** invisible by using Bar Size 0. In order to make **Stiletto** visible again, use the "Show Stiletto when mouse bumps screen edge" feature on the Bar dialog. Or, you can assign the built-in command Show/Move **Stiletto** to a menu, attach this menu to the desktop or to the window captions using the Menu Setup Dialog, and execute this command.

You can increase (or decrease) the number of buttons by specifying +n (or -n). For example, Bar Size +1 adds one more button. You could use the feature to add a new button to which you can quickly assign a new command by (eg) dropping a file from File Manager with the Alt key held down (see button and Menu Contents dialogs).

If you have a set of different sizes you want to quickly switch among, you may want to set up a submenu of size commands on a menu.

You can assign the Bar Size 0 command to the Start Up menu specified on the Menu Setup Dialog in order to make **Stiletto** initially invisible.

This command will be ignored while you are configuring **Stiletto** or if you put a non-numeric character after Bar Size.

Changing the Size of Stiletto Buttons

You can change the size of **Stiletto** buttons using the width and height magnify percentages on the Bar dialog and the Configure Button dialog. **Stiletto** sets the base button size using the Bar dialog and then modifies this size for each particular button with the percentages from the Configure Button dialog.

These numbers are percentages: setting them to 100 yields the default button size. You can experiment with any other value between 10 and 999.

If you choose too small or too large a value, you may find that you can no longer access the button with your Configure Stiletto command. As long as you can still access any button, press and hold it and then assign the configuration command to that button so you can reset the percentages. Or, another way to solve this problem is to edit the stiletto.ini configuration file; look in the [General] section and set Ymagnify=100 and Xmagnify=100.

Buttons Configuration

The Buttons dialog is selected by clicking on the "Buttons" tab from the Configuring Stiletto command.

Use the set of buttons in the dialog to configure the Stiletto buttons. Left click on a button to configure it. Right drag and drop to move or copy buttons. Any button can be displayed as a tray icon; in addition there are 10 buttons at the bottom of the dialog reserved for use as tray icons.

Use the edit box and spin controls to set the number of buttons displayed, or check "Set to Number Used" to have **Stiletto** set number of visible buttons to highest button which has a command assigned to any mouse click. You can also change the number of displayed buttons by clicking and dragging the right-hand side of the bar (bottom for vertical bars).

Use the edit box and spin box to set the number of rows of buttons (columns for vertical bar). (Not available for active bar buttons).

Use the drop down boxes at the right of the dialog to control the position of icons and text on button labels. (The above position applies to vertical button bars only).

There are controls for dealing with Active Task Buttons, and controls for specifying that commands on a button should be executed when the mouse is over held over the button without clicking.

Using the check boxes to specify which type of commands to execute: only *Show Menu commands, active bar buttons, or everything other than menus or active bar commands. Or you can check All to have all commands executed.

You can use the spin box to specify the time in milliseconds during which the mouse cursor must stay over the button for the command to be executed.

Special Configuration Options

Purpose

This dialog configures many options related to the way you interact with Windows.

Configuration

The Special GUI dialog is selected by clicking on the "Special GUI" tab from the Configuring Stiletto command.

For information on any control on the dialog, click the ? in the upper right and then click again over the control you are interested in.

There are check boxes for controlling Caps Lock and Scroll Lock keys.

There is a check box to specify that documents launched by **Stiletto** should be added to the Win95/NT4 recent documents list.

You can indicate that **Stiletto** should show more of windows which it activates and which are mainly off the screen.

You can indicate that windows should be centered when switched-to from the active window list or the active task buttons.

There is a checkbox to indicate that **Stiletto** should automatically track the history of plain text items pasted to the clipboard.

You can automatically restore saved desktop icons positions when the screen resolution changes; however, checking this box can lead to Explorer aborts with some programs which change resolution and then display a lot of information (eg those that play movies).

You can specify that alt-clicking on a button closes the corresponding program, if that program is running.

You can specify that **Stiletto** should act as a Win 4 App Bar and reserve a strip of screen space like the Windows task bar.

You can specify that right-clicking the title bar closes the corresponding window: however, this setting is intended for older configuration compatibility: use hot keys captions and window under mouse for new configurations.

You can specify that **Stiletto** should disable the screen saver while a RAS connection is active.

You can specify that **Stiletto** should enlarge the file list windows use in file open and save dialogs (only works for programs that use standard Windows dialogs).

You can specify whether all windows from all virtual desktops are shown on the task bar and whether or not **Stiletto** should activate an entire virtual desktop when one of its windows is activated. You can specify whether all commands on a saved virtual desktop should be relaunched each time you activate the desktop.

You can use the middle mouse button and mouse movement to scroll windows.

You can indicate that **Stiletto** should pan (move) windows into view when the mouse is held over them at the screen edge; you can set the speed of panning by setting the step size in pixels.

You can indicate that **Stiletto** should activate windows when the mouse passes over them and set a delay in milliseconds for how long the mouse has to be over the window for it to be activated. You can further specify that the active window should only be changed if the mouse is over a caption.

You can select a mouse plus modifier key to size any window. To use, click on the window with the mouse/key combination and drag to resize. If you move the mouse too quickly, resizing may stop: in this case release the mouse key and continue moving to size. Click again to stop the sizing.

You can ask **Stiletto** to automatically press buttons when the mouse is stopped over them for a specified time.

Use the check box on the Special GUI tab to do this. You can also set the stop time with the spin box. If you want **Stiletto** to automatically select standard menu items too, grey check the box.

By default, **Stiletto** will automatically press buttons (including radio buttons and check boxes), combo boxes, combo box list items, standard toolbars, and tabs in standard tabbed dialogs. **Stiletto** also will automatically press the minimize, maximize, close, help, and system menu buttons in captions and will automatically open standard menus in menu bars.

You can add or remove window types to this list as follows: Assign the command
*Stiletto Exec
autopress
to any hot key. Avoid using Alt as a modifier key as this will close open menus.

Move the mouse over the window of interest and activate the hot key. If the window type is not current one that is automatically pressed, **Stiletto** will add it to its list. If it is one that is automatically pressed, **Stiletto** will remove it. In both cases **Stiletto** notifies you of the results with a message box.

If you use "Cursor to default button" from Window Control tab, **Stiletto** will not press the button unless you move the mouse from where **Stiletto** positions it.

Clipboard Enhancement

Purpose

Stiletto can track text as you paste it to the clipboard and can subsequently display a list of copied text items on a menu; the selected item is recopied to the clipboard and optionally pasted. You can also manually copy the clipboard or selected text to a file. Furthermore, you can arrange permanent text snippets in submenus to provide a library of text phrases which can be copied to the clipboard and automatically pasted.

Configuration

To enable automatic tracking of plain text as you paste it to the clipboard, you must check "Keep clips" on the Special GUI dialog. When this is done, **Stiletto** will automatically track the most recent items pasted to the clipboard. Grey check to capture rich text format as well as plain text.

You have **Stiletto** place captured items in subfolders of the clip folder by entering filter strings in the filter edit box on the dialog. Filter strings take this form

String=subfolder

String is xxx*, *xxx, or *xxx* to match xxx at start, end, or middle of clipped item; subfolder is the name of the subfolder of the clip folder in your **Stiletto** directory where you want to put any item matching the String. For example

*.zip="zip files"

puts any captured item ending in .zip into the subfolder "zip files". Note that you must put the subfolder name in double quotes if it contains blanks.

You can separate multiple matching strings by commas:

.gif,.jpg,*.jpeg=Pictures

puts strings ending in .jpg, .jpeg, or .gif into the pictures folder. Avoid blanks in the matching String.

The strings in the clip filter edit box on the Special GUI dialog are processed in sequence:

try*=tryfiles *.zip="zip files"

would put any strings starting with try in tryfiles and then any other strings ending in .zip in "zip files".

If the captured item is longer than 250 characters, only the first and last 125 characters are used when checking filter strings.

You can show a menu of captured items with the command

Command *Clip

Parameter menu

The first characters of copied items are shown. Selecting an item copies it to the clipboard. If you use **menupaste** instead of **menu**, the selected item is automatically copied and pasted to the foreground application by send the Ctrl-v character (use **menuipaste** to send Shift-ins).

To paste any file to the clipboard, use:

Command *Clip

Parameter file filepath

Use **filepaste** or **fileipaste** to automatically paste the file as well.

Manual Copy to Clipboard

You can manually copy selected text to the clipboard and then to a specified file with

Command *Clip

Parameter copyto filepath

Stiletto will send Ctrl-c to copy selected text to the clipboard, and then will copy the clipboard to the filepath. Provide the full filepath with the extension **.clprtf**: **Stiletto** will always use the clprtf format for the saved information, even if rich text is not available. Use **to** instead of **clipto** to copy information without sending Ctrl-c.

Further Information

Clipboard plain text is actually stored in a .txt file in the clip subfolder of your main **Stiletto** folder. You can edit it with your standard editor. You can access that editor from the *Clip menu by right-clicking the menu item. Clipped rich format text is stored in .clprtf files which only **Stiletto** can read.

Using Explorer, you can create subfolders of the clip folder and use these subfolders to permanently store text snippets you want to access. Create the snippets by copying them from the main clip folder, by using clip filters, or by entering them directly by saving files from Notepad or any other editor which can save plain .txt files. You can then access these snippets from the *Clip menu.

You can clear the clipboard with

Command *Clip

Parameter clear

You can access only the clips in one folder xxx with

Command *Clip

Parameter menu xxx

You can show only the automatically tracked text items with:

Command *Clip

Parameter menu active

Clip Menu Format

The clip menu format can be changed by entering keywords in the work directory edit box. Enter **sorttime** to sort the clip lists by time instead of alphabetically. Enter **mne** to show menu mnemonics for the first 36 items to allow easy selection from the keyboard.

The *Clip menu command is actually implemented by a *Folder Contents Menu similar to the following:

Command *Folder Contents Menu

Parameter c:\program files\stiletto\clip

Work noext noicons folderstart folderdot cmd “*Clip file”

If you would like a different display of the clip menu, create your own *Folder Contents Menu command using the above as a model. Note the cmd field which runs a *Clip file on the selected *Folder Contents Menu item. (Use *Clip filepaste to autopaste).

Window Control Dialog

The Window Control dialog is selected by clicking on the "Window Ctrl" tab from the Configuring Stiletto command. It is used for control window behavior.

You can specify that **Stiletto** should move the mouse cursor to the default button for certain dialog windows and optionally press the button.

There is a check box and edit boxes to control Auto Minimize Mode. and Auto Hide.

You can specify that **Stiletto** should force settings for Explorer view and arrangement as detailed below.

You can specify that **Stiletto** should close any explorer window after you (double) click on that explorer window to launch a command.

You can also enter an omit list of comma-separated strings which will be deleted from the list of active windows and Active Task Buttons or which can be used to remove active windows entirely from these two.

There are a set of edit boxes:

- windows where the default button is be pressed (cursor to default button must be checked too)
- windows to be automatically hidden when run.
- windows to be minimized to the tray. instead of the task bar
- windows to appear on every virtual desktop (and never to be autominimized)
- windows to show on menus of active windows and the active bar if hidden and the corresponding menu setup checkbox is grayed.

For all of these edit boxes select the windows by caption or by exe name. You can type the whole window caption, or you can specify captions starting with xxx by xxx*, and captions ending with xxx by *xxx,. You can specify all windows for program filename.exe with =filename, (no path, no .exe.)

Automatically Moving the Mouse Cursor to a Dialog Button

Check the "Cursor to default button" checkbox on the special dialog to have **Stiletto** automatically move the mouse cursor to default button on a dialog.

You can omit certain dialogs by including their captions in the edit box beside the checkbox. You need not enter the whole caption: enter xxx* for captions starting with xxx or enter *yyy for captions ending in yyy.

You can have **Stiletto** automatically push the default button by including the caption of the window in the "Press default button" edit box at the bottom of the dialog. **Stiletto** will wait for 1 second before pressing the button by default; you can change this wait time with the internal PressDelay option.

If you gray check the checkbox, **Stiletto** only moves mouse cursor and presses the default button for captions specified in the "Press default button" edit box

No Click Command Execution

You can use the Buttons configuration dialog to specify that the commands on a button should be executed when the mouse is over held over the button without clicking.

Using the check boxes on this dialog, you can specify which type of commands to execute: only *Show Menu/*Folder Contents Menu commands, active bar buttons, or everything other than menus or active bar commands. Gray check the Menu checkbox to cause menus to be closed automatically if you move the mouse cursor to another button on the bar

You can use the spin box to specify the time in milliseconds during which the mouse cursor must stay over the button for the command to be executed; separate timers are available for active buttons and non-active buttons.

There is a drop down list to specify the no-click command execution default.

Stiletto normally uses the command associated with left button clicking. However, if the ctrl key is held down, **Stiletto** will look at the right button command (left if right is chosen as default), and if the shift button is held down, **Stiletto** will look at the middle button command (left if middle is chosen as default).

For active buttons, if Ctrl is held down, **Stiletto** will close the associated application.

Specifying Explorer View and Arrangement Settings

You can affect the view (large icon, small icon, detail, list) and arrange (date, name, type, size) settings for Explorer in two ways: you can force the settings for all cases using drop down boxes on the special configuration dialog, and you can change the settings for specific cases by sending keystrokes to Explorer windows.

To force the same settings for all newly-opened Explorer windows, use the drop-down boxes on the Window Control configuration dialog. Set the first drop down to **No, Single, Double, or All** to select which types of Explorer Windows to force, then select the desired view and arrangement options. These forced settings will normally override all folders, including the last 50 opened where Explorer also stores a setting, but if you hold down the shift key while opening the new window, **Stiletto** will not override the Explorer settings. As well, you can use the "except" edit box on the Window Control configuration dialog to stop settings from being forced for windows with these captions. In this edit box, separate captions by commas, use xxx* for captions starting with xxx and *yyy for captions ending in yyy.

For a convenient way to change the settings for Explorer windows while you are working with them, send keys to the active window (of course, you can use the tool bar as well). For example,

Command *Send Keys

Parameter * "a-v i d"

sends **Alt-V**, then **i**, then **d** to the active window which would set date sort arrangement for Explorer. You could attach the above command to a hot key or a menu attached to a hot key.

You can also use start Explorer at a specific folder and with specific settings as follows:

Command: c:\windows\explorer.exe

Parameters /select,D:\Program Files\euadora 3\Attach*.*<*send +**attach "a-v g"

This command launches Explorer and uses the Explorer command parameters to select folder **D:\Program Files\euadora 3\Attach**. It then sends key strokes **Alt-v g** to select large icon settings. The **+**attach** tells **Stiletto** to wait until a window with caption ending in **attach** appears before sending the keys.

You could create a menu of commands like the above for favourite folders.

If you send keys to Explorer when it is launched from **Stiletto**, the settings will replace any settings forced by the Window Control dialog.

Tracking Recently Opened Explorer Windows

Use the built-in *Explorer Windows command to re-open a folder that you recently used with Explorer, or to close or minimize all currently open Explorer Windows.

You must check the Windows Explorer option "Display the full path in the title bar" on Explorer View|Options.

If you then check "Track Explorer Windows" on the Window Control configuration dialog, **Stiletto** will remember the last 32 file folders that you open with Explorer. Activating the command

Command *Explorer Windows

Parameter

displays a menu these folders sorted by path. Select one to re-open Explorer for that folder. You can change the sort by putting **recent** in the parameters box to sort by most recently accessed or **drive** to sort by drive and then most recently accessed folder within drive.

Stiletto will remember whether you used a single or double pane Explorer window and use this configuration. If you wish, you can force a single pan window by holding down **shift** when you select a folder name from the menu, or you can force a double pane window by holding down **ctrl**.

You can reduce the number of explorer windows listed in the menu by putting the internal option NumTrackExplorer=n under [General] in your stiletto.ini file.

If the Explorer Windows command is part of a menu, you can embed its contents in the menu by putting **embed** in lower case in the work directory edit box.

You can clear the history of explorer windows with the command:

Command *Explorer Windows

Parameter new

Put this command in your startup menu to clear the history each timer **Stiletto** starts.

You can also use this command to close, minimize, or show all open Explorer windows. This command will close all open Explorer windows:

Command *Explorer Windows

Parameter close

This command will minimize all windows:

Command *Explorer Windows

Parameter min

This command will restore all minimized Explorer windows:

Command *Explorer Windows
Parameter show

Automatically Running Commands when Windows Open

You can automatically run commands when a window with a specified caption is first created.

Use a menu to do this. Each item on the menu corresponds to a command you want to run when a window opens. The menu item name specifies the caption of the window. Use `xxx*` as a menu item name to match any captions starting with `xxx`, `*yyy` to match any captions ending in `yyy`, and `*zzz*` for captions containing `zzz`.

Specify the menu name on the Window Control configuration dialog. Once this is done, each time a new window is opened and the caption matches a menu item name on that menu, **Stiletto** will execute the corresponding command from the menu.

To press specific buttons on the windows, use send keys to send alt-x, when x is the button mnemonic letter.

You can prevent a command from being executed for a window by holding down the shift key while the window is opened.

You can also specify a menu item name of `=exename` to match any window created by the program with .exe file name `exename` (no path, no .exe).

If you only want to run commands if the new window is a dialog, precede the caption/path with a `#`. If you only want to run the command if the new window is **not** a dialog, precede the caption/path with a `~`.

If you want the command to apply to single pane explorer windows (folder windows) only, precede it by an `!`. If you want the command to apply to 2-pane explorer windows only, precede it by an `@`.

If you want the command to run only if no existing window matches, precede the caption/path with a `$`.

Menu Setup

Purpose

This dialog controls what appears on menus listing active tasks, overall menu appearance, startup and shut down menus.

Configuration

The Menu Setup Dialog is selected by clicking on the Menu Setup tab from the Configure Stiletto dialog.

Control of Lists of Active Windows in Menu

You use these configuration controls to set a prefix string and whether or not hidden and tray-iconized windows are displayed .

Startup and Shutdown Menus

Using the drop down boxes, you can specify that the commands on a menu should be run each time **Stiletto** starts and you can specify a menu of commands to be run by **Stiletto** each time you shut down or log off Windows.

Menu Icons

For Win95/NT 4, you can specify that **Stiletto** should include icons on menus. You can specify special background or text color for **Stiletto** menus by checking the own box and selecting a color using the button. You can customize the layout of such menus with icons. You can omit icons from individual menus

For menus with icons, you can have **Stiletto** hold icons in a memory cache by checking the "Cache Icons" checkbox. Gray check for a larger cache. Caching icons will speed menu display but will take more memory.

Menu Position and Display CheckBoxes

You can specify that **Stiletto** always attempt to put menus at left of mouse cursor.

You can specify that **Stiletto** display a menu attached to a bar button as soon as the mouse is clicked down on the button. Gray check to select item as soon as mouse button released (ie to access item, you click down, move mouse over item, then release). Both Left and Right for Middle on Bar dialog must be unchecked or gray checked to enable this item.

You can force the mouse cursor to be over any Stiletto menu when it is first created; this is useful when the next item (force menus closed) to avoid closing the menu prematurely.

You can specify that menus should close automatically if the mouse cursor is moved off them for a specified time (set time 0 to disable).

Menu Color/Font

You can specify the font, background color, text color for menus by checking Own and Font/Color and pressing the appropriate button to select the color or font. Press "Use Win Font" to return to the standard windows font.

Further Information

(You can specify menus to appear when click the desktop or a part of a window. You can also specify the offset of these menu relative to the cursor to pre-position the cursor over a part of the menu. However, these settings are intended for compatibility with older configurations: use mouse action hot keys instead if you want this functionality for new configurations).

Running Commands at Stiletto Startup

You can specify that **Stiletto** should run a set of commands when its starts by putting the commands you want to run on a menu and specifying that menu on the Menu Setup Dialog.

Menu "0" cannot be used as the Start Up menu.

Note: if you hold the Shift key down when **Stiletto** starts, the startup menu will be ignored. This can be used, eg, to avoid an unwanted hide **Stiletto** (Bar Size 0) command.

Omitting Icons from some Menus

You can omit icons or text from individual menus when you have enabled icons for menus using the Menu Setup dialog.

To omit icons, insert a command at the start of the menu with the command edit box set to
*No Menu Icons

This command must be the first in the menu. The menu item name can be anything.

To omit text, insert a command at the start of the menu with the command edit box set to
*No Menu Text

This command must be the first in the menu. The menu item name can be anything. You can limit text to the first **n** characters by using this command with **n** in the parameters edit box.

You can also omit icons from Folder Contents Menu commands (which are not embedded in other menus) by putting **Nolcons** anywhere in the work directory edit box.

Accessing a Menu via Mouse Click

You can access a menu by clicking on the desktop or in a specified position of any window (or both). Use the Menu Setup dialog to specify which menu and when you want it to appear for clicks on either the desktop or a title bar or both

You can select how to make a menu appear by clicking on desktop:

Right:	right click on desktop
Middle	middle click
Right Center	click in middle 2/3 of screen avoids overlaps with Win95 menus

You can also select to make a menu appear by clicking on any window (including the desktop window):

Title Right	right click in non-client area title bar
Title Middle	middle click on non-client area bar
All Shift-left	left click anywhere when shift key held down
All Shift-Middle	middle click anywhere when shift key held down
All Shift-Right	right click anywhere when shift key held down
All Middle	middle click anywhere
All Right	right click anywhere
All Center Right	menu appears with right click in middle 2/3 of screen

If you only have a two-button mouse, you can attempt to use-right click only and insert the Stiletto Exec mouse right command in the menu to access the right menu of the underlying window's program. Note that plain right clicking for **Stiletto** menus is disabled over certain windows, eg Explorer, since it would interfere with right drag and drop.

The following types of commands are useful for menus which appear over any window:

- list of active windows
- access the menu bar of the current window
- Close/min/max/hide/on top/not on top/back window under mouse
- Sending menu accelerator keys to the active window to access frequently used menu items or special scrolling keys (eg *send * c-ho sends ctrl-home which often scrolls to top of document).
- Specifying that certain subsets of the menu should appear depending on the which program's window is under the mouse.

You can also make menus appear through hot keys and mouse actions by attaching a *Show Menu command to the hot key.

Menu Contents Tab

Purpose

Stiletto allows you to create menus of commands. When you select one of the menu entries, **Stiletto** runs the corresponding command.

One possible use for menus is to use them to access commands or files you use less often than the commands you assign directly to a button. Using this approach, you assign a few, most-used commands directly to buttons and many less-used commands to menus.

You may also find it helpful to show menus from hot keys.

Configuration

There are two steps involved in using menus: creating them with the Menu Contents dialog and then showing them by running the *Show Menu command.

One possible use for menus is to use them to access commands or files you use less often than the commands you assign directly to a button. Using this approach, you assign a few, most-used commands directly to buttons and many less-used commands to menus. Of course, this is just one idea; there are many other approaches (eg some people use no button bar and put all commands on menus).

You create or change menus with the Menu Contents tab, which is displayed when you click on Menu Contents on the Configuring Stiletto dialog, or when you click on a button which is assigned a *Show Menu command with the Alt key held down.

You run commands from menus with the built-in *Show Menu command. A **Stiletto** menu is displayed when you click a **Stiletto** button which has been assigned this command with a command parameter set to the menu name. Or you can assign the *Show Menu command to a hot key.

At the left of the menu contents tab of the configuration dialog is a list box which displays the currently selected menu. Submenus are shown indented. You can select which menu is displayed with the drop down box labeled Menu Name. You can add or rename (up to 50) menus with the add or rename buttons. You can delete menus by deleting each individual menu entry or by using the Delete All Entries button (the menu name is removed when you exit the configuration tabbed dialog).

To add a new menu entry, click New Before or New After and set the menu entry using the command entry controls at the right to the dialog. To change a menu entry, click on the entry in the list and then change it using the command entry controls at the right to the dialog.

When you click on a button which is assigned a menu command, **Stiletto** will display the menu: You can then release the button and select the desired item or, alternatively, move the pointer to the desired item and release. If you prefer the second method, **Stiletto** has an internal option to disable the first method.

Submenus

The Menu Contents dialog allows submenus to be created. When a menu is activated, the titles for these submenus are shown in the main menu. If you select one of these submenu titles, the submenu will be displayed and you can then select one its entries.

The entries for submenus are shown indented in a menu list. The start of the submenu is marked by >> in the left margin. The end of the submenu is marked by << in the left margin.

To create a new submenu:

- Press Add Before or Add After.
- Select the Begin Submenu built-in in command.
- Set the menu name to the submenu title.
- Insert all commands for the submenu.
- Press Add After
- Select the End Submenu command.

To remove a submenu:

- delete the Begin Submenu and End Submenu commands.

To move existing commands onto a submenu:

- add a Begin Submenu command before the existing commands
- and add an End Submenu command after the existing commands.

Program-Specific Menu Contents

You can specify that portions of a menu should only appear if specified window or program is active by using the *Start Context Menu and *End Context Menu built-in commands.

These commands are useful, for example, to set up menu entries to send keys in a **Stiletto** context (window) menu or hot key which depends on the active window under the mouse. Or they could be used on a button bar attached to active window for the same purpose.

You can create menus which include several different *Start Context Menu sequences so that different parts of the menu appear for each of the selected programs.

To create a program-specific portion of a menu, you insert a *Start Context Menu command into the menu. In the parameters edit box, put a list of window captions and exe file names. Use *xxx for captions ending in xxx, yyy* for captions starting with yyy, *zzz* for captions containing zzz, and =exename for all windows from the program with .exe file exename (no path, no .exe). Follow this command by the program-specific menu entries. End with the *End Context Menu command.

The following illustrates a set of menu entries to send control-I (view images) and Ctrl-arrow-left (go back) only if Netscape Navigator (netscape.exe) is active.

Menu Item Name: Netscape only
 Command *Start Context Menu
 Parameter =netscape

Menu Item Name: Images
 Command *Send Keys
 Parameter * ^i

Menu Item Name: Back
 Command *Send Keys
 Parameter * ^{al}

Menu Item Name: End
 Command *End Context Menu
 Parameter

You cannot use these commands in menus attached to the desktop.

Displaying a Menu Centered or Offset from the Mouse Cursor

To help position the menu near a most-used entry, you can specify a menu offset in the `*Show Menu` built-in

command. The format is:

```
Command    *Show Menu
Parameter  MenuName *x y
```

where x and y are the horizontal offsets, in screen pixels.

To position the menu at an absolute screen position, use

```
Command    *Show Menu
Parameter  MenuName *a x y
```

where x and y are the position, in screen pixels. The numbers x and y can be positive or negative. Note letter a after the asterisk.

You can center a menu on the screen with

```
Command    *Show Menu
Parameter  MenuName *center
```

Note the asterisk before center and x y.

You can also center the menu under the mouse

```
Command    *Show Menu
Parameter  MenuName *under
```

Note the asterisk before the word under.

Customizing Layout of Menus with Icons or Colors

You can customize the look of menus with icons by typing options directly into your stiletto.ini file under the [general] section.

```
MenuLayoutTop=n          sets the vertical space between entries to n pixels
MenuLayoutLeft=n         sets the left margin to n pixels
MenuLayoutSep=n          sets the separation between the icon and text to n pixels; used even if
                          there is no icon (ie you are just changing menu color)
```

```
MenuIconLarge=1          uses large icons instead of small icons
```

```
MenuIconsDefault=n       uses the nth system icon if a file has no icons; use -3 for no icon
```

```
MenuIconBuiltin=n        uses the nth system icon for Stiletto built-in commands; use -3 for no
                          icon and -2 for the Stiletto icon
```

Note: experiment with different small numbers (bigger than or equal to zero) to see the choices for system icons.

Displaying a Menu as a Toolbar

You can display the commands in a menu on a toolbar. Clicking the toolbar button runs the corresponding menu command.

The following shows a menu named xxx as a toolbar at screen position x y:

```
Command    *Show Menu
Parameter  xxx * x y
Work       bar
```

Drag the border of the displayed bar to resize it.

You can remove the caption on a displayed toolbar by right clicking any button on the toolbar. Right click again to restore the caption.

You can control the format of the toolbar by putting keywords into the work directory of the *Show Menu command:

vertical	shows the bar vertically
noflat	shows a bar with raised buttons
top	shows bar always on top
under	show text under buttons instead of beside it
notoolbar	show bar as normal window which will show up on taskbar
scroll	add scrolling arrows to bar
samesize	force buttons to be equal-sized (if omitted, button size varies to accommodate text/ icon)

You can eliminate icons on buttons by putting the *No Icons command at the start of the menu.

For example:

Command	*Show Menu
Parameter	mybar * 100 0
Work	bar vertical top

shows menu mybar as a vertical, on top bar at position 100, 0.

The scroll and samesize options require that you have at least version 4.71 of the Windows common control library comctr32.dll. The **Stiletto** info tab shows if you have this version. If not, one way to obtain it is to download and install IE4.

Pinning (Permanently Displaying) a Menu

You can Pin (permanently display) menus with the *pin menu built-in command. Pinned menus are displayed in a list box which fills a tool-bar style dialog box. Double clicking on a menu selection runs the menu item. To pin a menu called **mymenu**, execute this command:

Command:	*pin menu
Parameter:	mymenu

If you want to execute a command to pin a menu while it is displayed, you can include a command like the above within the menu.

Right clicking on the menu list box displays a configuration menu which can be used to show or hide the caption, to remove or reshown the sizing border, to put the menu always on top, and to close the menu. You can also set these options as well as the menu size with the *pin menu built-in command by putting an asterisk after the menu name in the Parameters edit box, and following the asterisk with any of these options:

nocaption	removes the caption
noborder	removes the sizing window border
single	single clicking launches command
ontop	put menu always on top
x123	show the menu at 123 pixels from the left of the screen
y22	show menu 22 pixels from the top of the screen
w50	set menu width to 50 pixels
h75	set menu height to 75 pixels

You must specify both height and width (h and w) if you want to use either of these options.

For example,

Command: *pin menu

Parameter: xyz * x10 y50 w100 h150 noborder nocaption

pins menu xyz at screen position 10, 50 and sets its width to 100 and height to 150.

Pinned menus will display icons if menus have been set up to do so. To avoid the icons, put *No Menu Icons as the first command in the menu. On the other hand, to remove text (and show only icons), put *No Menu Text as the first command.

Submenus, lists of active windows, new menu columns, and context menus are ignored when displaying pinned menus.

Time Setup

The Time Setup dialog is displayed when the time setup tab is clicked from the [Configuring Stiletto](#) dialog.

Using this dialog you can set check boxes to:

Have **Stiletto** ring alarms which occur when **Stiletto** is not active. Otherwise, missed alarms are not rung but are recycled or discarded according to the alarm setting. (However, alarms less than four minutes old are always rung).

Play the alarm sound when an alarm displays a message.

Play the alarm sound when a command is run by an alarm.

Keep a [timer log](#).

Keep an [alarm log](#).

Specify that seconds should be shown on time/date labels.

Specify that seconds should be shown on [timer](#) labels.

Specify that chimes should not be rung while a screen saver is running.

Specify that a ringing alarm should stop any running screen saver.

Specify whether or not alarm messages should be shown on top of the active window when the alarm rings. Gray-check to specify messages to be shown "always on top".

The Time Setup dialog also contains several drop down lists which you use to:

Set the format for dates in the alarm list.

Set the screen position for alarm message windows.

Set a chime at a regular time during the hour (eg every 15 minutes).

Play the alarm sound at regular intervals for any [timer](#).

Save [timers](#) at regular intervals so that **Windows** crashes do not cause autosave timer information to be lost.

Set a resource warning level percentage to have **Stiletto** display a message box whenever GDI or USER resources fall below this level. You can also monitor resource [usage](#) with a button label set by the [Configure Button](#) dialog.

Set the interval in seconds at which **Stiletto** updates [timers](#), checks the resources, and updates any resource usage button labels set by the [Configure Button](#) dialog.

Access the dialog to view or change [timers](#).

Use the [Sound](#) dialog to set the sound associated with alarms and chiming.

Alarm Details

The Alarm Details dialog is displayed when you click on the Alarm Details tab of the [Configuring Stiletto](#) dialog.

Use this dialog to set alarm times and the command or messages to be activated when the alarm rings. **Stiletto** only checks to see if an alarm should be rung once per minute. If you set an alarm for now, it will not ring until the next minute.

On the alarm configuration dialog, the left side of the dialog shows a list box containing all current alarms, sorted so that the earliest is at the top. (You can change the format of the dates in the list using the Timer Setup dialog). To add an entry, click New and then set the alarm command and time using the controls at the right. To change an entry, select the alarm and use the controls at the right of the dialog.

Enter a message date and time using the spin boxes.

You can use alarms to run programs after the system has been idle for a specified time by putting -1 in the alarm year and the idle time in the hours and minutes.

You can set alarms to be rung when **Stiletto** starts by setting the year to -2.

To quickly add a new message (reminder) alarm, see [Adding a Reminder](#).

You can automatically close Messages with alarms after a specified time period by putting the number of seconds the message is to appear in the parameter box. For example, to close the message after 5 seconds:

Command:	*Message
Parameter	5
Work/Message	The message for the alarm

When a message box alarm rings, you can change the message text and re-schedule it, if you like. You can select the time until the next alarm from a drop down box or by entering at as months:days:hours:minutes. When the alarm message is shown, for message box alarms which you show repeatedly, you can request that the message alarm be copied and shown again in 5 minutes as well as being saved to be shown again at the original time.

If helpful, you can create notes for a command by putting a semi-colon followed by the comments in the work directory (after any directory to be used with the command).

You can show messages always on top using the drop down on the command entry controls or by setting always on top as the default on Time Setup configuration.

Using Alarms to Close or Wait on Active Tasks

You can use alarms to close active tasks or to wait for active tasks to complete.

If you create an alarm with the work directory set to the string `*close` (or `*close force`), then the command will be closed when the alarm rings, if it is active (nothing happens if the command is not active when the alarm rings). If you use `*close`, the program may refuse to close; but if you follow `*close` by force (in lower case), **Stiletto** will attempt to force the window closed, at the risk of losing information. Do not use `*close force` with Dos programs.

If you create an alarm with the work directory set to the string `*wait`, then no further alarms will be processed until the command terminates (nothing happens if the command is not active when the alarm rings). You need to make sure the "Ring Missed Alarms" option is checked in the time setup dialog for `*wait to work`.

Close and wait alarms do **not** work with Dos programs; they only work with Windows programs.

Here is an example of the use of close and wait alarms: Suppose you have one modem which you use for two programs: a fax receiver (program `/winfaxx/winfaxx.exe`) and an offline mail reader with a script which automatically dials your Internet provider and downloads your mail (program `/fedora/fedora.exe myscript.hat`). Now suppose you want the fax reader to be always running on your machine except for 2:00 AM when you want your mail reader to download your mail. Your problem is that both programs cannot be running at the same time since they both use your single modem.

The following sequence of alarms will solve your problem by closing the fax program at 1:59, starting the mail reader at 2:00, and re-starting the fax program when the mail reader finishes.

Time	Command	Work Directory	After Alarm Rings
1 59 AM	<code>/winfaxx/winfaxx.exe</code>	<code>*close</code>	Alarm again in 1 day
2:00 AM	<code>/fedora/fedora.exe myscript.hat</code>		Alarm again in 1 day
2:01 AM	<code>/fedora/fedora.exe</code>	<code>*wait</code>	Alarm again in 1 day
2:02 AM	<code>/winfaxx/winfaxx.exe</code>		Alarm again in 1 day

Stiletto close and wait alarms are not intended to replace an "industrial strength" scheduling program. For example, they cannot handle situations where programs encounter errors or have unsaved data and so refuse to close. You are cautioned to use close and wait alarms only for non-critical data where it is easy to recover manually if exceptional circumstances arise.

Final note: if you believe that the user interface for close and wait alarms is ad hoc, ugly, and not Windows standard, you have the full agreement of the author.

Suspending Alarms

You can suspend ringing of alarms by executing the following command (eg though a button or menu item):

Command *Stiletto Exec
Parameter: alarms off

To resume alarm ringing, use

Command *Stiletto Exec
Parameter: alarms on

To reverse the status, ie suspend alarm ringing if it is active, or resume alarm ringing if it is suspended, use

Command *Stiletto Exec
Parameter: alarms toggle

To prompt for the change in alarm status, use:

Command *Stiletto Exec
Parameter: alarms ?

When alarm ringing is resumed, alarms which would have rung when alarm ringing was suspended are rung or discarded according to the setting of "Ring Missed Alarms" on the Time Setup configuration dialog.

Adding Reminder Message Alarms

You can quickly set up a reminder message or a command to appear at a specified time with the built-in command Add Reminder. This command displays a dialog box which provides a short cut way to add an alarm to display a message box or run a command or both.

Enter a message date and time using the spin boxes. After you have entered a valid time, date, and at least one character in the message edit box or the command, the "Add" button will be enabled and you will be able to save the reminder message.

When the alarm time occurs, the command (if any) will be executed and the message (if any) will be displayed. Put the command in quotation marks if it contains blanks.

The command can be the name of a wave file to play a sound when displaying the message.

You can use the After Alarm Rings drop down list to have the message automatically re-displayed at regular intervals.

You can specify that the message should be displayed using the Add Reminder dialog (rather than a plain Message Box) which will give you more flexibility in choosing the next alarm date/time if you decide to recycle the alarm.

Alarm Log

You can ask **Stiletto** to log alarm events by using the Keep Alarm Log check box on the Time Setup dialog.

The log file will have the same name as the configuration file used in the instance of **Stiletto**, except that the file extension will be .alo. For example, the log file for the default configuration is stiletto.alo. The log is always placed in the same directory as the **Stiletto** ini file. A log file entry will be written whenever an alarm rings. It will consist of the following fields, separated by blanks:

Current Year
Current Month
Current Hour
Current Minute
Current time expressed as seconds since 1970 01 01.
Alarm Year
Alarm Month
Alarm Hour
Alarm Minute
Alarm time expressed as seconds since 1970 01 01.
Alarm command and parameters.
Alarm work directory/message.

Sound Tab

The Sound dialog is selected by clicking the Sound tab on the [Configuring Stiletto](#) dialog

The Sound dialog is used to change the sound file (wav file) associated with any of the **Stiletto** [sounds](#).

To change sound information, select the sound entry and use the controls at the right of the dialog.

To add spice to your multimedia life, **Stiletto** can automatically change the sound file you select to any other wav file in the same directory. Use the How To Change Sound drop down box to determine whether and how **Stiletto** changes the sound file. Use the Mins Between Changes drop down box to determine how often **Stiletto** changes the sound file (this drop down is disabled if you selected No Change from the How To Change Sound drop down).
replaced.

The **Stiletto** noise sound is played at random in the percentage of minutes set by the “% of mins to play noise” edit box. Use this feature if you feel drowsy. The noise sound is changed each time it is played, unless the change method is set to “No Change”.

You may wish to group your wav files into directories (eg startup, shutdown, good, bad) and then use the **Stiletto** automatic sound file changer to sample the sounds from each directory.

The Change Sound [built-in](#) command changes any sounds which have not been marked as No Change. If you assign this command to a button or put it on a **Stiletto** menu, you can use it to have **Stiletto** immediately change sounds if a sound pattern that you dislike is selected.

Stiletto Sounds

Stiletto sounds are set from the [Sound dialog](#).

Stiletto supports the standard Windows sounds in the Registry plus these sounds:

- | | |
|-----------------------|--|
| Stiletto Chime | Plays whenever Stiletto chimes (see Time Setup dialog) |
| Stiletto Alarm | Plays whenever Stiletto alarms (see Time Setup dialog). Use an asterisk to have the PC Speaker beep for alarms. |
| Stiletto Noise | Windows background noise: sound plays at random minutes according to the percentage set in the Sound dialog. |

Paper/Saver Tab

The Paper/Saver dialog is selected by clicking the Paper/Saver tab on the Configuring Stiletto dialog.

The dialog is used to change the Windows wallpaper or the screen saver.

To specify a wallpaper file, type its name directly into the wallpaper file name edit box, drag and drop it from the Explorer/File Manager, or use one of the browse buttons to display and select from a list of bmp files. If you want to see the wallpaper as you browse the files, press the Browse and Display button; clicking on a file in the file browse dialog will display that wallpaper.

To add spice to your multimedia life, **Stiletto** can automatically change the wallpaper you select to any other bmp file in the same directory. Use the Method for Changing Wallpaper drop down box to determine whether and how **Stiletto** changes the wallpaper file. Use the Mins Between Automatic Changes drop down box to determine how often **Stiletto** changes the wallpaper file (this drop down is disabled if you selected No Change from the Method for Changing Wallpaper drop down). You can also use shortcuts in the folder of wallpaper files.

Use the "Select Random Directory from Parent Folder" check box to indicate that **Stiletto** should select a random folder from the parent folder of the current wallpaper file's directory before changing the wallpaper file. Use this feature when you have various wallpaper "themes" arranged in directories under a parent wallpaper folder. This random folder is only selected when **Stiletto** starts.

Use the Tile Wallpaper check box to set whether Windows tiles or centers your wallpaper.

Stiletto has a built-in Show Wallpaper command for previewing wallpaper.

You can also associate **Stiletto** with the bmp extension using Explorer/File Manager. Double clicking on a bmp file in Explorer/File Manager will then preview that file as wallpaper.

If you have registered **Stiletto**, you can use jpeg files as wallpaper. If you check the Include jpeg files checkbox, any file with type .jpg or .jpeg in the wallpaper directory will be usable as wallpaper.'

To specify a screen saver file, type its name directly into the saver file name edit box, drag and drop it from the Explorer/File Manager, or use the browse button to display and select from a list of scr files. If you want to try the screen saver press the test button.

Use Control Panel to configure the screen saver.

To add spice to your multimedia life, **Stiletto** can automatically change the saver you select to any other scr file in the same directory. Use the Method for Changing Screen Saver drop down box to determine whether and how **Stiletto** changes the screen saver file. Use the Mins Between Automatic Changes drop down box to determine how often **Stiletto** changes the file (this drop down is disabled if you selected No Change from the Method for Changing screen saver drop down). If you use this feature, you will speed **Stiletto** search and initialization by copying the .scr files you want to use to a separate directory and working with that directory.

Hot Keys and Mouse Action Commands

The Keys/Mouse dialog is displayed when you click on the Keys/Mouse tab of the Configuring Stiletto dialog.

With it, you can define any of the following hot keys/mouse actions to launch commands:

- keyboard keys with optional modifier keys (shift, alt, ctrl, win)
- semi-colon followed by a keystroke (no modifier key)
- tapping ctrl, alt, shift, caps lock, win
- moving the mouse to a screen corner or screen edge
- clicking or chording mouse buttons with optional modifier keys (shift, alt, ctrl, win)
- moving the mouse back and forth horizontally or up and down vertically
- clicking on desktop
- clicking on the window caption:
 - anywhere
 - left half
 - right half
 - system menu box
 - mimize box (in title bar)
- clicking on a window border
- pressing and holding a mouse button
- double clicking middle mouse button (corresponding single click must also be a hot key).

The left side of hot key configuration dialog shows a list box containing all hot keys and mouse action commands. Press New and use the controls at the right to add a new entry. Select an entry and use the controls at the right of the dialog to change the entry.

You can assign hot keys/mouse actions which run only when a specified windows are active by using the Target Window edit box. Leave this edit box blank to have the hot key apply to any window. You can have up to two bars with independent hot keys: for example, you could have hot keys defined on one main bar as well as hot keys defined on a context-sensitive tool bar.

You can create global macro keys to paste text phrases or paragraphs with the Send Keys built-in command and possibly the Clipboard commands.

You can assign double click to a mouse action by associating the mouse action with the *Stiletto Exec mouse command.

The win modifier key is also used internally by Windows; you cannot redefine hot keys that Windows has already defined.

Note on chording: some mouse drivers "miss" the second mouse up when two mouse keys are released at once leading to strange mouse behaviour; to clear, you may have to press and release each mouse key separately.

Use the Key Setup tab to fine tune hot keys and mouse actions:

You can specify a delay in milliseconds for the screen corner/edge commands; the command will only be executed if you leave the mouse cursor in the corner for at least the specified delay.

You can specify a delay in milliseconds for the tap key commands; the command will only be executed if hold the tap key down for **less** than the specified delay time. You can also specify that tap keys will only be activated by a double tap.

If helpful, you can create notes for a command by putting a semi-colon followed by the comments in the work directory (after any directory to be used with the command). Avoid the asterisk in such comments.

You can specify the minimum length and maximum deviation from horizontal/vertical for mouse move hot keys.

Program-Specific Hot Keys

You can define hot keys which only function when a specified program is the active window. To do so, define a hot key as usual, but add an asterisk followed by the .exe file name of the program to the work directory edit box of the hot key command.

For example, the following command definition sends the key sequence Alt-F S Alt-F4 to NotePad only (this sequence saves the active file and then exits):

```
Command:    *Send Keys
Parameter:  * "a-f s a-f4"
Work Dir    *Notepad
```

Note that you must use the .exe file name of the program, eg Excel for Microsoft Excel, iexplore for Microsoft Internet Explorer, Netscape for Netscape navigator. The directory path and .exe extension are optional.

Press the "See Target" button at the top of the dialog to see the program assigned to the hot key in the list of hot keys.

If you also want to put a working directory path in the Work Dir edit box, put the * exe name after the path name.

You can define the same hot key several times if you want to use the same command for several programs or you can define the same hot key to mean different things in different programs.

You can define a hot key to have specific meaning for certain programs and other meanings for other programs by defining the hot key multiple times with different Commands and Work Dir entries. Omit the * .exe name from the Work Dir for hot keys to be applied to all programs.

When you press a key which is a hot key, **Stiletto** uses the following searches to select from the possibilities:

First, search to see if there are any hot keys defined solely for the currently active program. If so use it or them (if more than one hot key is specifically defined for this key and program).

If there are no hot keys specifically for this program, but there are hot keys(s) for all programs, execute them.

If the only hot keys which are defined are specific to other programs, then send the raw input key to the currently active program

Command Entry Controls

Stiletto uses the same set of controls to enter commands for button configuration, menu, hot keys, timers, and alarms.

With these controls you enter the file to be run when you press a button, or select a menu item, or when an alarm rings. You can enter a program file (.exe file) or a file name with an associated command (eg .xls file for Microsoft Excel). **Stiletto** provides many ways to find the file name: you do not need to type it in. You can:

- browse for it
- drag and drop it from file manager or Explorer
- capture it from
 - a currently running program
 - or your Start Menu and its submenus (for Win 95/NT4)
- cut and paste it from a program manager command

You can also select a built-in command.

Right click on the command entry box to select a built-in command.

You can enter long file names with blanks in the command edit box. Do not use quotation marks.

Enter command parameters in the parameters box. Right click on it to select a menu name for the Menu command or to select a directory name for the Folder Contents command. Parameters are separated by blanks; if a blank is a part of a command parameter, you must surround that parameter by double quotations marks.

You also use the controls to set the start up (working) directory for the command and the initial window size for the command.

You can indicate that **Stiletto** should always launch a new instance of a program, or should switch to an already running instance if it exists. (Note, however, that many programs will only allow one instance of themselves to be launched).

Entering Command Entry Controls Information

To enter the command you want to run:

If you know the file name (ie the .exe file for commands or your file name for documents with associated commands:)

- type it directly into the edit box,
- or Browse for the command using the Browse Button,
- or drag and drop it from Explorer/File Manager onto the command edit combo box.

If you want to use a built in command or a Windows applet like File Manager:

- select one of the built-in commands from the drop down box.
- or right-click on the command-edit box and select from the menu

If you do not know the file name (.exe file) for your command:

- cut and paste it from a program manager group,
- or start the command you want to use and then use the capture button.

Stiletto supports file associations, so you can enter the name of a document file and **Stiletto** will run the associated command. For example, if you are working on a **Microsoft Excel** spreadsheet called mysheet.xls, you can insert this file name as a command (or browse for it and select it). **Stiletto** will run

Excel on mysheet.xls when you select the command or when the alarm corresponding to the command rings

You can put command parameters in the parameters edit box. Or, to prompt for command parameters, put a question mark (?) in the command parameter box. You can put a ? in the midst of other command parameter text, and **Stiletto** will replace the ? with the prompted-for text.

To determine how the window will look when the command starts: Using the drop down box under the command entry combo box, you can determine whether the command starts in an ordinary window, minimized, or maximized. You can also specify that the command windows should start as Always On Top. Finally, you can specify that the window from the command be hidden: this is intended for commands started by alarms.

To set the work (start-up) directory for a command: type the directory name into the work directory edit box or press the button underneath this edit box to set the work directory to the same directory as the command directory.

You also use the work directory edit box to:

Enter a message for the Message Box command built-in.

Enter a default group for the Program Mgr Groups built-in.

Enter a default starting directory and file extensions for the Browse and Run built-in.

Enter *close or *wait for close and wait alarms.

Enter a comment preceded by a semi-colon.

Enter |filepath.wav to play the sound "filepath.wav" when the command is started. If you also want to use a work directory, put the |filepath.wav immediately after the work directory name. (Works for all commands, including built-ins, except for above four).

Running Multiple Commands

If you only need a quick way to run more than one command, you can do so by separating multiple commands with a less than sign (<) in the command entry box and parameters edit box of the command entry controls. (The command wizard will add this character for you).

Put the first command name in the command entry box. Put its parameters and the remaining commands and their parameters in the parameters edit box.

If the command names or files in the parameter box contain blanks, you must put them in double quotations. Do not use quotations marks for the command edit box, however.

For example:

Command: c:\win\calc.exe
 Parameter: <c:\win\notepad.exe
 starts program calc.exe and notepad.exe.

For example:

Command: c:\my\path\prog.exe
 Parameter: "p1 with blanks" p2<"*Bar Size" 6
 runs prog.exe with parameters p1 with blanks and p2 and then changes bar size to 6 buttons (note double quotations).

You can insert pauses between commands with *wait n, where n is a single digit delay in seconds (*wait 0 waits for 1/3 of a second). This is mainly useful with Send Keys built in command, eg

Command: c:\prog\splash.exe
 Parameter: <*wait 4<*send ** "h e l l o"
 starts the program splash.exe, waits 4 seconds, and then sends the key sequence "hello".

Rather than waiting for a fixed number of seconds, you can use ***wait !** to wait for the last command launched by **Stiletto** to request input. The wait will last no longer than 10 seconds. **Stiletto** will be unresponsive while waiting:

Command: c:\prog\splash.exe
 Parameter: <*wait 4<*send ** "h e l l o"
 (Note: wait ! may not work when the command is launched from a shortcut).

There are situations where you only want to execute a sequence of commands when a preceding command is launched initially, not when it is switched to if active (example: sending a series of keys at start up). The built-in command *abifa (abort if active) will stop the processing of a series of multiple commands if a preceding command was switched to, instead of being launched.

If the default separator character causes you problems, or if you want to disable multiple commands, you can do so with the advanced dialog accessed from the Info tab.

Wait Command

Use the wait command in multiple commands or in when executing all commands on a menu in order to wait for some condition before executing some of the commands. Following are the wait options:

Wait for n seconds:

Command: *wait
 Parameter: n
 where n is any number waits for that number of seconds. If n is zero, waits for 1/3 of a second.

Wait until last command executed by Stiletto ready for input:

Command: *wait
 Parameter: ready

(You can also use an ! instead of the word ready).

Wait until last command executed by Stiletto exits

Command: *wait
Parameter: last

Wait for modem to be connected (Dial-Up Networking RAS connection only):

Command: *wait
Parameter: modem

You can also put a number ahead of the word modem; Stiletto will wait for either that number of seconds, or until the modem is connected, whichever is smaller. For example, "8 modem" waits for up to 8 seconds or until the modem is connected.

Wait for modem to be disconnected (Dial-Up Networking RAS connection only):

Command: *wait
Parameter: nomodem

Wait until window with specified caption is active (foreground):

Command: *wait
Parameter: active xxx

waits until any program with caption xxx is active (foreground). Put caption in double quotes if it contains blanks. Use xxx* for captions starting with xxx, *yyy with captions ending with yyy, and *zzz* for captions containing zzz anywhere. You can use multiple captions separated by commas. Use =progrname for any window from program with exe file name progrname (no path, no .exe). You can put a number n ahead of the caption to limit wait to n seconds. .

Wait until window with specified caption is not active (foreground):

Command: *wait
Parameter: noactive xxx

waits until any program with caption xxx is not active. Put caption in double quotes if it contains blanks. Use xxx* for captions starting with xxx, and *yyy with captions ending with yyy and *zzz* for captions containing zzz anywhere. You can use multiple captions separated by commas. Use =progrname for any window from program with exe file name progrname (no path, no .exe). You can put a number n ahead of the caption to limit wait to n seconds.

Wait until window with specified caption is running:

Command: *wait
Parameter: window xxx

waits until any program with caption xxx is running. Put caption in double quotes if it contains blanks. Use xxx* for captions starting with xxx, *yyy with captions ending with yyy, and *zzz* for captions containing zzz anywhere. You can use multiple captions separated by commas. Use =progrname for any window from program with exe file name progrname (no path, no .exe). You can put a number n ahead of the caption to limit wait to n seconds. . (You can use **caption** instead of **window**). Use **visiblewindow** to ensure the window is visible.

Wait until window with specified caption exits:

Command: *wait
Parameter: nowindow xxx

waits until any program with caption xxx exits. Put caption in double quotes if it contains blanks. Use xxx* for captions starting with xxx, and *yyy with captions ending with yyy and *zzz* for captions containing zzz anywhere. You can use multiple captions separated by commas. Use =progrname for any window from program with exe file name progrname (no path, no .exe). You can put a number n ahead of the caption to limit wait to n seconds. (You can use **nocaption** instead of **nowindow**). Use **novisiblewindow** to omit invisible windows.

Wait until command with specified exe path is running:

Command: *wait
Parameter: path c:\path\prog.exe

waits until any program executed from c:\path\prog.exe is running. Put path in double quotes if it contains blanks. You can put a number n ahead of the path to limit wait to n seconds.

Wait until command with specified exe path exits:

Command: *wait

Parameter: nopath c:\path\prog.exe

waits until any program executed from c:\path\prog.exe exits. Put path in double quotes if it contains blanks. You can put a number n ahead of the path to limit wait to n seconds.

Wait for mouse or keyboard activity

Command: *wait

Parameter: activity

Waits until mouse or keyboard activity. Always waits at least 3 seconds to ignore activity associated with launching the command.

Wait with a message box and a count down timer:

Command: *wait

Parameter: message n text

displays a message box containing **text** and a countdown timer which starts at n seconds. If n reaches 0 or the "Start Now" button on the message box is pressed, then the wait ends and the next command is run; if the cancel button is pressed, the wait ends and all following commands are ignored.

You must put *wait in lower case.

If you reconfigure **Stiletto**, all outstanding waits will be ended.

You can have at most eight outstanding waits.

You can terminate all outstanding waits by running the command:

Command: *Stiletto Exec

Parameter: quitwait

Stiletto Built-In Commands

Stiletto comes with a set of built-in commands.

You will find the built in commands in the drop down box of the **Stiletto command entry controls** on the button configuration dialog, menu dialog, hot keys and alarms dialog. This drop down box also allows you to select one of the Windows applets, like File Manager or Control Panel.

Right-click on the command edit box to select a built-in command from an alphabetically sorted menu.

Starting with version 1.9t, built-in commands are prefixed with an asterisk (*) to avoid conflicts with long file names which include blanks.

You can find explanations of each command in the following sections or, if you know the command name, you can start with the alphabetical list.

Built-in Commands: Menu Structure

Menu	Displays the Stiletto menu whose name is entered in the command parameters field. (Right click on parameters edit box to select menu). This command can also be used to launch many <u>commands</u> . You can control the position of the menu on your <u>screen</u> .
Menu Separator	Separates displayed menu entries; only available for <u>menus</u> .
Start Submenu	Starts a <u>submenu</u> ; only available for <u>menus</u>
End Submenu	Ends a <u>submenu</u> ; only available for <u>menus</u> .
Start Context Menu	Starts a <u>context menu</u> only available for <u>menus</u>
End Context Menu	Ends a <u>context menu</u> only available for <u>menus</u>
New Menu Column	Starts a new column in the menu; only available for <u>menus</u> .
Menu of Win Under Mouse	Copies menu bar of window under mouse (Win95 only); mainly used for window <u>menus</u> .
All Windows to Menu	<u>Fills</u> menu with all active windows.
Start Menu (Win 4)	Displays the Win 4 start menu. If you have problems with the menu not moving properly to your cursor, use the <u>internal</u> option StartMenuDelay to control the delay; you may need to use this feature if Stiletto is an app bar on the <u>Shell/Buttons</u> dialog. Put nomove in command parameters to stop Stiletto moving menu to mouse cursor.
Folder Contents Menu	Displays and allows you to execute the contents of file directories as a <u>menu</u> .
Clear Recent Docs	For NT4/Win95, clears recent docs list.

Built-in Commands: Active Task Manipulation

Control Window	Lets you close, min, max, tray min, rollup, hide, etc any window, all windows, or a window selected from a menu.
Active windows list	Inserts a <u>list of active windows</u> into the menu. When a list item is selected, that window is activated. Only available from <u>menu</u>
Close Windows List	Inserts a <u>list of active windows</u> into the menu. When a list item is selected, that window is ended. Only available from <u>menu</u> .
Window On Top List	Inserts <u>list of active windows</u> into the menu. When list item is selected, that window is set to "always on top". Only available from <u>menu</u> .
Window Not On Top List	Inserts <u>list of active windows</u> into the menu. When list item is selected, that window will no longer be "always on top". Only available from <u>menu</u> .
Minimize Window List	Inserts <u>list of active windows</u> into the menu. When list item is selected, that window will be minimized. Only available from <u>menu</u> .
Recently Run Cmds List	Inserts a list of the 10 most recently run command lines into the menu. When a list item is selected, that command is executed. If the last command was the name of a document, the command line will consist of the associated .exe file followed by the document name (only the .exe file in Win95/NT4). Only available from <u>menu</u> . (Put hidden in command parameters if you wish to include commands from hidden windows.
Close/... Window Under Mouse	Close/min/max/hide/on top/not on top/to back the window underneath the mouse. These commands act on the window under the mouse cursor and so are most useful for Stiletto in to the <u>active window caption</u> or on a <u>menu</u> set to appear when the mouse is clicked on the a window or window title. The action applies to where the mouse was originally clicked, not the menu selection position. If the mouse is clicked on an MDI child window, the action applies only to that window; else it applies to the main, top-level window.
Tile Windows	Tile active <u>windows</u> .
Send Keys	Used for <u>sending keys</u> .
Hide Window	Used to select and then <u>hide a window</u> .
Auto Minimize Mode	Turns off or on <u>autominimize</u> .
Process All	Process all window or a list of windows as specified by parameters edit box. Start parameters edit box with min to minimize, restore to show minimized, close to close, hide to minimize and hide, show to show hidden minimized windows, normclose to close unminimized windows, max to maximize windows. Optionally follow keyword by list of window captions to apply only to those windows. Use xxx* for captions starting with xxx; *yyy for windows ending in yyy.

Built-in Commands: Messages, Alarms, and Timers

Message	Displays a message box containing the text typed into the work directory of the command entry controls. It is intended for alarms, but can be used anywhere.
Add Reminder	Add a reminder message <u>alarm</u> .

View/Change Timer	<u>View/Change</u> the <u>timer</u> information.
Start Timer	Starts a <u>timer</u> .
Stop Timer	Stops a <u>timer</u>
Clear Timer	Zeros a <u>timer</u>
Toggle Timer	Stops a <u>timer</u> if it is running; starts it if it is stopped.
Set Timer	Starts, stops, toggles a <u>timer</u> and resets its value.
Timer Id	Sets the <u>timer id</u> of the <u>timer</u> displayed on the button.
Show System Resources	Shows system <u>resources</u>

Built-in Commands: Stiletto Position and Size

Move Stiletto	Makes the Stiletto button bar <u>move</u> to the opposite side of the desktop.
Position Stiletto	You can drag Stiletto to an arbitrary <u>position</u> by clicking and dragging on a button which has been assigned this command.
Configure Stiletto	Activates the Stiletto <u>Configuring Stiletto</u> tabbed folder. Normally, the Info tab is displayed first, but you can select another tab by entering its index number (0 to 12) in the command parameters field. You can also start the Configuration dialog by left-clicking anywhere on the Stiletto bar with the Ctrl key pressed.
Show/Move Stiletto	Brings every Stiletto button bar to top, making them visible (but does not change whether they are permanently on top). To bring only one button bar to the top, put its configuration file (ini file) name in the command parameters edit box. Put *move in parameters box to temporarily move bar to mouse cursor. Put *toggle to reverse bar visibility.
Hide Stiletto	Hides every Stiletto button bar, making them invisible. To hide only one button bar, put its configuration file (ini file) name in the command parameters edit box
Reset All Stiletto	Restarts all Stiletto bars.
Bar Size	The command Bar Size n changes the <u>number of buttons displayed</u> .

Built-in Commands: Exit Windows or Stiletto

You can use the following keywords in the parameters edit box of Windows Exit commands:

Put **logoff** in parameters box to logoff instead of shutting down.

Put **force** in parameters box to force all applications to close (you will lose unsaved information).

Starting with version 98g, **Stiletto** does a complete shutdown for NT; put **partial** in parameters box to activate partial shutdown behavior of previous versions.

Quick Win Exit	Shuts down Windows (you will get a chance to save unsaved documents).
----------------	---

Quick Win Restart	Exits Windows to DOS mode and then restarts Windows (you will get a chance to save unsaved documents). Only available in Win95, not Win NT.
Quick System Boot	Exits Windows and restarts Windows (you will get a chance to save unsaved documents).
Confirmed Win Exit	After asking you to confirm, shuts down Windows (you will get a chance to save unsaved documents).
Confirmed Win Restart	After asking you to confirm, exits Windows to DOS mode and then restarts Windows (you will get a chance to save unsaved documents). Only available in Win95, not NT.
Confirmed System Boot	After asking you to confirm, exits Windows and restarts Windows (you will get a chance to save unsaved documents).
Exit Win and Exec Dos	Exits Windows to DOS mode, executes the command in the parameters edit box, and then restarts Windows (you will get a chance to save unsaved documents). Only available in Win95, not Win NT.
Exit Stiletto	Terminates Stiletto . Will normally prompt to confirm, but you can avoid the prompt by putting silent in the parameters box.
Windows Shutdown	Activates standard Windows shutdown dialog.
Restart Stiletto	Restarts Stiletto based on the disk image of the ini file; to use a new ini file put its name, including the file extension, in the command parameters (it must be in the same directory as the current ini file).

Built-In Commands: Execute Files, Documents, or Prog Mgr Commands

Start Menu (Win 4)	Displays the Win 4 start menu. If you have problems with the menu not moving properly to your cursor, use the <u>internal</u> option StartMenuDelay to control the delay; you may need to use this feature if Stiletto is an app bar on the <u>Shell/Buttons</u> dialog.
Folder Contents Menu	Displays and allows you to execute the contents of file directories as a <u>menu</u> .
Recently Run Cmds List	Inserts a list of the 10 most recently run command lines into the menu. When a list item is selected, that command is executed. If the last command was the name of a document, the command line will consist of the associated .exe file followed by the document name (only the .exe file in Win95/NT4). Only available from <u>menu</u> . (In rare cases, this command may cause GPFs when used with File Manager; if so, you will need to remove it).
Browse and Run	Displays a file selection dialog; the selected file is executed. Stiletto will follow any File Associations when executing the selected file. Optionally, you can put a default starting directory in the work directory edit box. The browse dialog contains a history of commands/parameters and an edit box for new command parameters. You can also specify the default file extensions; see <u>Browse and Run</u> for details.

Dos Command	Uses the stildos.pif file to execute the <u>dos commands</u> entered in the parameter box.
Tiny Run Box	Displays a small dialog box which you can type a command to be <u>run</u> .
Type and Run	Displays a dialog with an combo box in which you can type the name of a file (with parameters) or document to run. Stiletto will follow any File Associations when executing the entered file. The combo box contains a history of commands/parameters from which you can select the file to be run. You will probably find the <u>Tiny Run Box</u> more effective
Prog Mgr/Shell Grps	Lets you execute a command from any existing <u>program manager or other desktop shell group</u> .
Built-In Cmds List	Displays a list box from which you can execute any of the Stiletto built-in commands. To have the list sorted when the command starts, put the word "sort" in the command parameter field.
Stiletto Exec	Used for <u>close/min</u> , <u>clipboard</u> , and <u>filing</u> , <u>positioning</u> , mouse click <u>sending</u> commands.
Hide Desk Icon/Taskbar	Hides Program Manager Window/Icon for NT; hides desktop icons for Win95/NT4. Put taskbar in parameters edit box to hide taskbar instead. Put taskbar toggle to switch hidden/shown state of taskbar (ie show if invisible; hide if visible).
Show Desk Icon/Taskbar	Shows Program Manager Window/Icon; shows desktop icons for Win95/NT4. Put taskbar in parameters edit box to show taskbar instead. Put taskbar toggle to switch hidden/shown state of taskbar (ie show if invisible; hide if visible).
Send Keys	Used for <u>sending keys</u> .

Built-in Commands: Screen Saver

Start Screen Saver	Starts your screen saver. Put 1, 2, or 3 in parameters box to delay start by 1, 2, or 3 seconds.
Enable Screen Saver/Power	Enables your screen saver if parameter field blank or if it contains the word "saver". If parameter field includes "poweroff", enables power off feature of Energy Star monitor. If parameter field includes field "lowpower", enables low power feature of Energy Star monitor.
Disable Screen Saver/Power	Disables your screen saver if parameter field blank or if parameter field contains the word "saver". If parameter field includes "poweroff", disables power off feature of Energy Star monitor. If parameter field includes field "lowpower", disables low power feature of Energy Star monitor. (Win95 only for power control).
Temp Disable Saver/Power	Disables your screen saver or power down while mouse remains where it was when command was executed. Re-enables saver as soon as mouse is moved (note that enabling is different from starting the saver). Especially useful as a screen corner command. Disables your screen saver if parameter field blank or if parameter field contains the word "saver". If parameter field includes "poweroff", disables power off

feature of Energy Star monitor. If parameter field includes field "lowpower", disables low power feature of Energy Star monitor. (Win95 only for power control)

Screen Saver Timeout	Changes the screen saver time out to the value in the parameters edit box, which is assumed to be in minutes.
Change Display Res	Changes screen <u>resolution</u>
Change Screen Saver	Changes the screen saver using the method selected in the <u>Paper/Saver dialog</u> . If "no change" is selected, the saver changed to a random .scr file in the same directory as the current screen saver file.

Built-in Commands: WallPaper, Screen Saver, and Sound

Drag and Drop Rename	Used to <u>rename</u> a file.
Change Paper	Changes the wallpaper using the method selected in the <u>Paper/Saver dialog</u> . If "no change" is selected, the wallpaper is changed to a random .bmp file in the same directory as the current wallpaper file.
Change Screen Saver	Changes the screen saver using the method selected in the <u>Paper/Saver dialog</u> . If "no change" is selected, the saver changed to a random .scr file in the same directory as the current screen saver file.
Change Sound	Changes the sound files associated with all sounds in the <u>Sound dialog</u> that are not set to the "no change" sound method. Note that this is different from the Change Wallpaper command: the Change Wallpaper command always changes the wallpaper but the Change Sound command only changes sounds where the method for changing is not set to "no change".
Play Sound	Plays a wav file drag/dropped onto the button. Stops any playing sound if no file is drag/dropped. You can also associate Stiletto with the WAV extension using Explorer/File Manager. Double clicking on a WAV file in Explorer/File Manager will then play that sound file.
Show Wallpaper	Shows wallpaper drag/dropped onto the button. Does not save it (use <u>Wallpaper</u> dialog for this). If no file dropped, shows wallpaper saved in configuration file. You can also associate Stiletto with the BMP extension using Explorer/File Manager. Double clicking on a BMP file in Explorer/File Manager will then preview that file as wallpaper.
Save/Restore Desk Icon Pos	Save/restore icon <u>positions</u> .

Alphabetic List of Built-in Commands

Add Reminder Msg	Add a reminder message <u>alarm</u> .
Active windows list	Inserts a <u>list f active windws</u>
All Windows to Menu	<u>Fills</u> menu with active windows
Auto Minimize Mode	Turns off or on <u>autminimize</u> .
Bar Size	Changes the <u>number f buttns displayed</u> .
Bring Stiletto to Top	Brings every Stiletto button bar to top,
Browse and Run	Displays a file selection dialog; runs selection
Built-In Cmds List	Displays a list box from which you can execute
Change Sound	Changes the sound files associations
Change Screen Saver	Changes the screen saver
Change Wallpaper	Changes the wallpaper
Change Display Res	Changes screen <u>reslutin</u>
Clear Timer	Zeros a <u>timer</u>
Clear Recent Docs	For Win95/NT4, clears recent docs list
Close/... Window Under Mouse	Use *Control Window instead.
Close Windows List	Inserts a <u>list f active windws</u> into the menu.
Configure Stiletto	Activates the Stiletto Cnfiguring Stilett
Confirmed Win Exit	After asking you to confirm, exits Windows to DOS (you
Confirmed Win Restart	After asking you to confirm, exits Windows to DOS
Control Window	Close, min, etc, any window.
Confirmed System Boot	After asking you to confirm, exits Windows and re-boots
Disable Screen Saver	Disables your screen saver.
Dos Command	Executes the <u>ds cmmands</u>
Drag and Drop Rename	Used to <u>rename</u> a file.
Enable Screen Saver	Enables your screen saver
End Submenu	Ends a <u>submenu</u>
End Context Menu	Ends a <u>cntext menu</u>
Exit Stiletto	Terminates Stiletto .
Folder Contents Menu	Display directory contents as menu
Hide Stiletto	Hides every Stiletto button bar, making them
Hide Desk Icon	Hides Desk Icons/Prog Mgr
Hide Window	Used to select / <u>hide a windw</u>
Menu	Displays the Stiletto menu
Menu Separator	Separates displayed menu entries
Message Box	Displays a message box containing
Menu of Win Under Mouse	Copies menu bar of window (win95 only).
Minimize Window List	<u>List f active windws</u> in menu
Move Stiletto	Makes the Stiletto button bar <u>mve</u>
New Menu Column	Starts a new column in the menu;
Play Sound	Plays a wav file drag/dropped onto the button
Position Stiletto	Drag Stiletto to an arbitrary position
Process All	Minimizes/closes all top-level windows
Prog Mgr/Shell Grps	Execute a <u>prgram manager cmd</u>
Quick Win Exit	Exits Windows
Quick Win Restart	Exits Windows to DOS mode and then restarts Windows
Quick System Boot	Exits Windows and re-boots DOS
Random File	Runs a command with a randomly selected file
Recently Run Cmds List	Inserts a list of the 10 most recently run cmds

Reset All Stiletto	Restarts all Stiletto bars.
Restart Stiletto	Restarts Stiletto based on a new ini file.
Recently Run Cmds List	Inserts a list of the 10 most recently run cmds
Save/Restore Desktop Icon Pos	Save/restore icon <u>psitins</u> .
Send Keys	Used for <u>sending keys</u> .
Show Desk Icon	Shows Desk Icon/Prog Mgr
Set Timer	Sets a <u>timer</u> .
Show/Move Stiletto	Show and optionally move bar.
Show System Resources	Shows system <u>resurces</u>
Show Wallpaper	Shows wallpaper drag/dropped onto the button
Start Screen Saver	Starts your screen saver.
Start Submenu	Starts a <u>submenu</u> ;
Start Context Menu	Starts a context <u>menu</u>
Start Timer	Starts a <u>timer</u> .
Stiletto Exec	For <u>clse/min</u> , <u>filing</u> , <u>clipboard</u> , <u>psitining</u>
Stop Timer	Stops a <u>timer</u>
Temp Disable Saver	Disables your screen saver temporarily.
Tile Windows	Re-arrange active <u>windws</u> .
Timer Id	Sets the <u>timer id</u>
Tiny Run Box	Displays a small dialog box for command entry
Toggle Timer	Stops a <u>timer</u> if it is running;
Type and Run	Displays a dialog with an combo box in which
View/Change Timer	View/Change the <u>timer</u> information.
Virtual Desktop	Create or switch-to virtual <u>desktp</u> .
Window On Top List	Inserts <u>list f active windws</u> into the menu
Window Not On Top List	Inserts <u>active windws</u> into menu.
Windows Shutdown	Activates standard Windows shutdown dialog

Manipulating Windows of Running Programs

Purpose

Use the *Control Window command to ask **Stiletto** to close, minimize, tray minimize, rollup to caption and perform many other actions with the windows on your system. You can specify the windows to be controlled by selecting the active window, the window under the mouse, a window from a menu of active windows that **Stiletto** shows, a list of window captions, or all windows on your system.

Configuration

The command has this format:

```
Command      *Control Window
Parameter    action windowID
```

The action specifies what to do; the windowID species which windows to perform the action on. See following sections for more explanation of these fields.

Examples

```
Command      *Control Window
Parameter    min active
minimizes the active window.
```

```
Command      *Control Window
Parameter    rollup menu
displays a menu of active windows; the selected one is rolled up to the caption.
```

```
Command      *Control Window
Parameter    show menu hidden
displays a menu of active windows including hidden windows; the selected one is shown and activated.
```

```
Command      *Control Window
Parameter    close all
closes all windows on your desktop.
```

```
Command      *Control Window
Parameter    minmemory "*Netscape,*Internet Explorer"
swaps Netscape or Internet Explorer out to disk (NT only).
```

Specifying Action for Control Window Command

Following are the possible values for the action of the *Control Window command:

close	closes window
closeforce	forces the window to close; you may lose unsaved information
min	minimizes the window
max	maximizes the window
normal	displays as non-minimized, non-maximized
move	move the window
size	size the window
hide	makes window invisible
ontop	displays always on top (ontop is one word)
nottop	removes always on top setting (nottop is one word)
show	activates the window and shows it if hidden

back	sends window to bottom of stack of displayed windows
backshow	sends window to back if it is foremost; activates if it is not
center	centers within full screen
rollup	rolls up the window to just caption; shows if it is already rolled-up
toggle	maximizes normal window; makes maximized window normal
traymin	minimizes window to tray
automin	minimizes window to tray if window matches autotraymin on Window Control tab;
ordinary minimize	otherwise
minmemory	minimize memory by setting working set to zero (NT only)

Specifying the WindowID for the Control Window Command

Select one of the following options for the WindowID of the *Control Window command:

active	Selects the active window.
*	Selects the active window.
activebar	Window corresponding to last active bar button pushed.
autorun	last window matched by autorun menu
under	Selects the window under the mouse. For applications which use the Multiple Document Interface, the commands close, min, max, rollup will operate on the MDI child only; put Parent after under to avoid this and ensure the command always runs on the parent window.
menu	Displays a menu of active windows; select one for the action. Put hidden after menu to include hidden windows. If the *Control Window menu command is included in a Stiletto menu, the generated menu will be embedded in the outer menu. To avoid this, put noembd in lower case in the work directory of the *Control Window command.
all	Selects all visible windows, including minimized windows.
window_list	Selects the windows specified in the list. Enter one or more window captions, separated by commas. Enter xxx* for captions starting with xxx, *yyy for captions ending in yyy and *zzz* for captions containing zzz. Or you can enter =exename to select all windows shown by the program with file name exename (you must only enter the file name: not the path and not the extension). Put ~ at the start of the window list to avoid an error message if no matching window is found. Put the window_list in double quotation marks if it contains blanks. Example: “*Notepad,*Internet Explorer, =calc” selects notepad windows, Internet Explorer windows, and Calculator windows.

Switching to another Active Window

Stiletto provides the capability to switch among active top-level windows using either a menu or a button bar.

To use the menu, you need to put the built-in command “Active Window List” directly into the menu or as part of a submenu of the menu. When the menu is activated, a list of currently active windows is displayed. Selecting one of the windows causes that window to be activated. It will be opened and put on top of the desktop if it is minimized or hidden beneath other windows.

Use the Menu/ Setup dialog to control whether or not hidden windows appear on the active window list, what prefix text to put in front of each entry of the menu to remind you that the selected entry will be activated.

You can ask **Stiletto** to automatically minimize non-active windows

Closing an Active Window

Stiletto provides the capability to close any active window using either a menu or a button bar.

To use the menu, you need to put the built-in command "Close Window List" directly into a main menu or as part of a sub-menu of a menu. When a menu is activated, a list of currently active windows is displayed. Selecting one of the windows causes that window to be closed. You will first be asked if you want to save any unsaved information.

Use the Menu Setup dialog to control whether or not Hidden windows appear on the close window active window list and to specify prefix text to put in front of each entry of the menu to remind you that the selected entry will be closed.

Stiletto will normally use a polite close which a program which has unsaved data may refuse or which a program which is not responding may ignore. You can try to force such a program closed by holding both the Shift and Control key down when you select the window from the "Close Window List".

You can also close an active task with an alarm or a **Stiletto** command or close a **Stiletto** bar from a command

Making a Window Not Always On Top

Stiletto provides the capability to stop a window staying always on top of the other windows on your desktop.

To use this capability, you need to put the built-in command "Window Not On Top List" directly into the main menu or as part of a sub-menu of a menu. When a menu is activated, a list of currently active windows is displayed. Selecting one of the windows causes the window to be no longer be fixed on top of other task windows.

Use the Menu Setup dialog to control whether or not Hidden windows appear on the put on top active window list and to specify prefix text to put in front of each entry of the menu to remind you that you that the selected entry will no longer be on top.

You can put windows on top with Windows on Top.

Making a Window "Always On Top"

Stiletto provides the capability to make any window stay always on top of the other windows on your desktop.

To use this capability, you need to put the built-in command "Window On Top List" directly into the main menu or as part of a sub-menu of a menu. When a menu is activated, a list of currently active windows is displayed. Selecting one of the windows causes the window to be fixed on top of other task windows.

Use the Menu Setup dialog to control whether or not Hidden windows appear on the put on top active window list and to specify prefix text to put in front of each entry of the menu to remind you that the selected entry will be put on top.

You can reverse the on top status with Windows Not On Top.

Minimizing a Window

Stiletto provides the capability to minimize an active window.

To use this capability, you need to put the built-in command "Minimize Window List" in a menu or as part of a sub-menu of a menu. When a menu is activated, a list of currently active windows is displayed. Selecting one of the windows causes the window to be minimized.

See also using a command to minimize.

Active Window Switching with Buttons

You can create **Stiletto** buttons which automatically track each top-level window on your system so you can quickly switch to a new active window by left clicking a button and close any visible window by middle-clicking (shift-left) the corresponding button. This is an alternative to the [list of active windows](#) menu item which can also be used for active window switching. The advantages of buttons are that all active windows are immediately visible on the button bar and that switching is done with a single click. The disadvantage of the button bar is that extra desktop space is used.

You set up active bar buttons with the [Buttons](#) dialog. Check the Active Task Buttons check box and then select options as follows:

Select the start at button: buttons starting from this number up to the number of displayed buttons as set in the [Buttons](#) dialog will track active tasks. You can set this to 1 if you want the entire active bar to track active tasks (see also [multiple button bars](#)).

If you want icons to appear on the active bar, select icon small, icon large, or icon stretch from the drop down box (see [buttons](#) for an explanation of these terms).

If you want to specify the icon for the button for a window, set the icon menu drop down; see below for details.

If you want icons and text, select an icon type from the drop down box and check the text check box. The percentage of the button face used for the icon is taken from the percentage set in the start at button. Gray check the text box to avoid automatic update of the bar whenever the caption text changes.

Use the checkbox to specify that only one window per active task be represented on the bar.

You can specify whether hidden windows should be displayed as buttons on the bar.

You can specify whether the right or middle mouse click should be used to close tasks. Gray check to specify that neither should be preset to close.

You can specify that icon size, color, button size, and font information from the first active bar button (the start at button) is automatically copied to all active bar buttons each time you reconfigure.

You can specify that the bar will shrink and grow so that only buttons with corresponding active tasks are visible or you can specify that the bar should stay fully visible where buttons without corresponding active tasks are blank.

You can specify that **Stiletto** should use the icon from the window class for the active bar button (otherwise, **Stiletto** will use the first icon in the exe file). The Windows task bar also uses the window class icon.

(For NT 3.51, you can also specify that minimized (iconized) non-Dos windows be hidden).

Gray check the active bar check box to show the button corresponding to the foreground window as being pressed down.

Once you have set the above options, **Stiletto** will display a button for each top-level window of each active task, starting at the start at button, and up to the maximum number of buttons set by the number of buttons displayed in the [Buttons](#) dialog.

[Flyover](#) help for the left button of the Active Bar displays the window title of the corresponding active window. You may want to set a small delay for flyover help for the Active Bar using the [bar](#) dialog so this information appears as soon as you move the mouse cursor over the Active Bar.

You can use the [omit list](#) to cause any active window to be excluded from the Active Bar or to edit the name of text for the Active Bar. Or, you can use the [Hide Window built-in](#) command to hide the window.

When displaying the icon for a window on a button, **Stiletto** normally uses the first icon in the exe or the window class icon. You can specify your own icons for programs by creating a special menu and selecting this menu with the icon menu drop down. Create one entry in the menu for each program with an icon that you want to specify. Set the menu item name to =exename, where exename is the name of the program exe file without the path and without the .exe extension (eg =winword for Microsoft Word).

Leave the menu item command set to (none). Set the menu item icon to the icon you wish to use for all windows from exename.

You can cause activated task windows to be centered using a switch on the Special GUI dialog.

Stiletto normally uses the icon in the exe file for the Active Bar. Some programs create a new icon when they run. If you wish to use this icon, there is an internal option to do this. **Stiletto** normally does not show win95/NT4 toolbar-style windows on the active bar; if you would like to see them, use the internal ShowToolbar option.

Omitting Windows and Words from Active Window Lists

You can use the omit list edit box on the Window Control dialog to omit either words from a window name or to omit windows completely from the list of active windows or the Active Task Buttons.

To omit a word, type the word followed by a comma. For example, you could use this technique to delete vendor names.

To omit an entire window, type the window name as it appears in the caption title of the window followed by a comma, e.g. Program Manager in the omit list will mean that no entry for Program Manager will appear.

You can also delete any window associated with the program filename.exe by including =filename in the omit list (no .exe, no path).

If you include a string followed by an asterisk (*) and comma in the omit list, then any active window with caption text starting with that string will be deleted. For example, 1MBFort* will delete any program name starting with 1MBFort.

Tiling Active Windows

You can tile active, top-level windows with the Tile Windows built-in command. It re-arranges active windows as follows:

If the Parameters edit box starts with the letter **c** or the Control Key is held down, windows are arranged in equal-sized columns.

If the Parameters edit box starts with the letter **t** or the Shift Key is held down, windows are arranged in equal-sized rectangles.

Otherwise, windows are arranged in equal-sized rows.

You can restrict the affected windows to only those containing "text" in their window captions by putting *text in the Parameters edit box. For example, *explor means that only Explorer windows are affected.

Use *tile for multiple commands.

To start two instances of Explorer and tile them use the following:

```
Command: Explorer.exe  
Parameters: <explorer.exe<*wait 1<*tile *explor
```

You also need to ensure the "Switch To If Active" box is unchecked.

Sending Key Strokes to Other Programs

There are two ways to send keystrokes to other programs.

Command *Keys
 Parameter "keystrokes"
 sends keys to the active window. The more advanced *Send Keys command allows you to specify the window to receive the keys:

Command *Send Keys:
 Parameter window-id "keystrokes"

See the following sections for details.

Your mouse and keyboard will "freeze" while keys are being sent; this is normal operation.

To set up a global macro key to play a commonly used sequence of keys, assign the Keys command to a hot key. If you have many such keys you may prefer to create a menu of these keys and attach the menu to a hot key or mouse action.

You can use this technique to set up program specific hot keys or menus of keys. To send mouse clicks, use Stiletto Exec mouse. You can control the delay between sent keys with the SendKeyDelay internal options. You cannot send keys to Dos windows.

Stiletto will wait for Ctrl, Alt, and Shift to be up before sending keys; you can control the maximum length of the wait with the HotKeyAllUpWait internal option

Specifying the Window to Receive the Keys

The parameters of the *Send Key command start with the window id to receive the keys. You will usually use an asterisk to specify the currently active window (excluding **Stiletto**; it remembers the window which was active before it). But there are other possibilities:

*	sends keys to current active window
=File Path	sends keys to program run from that "File Path"
Title	sends keys to window with caption "Title"
PartTitle*	sends keys to window with caption starting with "Part Title" (Note asterisk at end)
*PartTitle	sends keys to window with caption end with "Part Title" (Note asterisks at start)
**	sends keys to window of last command started by Stiletto
autorun	sends keys to window last matched by autorun command
activebar	sends keys to last window referenced by active bar button

If the window file or caption that you want contains spaces, enclose it in double quotation marks.

Stiletto shows and activates the main window of the program to receive the keys. This window will normally set the keyboard focus to the one of its child windows which should receive the keys. Sometimes, the wrong child window is chosen. For example, if you select an item from a listbox and it is not highlighted when keys are sent, this could indicate this problem. To try to solve it, precede the window id by an at sign, eg @* for the active window. **Stiletto** will then set the keyboard focus to the child window under the mouse cursor, if this child window belongs to the target program.

Specifying the Keys to be Sent using Send Keys

After the window id, the parameters of the *Send Key and *Keys commands specify the keys to be sent.

You can use the recorder in the wizard to capture the keys or you can type them directly as follows:

Send letters, numbers, and other keyboard characters by typing them as you want them to be sent. If you are sending spaces, enclose the keys to be sent in double quotes.

To specify an Alt-prefixed key, prefix it by %; similarly use ^ for Ctrl key, + for Shift, and combine as needed (eg %^ for both Ctrl and Alt). Alternatively, you can use {alt} to toggle Alt up/down which allows multiple keys to be sent with Alt down: eg {alt}ab{alt} sends Alt-Down, a, b, Alt-Up. Similar for {ctrl} and {shift}.

Use {datelong}, {dateshort}, {time} for sending the current date and time. To send the time without seconds, use {time}{back}{back}{back}.

You can change the either the { or the } or both to any non-alphanumeric using the Advanced dialog. They can be set to the same character.

Use the following character pairs enclosed in {} for special characters. You have a choice between the long form and a two-letter abbreviation:

{cmdsep}, {cs}	Insert the command separator character (default <).
{param}, {pp}	Insert the parameter prompt character (default ?).
{clip}, {cc}	Insert the clipboard character.
{var}, {sv}	Insert the script variable character.
{plus}, {pl}	Plus (also can use {+})
{percent}, {pe}	Percent sign (also can use {%})
{caret}, {ca}	Caret (also can use {^})
{}	Inserts the brace. Change the {} to whatever you have changed { to on the Advanced dialog.
{brace}, {br}	Curly Brace (
{enter}, {en}	Enter
{space}, {sp}	Space
{quote}, {qu}	double quote
{question}, {qn}	question mark
{greater}, {gt}	greater than sign >
{less}, {lt}	less than sign <
{tab}, {ta}	Tab
{esc}, {es}	Escape
{up}, {au}	Up arrow
{down}, {ad}	Down arrow
{left}, {al}	Left Arrow
{right}, {ar}	Right Arrow
{ins}, {in}	Insert Key
{del}, {de}	Delete Key
{back}, {ba}	Backspace Key
{home}, {ho}	Home Key
{end}, {ed}	End Key
{pgup}, {pu}	Page Up
{pgdn}, {pd}	Page Down
{pad+}, {p+}	Numeric Pad +
{pad-}, {p-}	Numeric Pad -

<code>{pad*},{p*}</code>	Numeric Pad *
<code>{pad/},{p/}</code>	Numeric Pad /
<code>{dateshort},{ds}</code>	date in Windows short format
<code>{datelong},{dl}</code>	date in Windows long format
<code>{time},{ti}</code>	time in Windows format
<code>{fn}</code>	Function Key "n" (eg <code>{f1}</code> for function key 1; do not actually use the letter n)
<code>{wn}</code>	Wait n tenths of a second (eg <code>{w1}</code> to wait one tenth of a second).
<code>{nnn}</code>	Send character with decimal ascii code nnn (first n cannot be 0).
<code>{alt},{at}</code>	Toggle Alt down/up; use <code>{at}ab{at}</code> to send Alt-down, a, b, Alt-up.
<code>{shift},{sh}</code>	Toggle Shift down/up; use <code>{sh}def{sh}</code> to send DEF.
<code>{ctrl},{co}</code>	Toggle Ctrl down/up; use <code>{co}{ta}{ta}{co}</code> to send Ctrl-Down, tab, tab, Ctrl-up.
<code>{win},{wi}</code>	Toggle Win Apps down/up; use <code>{wi}p{al}</code> to open accessories.

Example: `"%fnhello^v%{f4}"` sends Alt-F, then n, then hello, then ctrl-v, then alt-f4.

If you have only one key to send, the surrounding double quotes are not needed. You can send at most 200 keys.

You can send Alt+0xxx keys (eg `alt+0181 =µ`) but on some systems these may not work as the first character sent. Put `{sp}{ba}` in front to work around this problem (space, backspace).

If you have only one key to send, the surrounding double quotes are not needed. You can send at most 200 keys.

Specifying the Keys to be Sent using Send Keys (Blank Separator Approach)

Starting with version 97f, the default way to specify keys to be sent uses no blanks between keys and uses `{}`'s to surround special characters. However, the older, blank-separator approach is still allowed; it is specified by including `compresskey=0` under `[general]` in `stiletto.ini`, or by omitting `compresskey` entirely from the ini file. Following is a description of this approach.

After the window id, the parameters of the `*Send Key` command specify the keys to be sent.

Send letters, numbers, and other keyboard characters by typing them separated by spaces and enclosed in double quotes.

To specify an Alt-prefixed key, prefix it by `a-`; similarly use `c-` for Ctrl key, `s-` for Shift, and combine as needed (eg `a-c-` for both Ctrl and Alt).

Use the following character pairs for special characters:

<code>en</code>	Enter
<code>sp</code>	Space
<code>qu</code>	double quote
<code>qn</code>	question mark
<code>gt</code>	greater than sign >
<code>lt</code>	less than sign <
<code>ta</code>	Tab
<code>es</code>	Escape
<code>au</code>	Up arrow
<code>ad</code>	Down arrow
<code>al</code>	Left Arrow
<code>ar</code>	Right Arrow

in	Insert Key
de	Delete Key
ba	Backspace Key
ho	Home Key
at	Alt Key
co	Ctrl Key
ed	End Key
pu	Page Up
pd	Page Down
p+	Numeric Pad +
p-	Numeric Pad -
p*	Numeric Pad *
p/	Numeric Pad /
ds	date in Windows short format
dl	date in Windows long format
ti	time in Windows format
fn	Function Key "n" (eg f1 for function key 1; do not actually use the letter n)

Example: "a-f n h e l l o f1" sends Alt-F, then "nhello", then function key 1.

If you have only one key to send, the surrounding double quotes are not needed. You can send at most 200 keys.

You can send Alt+0xxx keys (eg alt+0181 =µ) but on some systems these may not work as the first character sent. Put sp ba in front to work around this problem (space, backspace).

Examples of Send Key Commands

Command	*Send Keys
Parameter	* "c-ed"

Sends Ctrl+End to the active window. This key combination often tells the program to go to the end of the information being displayed.

Command: *Send Keys
 Parameter c:\yourpath\prog.exe "c-ho a b c"

Sends Ctrl-Home followed by **abc** to window started from c:\yourpath\prog.exe.

Command: *Send Keys
 Parameter: **Notepad "a-f o"

Sends Alt-f followed by **o** to the window with caption ending in **Notepad**. This would select the open command from the menu.

Command: *Send Keys
 Parameter: +**Down "a-v l"

Waits for up to 5 seconds for the window with the appropriate caption to appear, then sends Alt-v followed by **l** to the window with caption starting with **Down**. This could set the list view for Explorer.

Sending Keys to Programs When They Are Started

Since Windows is a multitasking system, starting programs and sending them keys requires care. You must make sure the program you are starting is ready to receive them.

To start a program and send it keys at start up, use multiple commands. For example, to start c:\ql\myprog and send alt-g n, specify

Command c:\ql\myprog.exe
 Parameters: <*wait ready<*send * "%gn"

Note how * is used to refer to the active window, which will be the command just started. The sequence <*wait ready causes **Stiletto** to wait until the program is ready to accept input before sending the keys.

You can use the *Keys command instead of *Send *.

If the *wait ready does not work for some reason, try *wait 2 (or some other digit) to wait 2 seconds.

You can also wait for up to 5 seconds until a window with a specified caption appears by preceding the caption with a +

Command: *Explorer
 Parameter: <*send +*Exploring* "%v l"

Start Explorer, then waits for up to 3 seconds for the window with a caption containing **Exploring** to appear, then sends Alt-v followed by **l** to the window with caption starting with **Exploring**. This could set the list view for Explorer. This is especially useful with Explorer, where the ! may not work (since Explorer is always running). You must use the *Send Keys command for this approach.

Send Mouse Clicks to the Active Window

You can send single mouse clicks to the active window with a command of this form

```
Command    *Stiletto Exec
Parameter  mouse xxx
```

where xxx is left, middle, or right.

You can send double mouse clicks to the active window with a command of this form

```
Command    *Stiletto Exec
Parameter  mouse xxx double
```

where xxx is left, middle, or right.

For example

```
Command    *Stiletto Exec
Parameter  mouse right
```

will bring up the right (context) menu; this is useful, for example, in a **Stiletto** window (context) menu which you have attached to right-clicking.

For example

```
Command    *Stiletto Exec
Parameter  mouse left double
```

sends a left double click to the underlying window. Attach it to a mouse middle click hot key if you want to simulate left double click by middle clicking.

For single mouse clicks, you can specify the screen coordinates to which the mouse click is sent as two numbers at the end of the Parameters.

```
Command    *Stiletto Exec
Parameter  mouse stiletto
```

Activating the command positions the mouse cursor on the bar and readies the bar to receive keystrokes. You can put a number n after stiletto to move to the nth button.

Sending Mouse Clicks and Moves to the Active Window

You can also use the *mouse command to send a sequence of mouse clicks, mouse positions, and mouse moves to the active window. The parameters field of this command contains a series of two letter commands which indicate the mouse actions to perform. The commands which move or position the mouse are followed by two numbers giving the move or position value in pixels. Here are the two letter commands:

```
lc          left click (both left down and left up)
ld          left down
lu          left up
ll          double click left (note: lc lc will not work)
mc          middle click (both middle down and middle up)
md          middle down
```

mu middle up
 mm double click middle (note: mc mc will not work)
 rc right click (both right down and right up)
 rd right down
 ru right up
 rr double click right (note: rc rc will not work)

mo x y move mouse x pixels right, y down (x or y can be negative)
 ab x y set mouse to absolute position x y (absolute means 0 0 is top left of screen)
 re x y set mouse to relative position x y (relative means 0 0 is top left of active window)

Examples

Command *mouse
 Parameter ll
 sends double left click.

Command *mouse
 Parameter re 20 50 rc
 position mouse at 20 50 with respect to active window then sends right click

Command *mouse
 Parameter ld mo 30 -40 lu
 sends left down, move 30 right, 40 up, left up (eg will draw a line in MS Paint).

Command *mouse
 Parameter ab 40 60 lc<*send * abc
 moves mouse to absolute position 40 60, sends a left click, then sends letters abc

Sending URLs to a Running Web Browser

You can send a URL to a running browser (either Netscape or Internet Explorer) with a command line this:

Command *Stiletto Exec
 Parameter url http://www.inforamp.net/~crs2086/index.htm
 which sends the url of the **Stiletto** home page. The keyword **url** is followed by the url itself.

Create a menu of such commands or attach the command to hot keys to provide an alternative approach to browser bookmarks.

You can also type URLs into the tiny command line.

Running a Command with a Randomly Selected File

Purpose

The *Random File command selects a file at random using a file path with wild cards that you provide. The selected file is then copied to a specified target file; or is used to change the wallpaper (desktop background), screen saver, or sound event; or is used in a command line to execute a program or batch file that you specify; or it is run.

One use of this command is to set up your own randomization routines for system files. For example, you can randomize the Windows 95 shutdown screens by creating .bmp files with the appropriate size and color depth, putting the files into a folder, and using the command to copy a randomly selected file over c:\windows\logow.sys. Take a backup copy of logow.sys before experimenting with this.

To implement this type of randomization, you can put the *Random File command in your startup menu or as a scheduled command.

Configuration

The parameters edit box of the *Random File command specifies the action to be taken. In general, it takes this form:

Parameters: action filepath args

where action is one of

copy to copy a random file to an output file file

run to run the file

saver to change the screen saver

paper to change the wallpaper

"sound event" to change a sound event

else any other string is a file path specifying a command to be run or a **Stiletto** built-in command like "*Play Sound".

The filepath must contain wildcards, such as *.bmp or *.*. A randomly selected file is selected from the directly which matches the path. You can also randomly select a file folder and then a file by using a single exclamation mark as the last folder name in the path:

c:\parent\!* .jpg

selects a random folder under c:\parent, then selects a random jpg file from that folder.

If any of commandpath, filepath, or outfile contain blanks, the path must be placed in double quotation marks.

Further Information

Copy: Parameters: copy filepath outfile

selects a file at random from the filepath (which must contain wildcards like *.bmp) and copies it over outfile.

Paper: Parameters: paper filepath

selects a file at random from the filepath and sets the wallpaper to this file.

Sound: Parameters: sound "Application" "Event" filepath

selects the sound for specified application and event; put them in quotes if they contain blanks. Use Control Panel | Sound to see the list of applications and events.

Screen Saver: Parameters: saver filepath

selects a file at random from the filepath and sets the screen saver to this file.

Run: Parameters: commandpath filepath args

selects a file at random file filepath (which must contain wildcards like *.*), then runs the command given by commandpath using a command line consisting of the commandname, the selected file, and finally the args. If you want the command to be run invisibly, put ***hide** after the args at the end of the parameters edit box. If the command being run is a .bat file, you may want to use explorer to set its properties to include **close on exit**. This is especially important for commands run invisibly.

Examples:

Command *Random File

Parameters: copy "c:\my logo files*.bmp" c:\windows\logow.sys
copies a random bmp file from c:\my logo files over the logow.sys file.

Command *Random File

Parameters: paper "c:\paper files\!* .jpg"
sets wallpaper to a random .jpg file from a random directly under c:\paper files.

Command *Random File

Parameters: sound Windows "Close Program" "c:\zounds*.wav"
sets the sound for Window Close Program to a random wav file from c:\zounds.

Command *Random File
 Parameters: "*Play Sound" "c:\zounds*.wav"
 plays a random wav file from c:\zounds.

Command *Random File
 Parameters: "c:\program files\bat\exec.bat" "c:\random*.*" arg2 arg3 *hide
 selects a random file from c:\random, then executes the exec.bat file with the selected file as the first argument, then arguments arg2 and arg3. The command is run in an invisible window.

KeyBoard Macros

Purpose

Keyboard macros let you replace one set of typed characters by others. You can also use keyboard macros to run Windows programs or to execute **Stiletto** Windows configuration features or builtin commands.

For example, you could define **.me** to be replaced with **Your Name**. Or you could define **Alt-tm** to minimize the current window.

Configuration

To define a set of keyboard macros, you need to do two things: define the macros and define the macro signal character.

You define the macros and the corresponding actions by creating a menu. Put each macro as the menu item name and the macro command as the corresponding menu command. Use only letters, digits, and spaces in the menu item name. Use the ***Keys** command to send keystrokes if you want to define a macro abbreviation for the corresponding keystrokes.

After defining the macros, you need to define a hot key character which is used to signal that a macro may follow. You do this by defining any hot key and assigning it the ***Macro** command. Put the name of the menu with the macros in the parameter edit box of the command.

For example, suppose you define a menu **mymacros** with these four entries:

Menu Item Name	me
Menu Item Command	*Send Keys
Menu Item Parameters	* yourname@yourdomain.com

Menu Item Name	new
Menu Item Command	*Send Keys
Menu Item Parameters	* %fn

Menu Item Name	sq
Menu Item Command	*Send Keys
Menu Item Parameters	* {sp}{ba} ²

Menu Item Name	xp
Menu Item Command	c:\windows\explorer.exe
Menu Item Parameters	

Also suppose that the period is defined as a hot key as follows
 Hot Key .

Hot Key Command	*Macro
Hot Key Parameters	mymacros

When you type **.me**, **Stiletto** would replace the **.me** by **yourname@yourdomain.com**. Similarly, **.new** would be replaced by **Alt-fn**, and **.sq** would be replaced by the superscript 2 (²). Finally, typing **.xp** would cause Windows Explorer to be started.

If you type period followed by any other sequence of characters, nothing will happen – the typed characters will not be changed.

Further Information

Be careful when you define macros: **Stiletto** will execute the shortest macro that applies. For example, if you define one macro **ab** and another one called **abc**, then the **abc** macro would never be executed since the **ab** macro would also be matched first. To help avoid this, you can put spaces in macros, including spaces at the end. The space then has to be typed for the macro to be executed.

You can have as many combinations of macro signal characters and menu tables for macros as you want.

You can use program-specific hot keys to limit macro expansion to certain windows or to avoid checking for a macro with certain windows.

The ***Macro** command can only be used with hot keys. You will get an error message if you use it in any other context (eg as a button command).

Timers

Stiletto has 26 timers that you can control and optionally display as button labels. The timers are identified by the single-letter labels a, b, c, ..., z.

Use the button on the time setup dialog or execute the Reset Timer built-in command to access timers from a dialog. Using this dialog, you can start or stop a timer, change its value, and associate commands with the timer starting, stopping, and resetting. The timer reset value is also set on the dialog.

You can also start, stop, toggle, and clear any of the timers with the built-in commands of the same name. Put the letter of the timer to be accessed in the command parameters box. You can access multiple timers by listing all the timers ids, **not** separated by blanks. For example,

```
Command:   Start Timer
Parameters: bcg
```

starts timers b, c, and g.

You can also use the Set Timer command to start, stop, or toggle timers and to set their value.

Using the button dialog, you can have **Stiletto** place a timer as the label on any button. Use the timer id field in this dialog to indicate which timer is to be displayed. You can also specify that the timer label is to be shown with the timer value.

A running timer is displayed in the form **hhhh.mm** (hours, then a period, then minutes).

A stopped timer is displayed in the form **hhhhxmm**.

Timers are displayed rounded to the nearest minute, unless you specify that seconds should be shown in the Time Setup dialog. In this case, seconds are always shown as **.ss** at the end of the timer label.

Timers are updated at the same interval as the resource usage check set in the Time Setup dialog. But, unless you specify that seconds should be shown, the button display is only changed once per minute. You can always see the timer value to the nearest second with the Reset Timer Command.

You can have **Stiletto** play an alarm sound at regular intervals for all timers using Time Setup dialog. The alarm sounds two minutes before the interval to give you time to react (eg alarms every 30 minutes will sound at 28 minutes, 58 minutes, and so on).

You can start and stop timers from external programs, such as dialer scripts. You can also use the external timer commands to control timers using a command launched from any button.

To automatically clear a saved timers once per day, set up an alarm with these characteristics (using timers c and g for example):

```
Time:           12:01 AM
After alarm rings: Alarm again in 1 day
Command:        Clear Timer
Parameters      cg
```

The "Ring Missed Alarms" checkbox on the Time Setup dialog must also be checked for this to work (unless you start **Stiletto** each day at 12:01!). You can use a similar technique to clear timers once per month (ring on first of month at 12:01)

You can start and stop all timers on a button bar based on the status of your modem.

You can ask **Stiletto** to log all timer events in a file.

Starting with version 1.9s, timers are normally updated by computing the difference between the current clock and the clock setting with the timer was last calculated. This method keeps timers accurate even if another application monopolizes the CPU for an extended period. However, it does mean that timers

must be manually reset if you change the system time. An alternative approach which is insensitive to clock changes is to update the timer based on assuming that timer notifications will not be lost due to other applications. You can ask for this approach with the TickTimer [internaloption](#).

Controlling Timers Externally

It is possible to start, stop, toggle, and clear [timers](#) from external programs, such as batch programs or dialers, if these programs can execute windows commands. For example, the Trumpet win sockets program script language can do this with the exec command, which looks like this: `exec "c:\exepath\exename.exe parameters"`.

If your dialer does not support external commands, you can run both the **Stiletto** timer command and your dialer as [multiple commandsto](#) start the program.

The instance of **Stiletto** with the timers to be accessed must already be running. Also, since more than one [instance](#) of **Stiletto** can be running, you need to specify the configuration file for the instance with the timer label that you want to reference.

You can also access timers by running a command like this from a batch file or dialer script:

```
c:\yourpath\stiletto.exe ini_file [start|stop|toggle|clear] timer_ids
```

where

ini_file	is the configuration file of the instance of Stiletto to be accessed
[start...]	is one of the four commands
timer_ids	is one or more timers ids, not separated by blanks.

For example

```
c:\mypath\stiletto stiletto.ini toggle fg
```

toggles timers f and g in the **Stiletto** bar started with configuration file stiletto.ini (the default).

You can also use this technique from within **Stiletto** to clear, stop, or start a timer on a button which is different from the button from which the timer command was launched: To do so, assign a command of the form given above to a button or menu within **Stiletto**.

You can play a sound whenever you execute one of these commands by placing |filepath.wav at the end of the command line, for example:

```
c:\path\stiletto.exe stilet2.ini start g |c:\win\tada.wav
```

starts the timer g and plays the sound file c:\win\tada.wav. You need to provide the entire file path, including the .wav, and you **cannot** place any blanks after the |.

You can also control timers via the status of your [modem](#).

You can also start and [close](#) an instance of **Stiletto** from your dialer if it supports command execution from scripts.

Controlling Timers by the Status of the Modem

It is possible to start, stop, and clear [timers](#) based on the status of RAS connections using your modem.

The simplest approach to doing this is to use the Setting Timers and Associate Commands dialog, accessed from the Time Setup tab, which allows you to indicate that a timer is to run when a RAS connection of the same name is active.

You can also use the *comstart, *comstop, and *comboth commands to give more precise control.

To start timers when your modem is connected, assign the following command to a button or menu and execute it:

```
*comstart b
```

where b is the timer id to be controlled; you can control multiple timers by listed their single-letter ids, **not** separated by blanks.

To stop timers when your modem is disconnected,

```
*comstop b
```

After this command is executed, **Stiletto** will wait for the modem to be connected and then stop timer b when the modem disconnects.

If you want to both start and stop timers, specify

```
*comboth b
```

By combining this with the execution of multiple commands, you can put your timers under modem control and start your modem communication program:

```
comboth ac<c:\comm\commprog.exe
```

Assuming < (the default) is the command separation character, this command puts timers a and c under control of the modem and then starts the program commprog.

You can also execute these commands from external programs such as dialer scripts., as follows:

The instance of **Stiletto** with the timers to be accessed must already be running. Also, since more than one instance of **Stiletto** can be running, you need to specify the configuration file for the instance with the timer label that you want to reference.

To control a timer externally, you execute a stiletto.exe command as if you wanted to start a new instance of **Stiletto** with a command line that has three parameters:

1. The name of the ini (configuration) file; use stiletto.ini if you are running only one instance.
2. One of comstart, comstop, comboth.
3. The timer id b.

For example:

```
c:\path\stiletto.exe stiletto.ini comstart x
```

puts the timer id x for the **Stiletto** instance with configuration file stiletto.ini under modem control. If your dialer supports script files, you can combine external timer commands with modem control commands, like this:

```
c:\path\stiletto.exe stiletto.ini comstop x
```

```
c:\path\stiletto.exe stiletto.ini start x
```

With these commands in the login script file for your dialer, timer x will be started when the script file is executed and all timers will be stopped when the modem disconnects.

Once you execute a modem control command, it will continue to stop/start timers as long as **Stiletto** is running. You can discontinue modem control at any time by replacing the button number b by the word off and issuing any of the above commands.

These commands only work Win32 RAS connections.

Setting Timers and Associated Commands

You can access the Reset Timer dialog by using the built-in command or from the time setup dialog.

You can set or clear any timer, start or stop it, associate a timer with a RAS connection, and assign a label to the timer (the label can be displayed on the button with the timer and in the timer log.)

You can specify that the timer should start automatically when **Stiletto** starts. You can specify that the timer values should be saved and restored when **Stiletto** starts and stops. You can indicate that the timer should count down.

You can specify that a timer should run only when a RAS connection is active or when a specified program is active (the foreground window).

To associate a timer with a RAS (dial-up) connection, check the "Run Timer when Dialup (gray program) Active" check box and set the timer name to the dial up name. **Stiletto** will automatically start and stop the timer according to the status of the RAS connection. You can associate more than one timer with the same connection: eg have a daily timer and a monthly timer. (To create a daily/monthly timer, add an alarm which clears the timer daily/monthly).

To associate a timer with a program, gray-check the "Run Timer when Dialup (gray program) Active" check box and set the timer name to the exe file name of the program to be timed (eg netscape for Netscape Communicator) **Stiletto** will arrange for the timer to be running only when the specified program is the foreground (active) program

You can also associate a command with starting, stopping, and resetting the timer using the command entry controls.

The reset command is used in conjunction with the Reset Hour and Reset Minute values.

For timers which count down, whenever the timer reaches zero, any associated command is executed. If the either of the Reset Hour or Minute is greater than zero, the timer is reset to that value. Otherwise, the timer is stopped.

For timers which count up, if either of the Reset Hour or Minute is greater than zero, the associated command is executed whenever the timer reaches a multiple of the Reset Hour and Minute.

You can also use the Set Timer built-in command to set a timer value and state.

Setting Timer Value and State

Use the built-in Set Timer command to set the value and state of one or more timers. The parameters edit box of the command is structured as follows:

If it starts with +, the timer is started; with - the timer is stopped, and with * the timer is toggled.

Use of one of these characters is optional: if omitted, the timer state is unchanged.

Next come the single letter timer ids of the timers to be adjusted, with no blanks.

Finally, the new timer value is indicated as three numbers: hours, minutes, seconds, separated by blanks.

Examples:

+a 0 0 0 Clear timer a and start it.

be 0 10 20 Reset timers b and e to 10 minutes, 20 seconds; leave their running/stopped state unchanged.

-c 1 0 0 Stop timer c and set its value to one hour.

Use *setimer (one t, lower case) for multiple commands.

Timer Logs

You can ask **Stiletto** to log timer events by using the Keep Timer Log check box on the Time Setup dialog.

The log file will have the same name as the configuration file used in the instance of **Stiletto**, except that the file extension will be .tlo. For example, the log file for the default configuration is stiletto.tlo. The log is always placed in the same directory as the **Stiletto** ini file.

A log file entry will be written whenever a timer starts, stops, or is re-set. As well, when **Stiletto** shuts down, a stop timer entry will be written for any running timers. When **Stiletto** starts up, a start timer entry will be written for any automatic start timers.

The logs have fixed-format records structured as follows

Column Contents

1	Always blank.
2-8	Button of last timer command.
9	Always blank.
10	Timer id (single character).
11	Always blank.
12	Action: "+" if timer started, "-" if timer stopped, "0" if reset
13	Always blank.
14-17	Year when event recorded.
18	Always blank.
19-20	Month.
21	Always blank.
22-23	Day.
24	Always blank.
25-26	Hour (military clock, ie 24 hour time)
27	Always blank.
28-29	Minute
30	Always blank.
31-32	Second
33	Always blank.
34-41	Total timer value in seconds.
42	Always blank.
43-47	Whole hours in the timer.
48	Always blank.
49-50	Whole minutes in the timer.
51	Always blank.
52-53	Seconds in the timer.

To be clear: the timer value is shown in two different formats: columns 29-36 show the timer value in seconds. Columns 38-48 show the timer value as hours, minutes, seconds.

Changing the Timer a Button is Displaying

You can change the timer a button is displaying with built-in timer Id command. The command
 command Timer Id
 parameters single letter timer label
 changes the timer associated with the button to the single character in the parameters field. If the button is displaying a timer, the display will switch to the new timer id.

The command always affects the button from which it was launched, either by button or menu.

One use for this command would be to track the time being used by two different online services and switch the button display to the timer associated with the service that you are currently signed on to.

Suppose you are using timer x for service 1, timer z for service 2, and DunDial, the command line RAS dialer that comes in the **Stiletto** zip file. You can use multiple commands to start a service and set the timer to be started and stopped along with the modem connection. (Use *timerid for the command in multiple commands). To start service one:

Command Dundial

Parameter service_1<*comboth x<*timerid x

To start service two:

Command Dundial

Parameter service_2<*comboth z<*timerid z

Assign both these commands to the button or a menu for the button on which you are displaying the timer.

Folder Contents Menu

Using the builtin Folder Contents Menu command, you can create a menu which includes the files from a directory tree and then select an entry from the menu to execute that file. In addition, for Win95/Win NT4, you can also use this command to create menus from any special folder such as a program folder like "Accessories", or the most recently accessed documents, or the shortcuts on your desktop.

If you right click to select a file, **Stiletto** displays the context menu entry for that file.

In the the parameter edit box, enter the name of the directory containing the files to be displayed or the special folders. You can list many directories or folders, separated by commas. You can use the word "Sep" to show a menu separator. You can use the word "ColSep" to start a new column in the menu.

The work directory edit box is used to hold keywords which control which files are displayed and how they are displayed; see below for more details.

Examples:

Command: Folder Contents Menu

Parameter: Desktop

to display a menu of the shortcuts on your desktop.

Command: Folder Contents Menu

Parameter: c:\work\monthly report

to display a menu of the files in c:\work\monthly report.

Command: Folder Contents Menu

Parameter: Control Panel, c:\ut\myfiles, Sep, Programs Startup

to display a menu of your Control Panel, all files in c:\ut\myfiles, programs file Start up, with menu separator after c:\ut\myfiles.

Stiletto limits the menu and submenus shown by *Folder Contents Menu to at most 1000 files.

To access folder trees with more files, use one of the following approaches.

To show an entire disk, use NoSubDir in work directory:

Command: *Folder Contents Menu

Parameter: c:\

Work nosubdir autocol 16 folderback

Shows a menu of all files/folders for top level of drive C; selecting one folder shows that folder as menu. Or, if Shift key held down when selecting from menu, shows entire folder as explorer Window. (Autocol 16 automatically starts a new column in the menu every 16 entries). Also shows a back entry in each menu to allow you to go back up the folder tree.

Another approach for large directory tree:

Command: *Folder Contents Menu

Parameter: c:\

Work explorer nofiles

Shows a menu of all folders for drive C with single explore entry in menu for each folder. Left click on this entry to show files for that folder in Explorer window. Right click to show files in *FolderContents Menu.

The first technique shows the menu faster, but requires clicks to go up or down the folder tree.

The command will try to calculate the appropriate number of entries per menu column based on screen resolution and menu font; if you are unhappy with the choice you can set it to n by putting the following line under [General] in the stiletto.ini file:

MaxMenuColumn=n

Put RightContext=0 under [General] in the stiletto.ini file to avoid showing the context menu for right clicks.

Entering Work Directory Information for Folder Contents Command

See Folder Contents [Menu](#) for an introduction. You can use the work directory edit box to control the files displayed in the menu. Use the keywords shown in bold in this table.

Columns	<p>Use autocol n to automatically start a new column every n entries; this gives the menu a toolbar look (applies to top level menu only, not submenus). Use autosoftcol n to automatically start a new column every n entries without including a bar between the columns (applies to top level menu only, not submenus).</p> <p>Use autocolall n to automatically start a new column every n entries; this gives the menu a toolbar look (applies to top level menu and submenus). Use autosoftcolall n to automatically start a new column every n entries without including a bar between the columns (applies to top level menu and submenus).</p>
Text labels	<p>Using maxtext n limits text labels to n characters.</p> <p>Using omit deletes the phrases in the "omit strings..." edit box on the special config tab; omit is applied before maxtext.</p> <p>Using mne in the menu box means Stiletto will assign single character menu mnemonics to the first 36 items on the main menu to allow them to be easily selected with the keyboard.</p> <p>Using noext in the edit box means file extensions will be removed from menu item names.</p>
Position	<p>Placing offset n1 n2 shows the menu offset n1 characters to the right and n2 characters below the mouse cursor; n1 or n2 can be negative.</p>
Sorting	<p>Use nosort in the edit box so that the items will not be sorted.</p> <p>Using sortext in the edit box means items to sort by file extension.</p> <p>Put sorttime to sort most recently change files first.</p>
Subfolders	<p>Use folderdots in the edit box means "..." is added to folder names; this is useful with NoSubDir if you do not use icons in menus.</p> <p>Use folderstart in the edit box sorts menu entries with folders at start.</p> <p>Put folderback in edit box to add Back (previous folder) entry when NoSubDir specified.</p> <p>Use nofolders to omit all subfolders.</p> <p>Use nosubmenu in the edit box means all files from subdirectories will be listed in the main menu.</p> <p>Use empty in the edit box means empty folders will be included in the menu (normally, they are excluded).</p> <p>Use nosubdir in the edit box means no subdirectories will be included. The names of subdirectories are still shown; if selected, a *Folder Contents Menu is shown for that subdirectory.</p>
Explorer	<p>Use nofiles in the edit box means only folder will be shown and not files; useful with the explorer option to traverse large folder trees.</p> <p>Place explorer in the edit box to add a menu entry "Explore" to all submenus; left clicking on it will open a single-pane Explorer window on the selected directory and right clicking will show an *Folder Contents menu for the folder (useful with nofiles). Uncheck "Switch to if active" to allow new Explorer window to open if explorer is already running.</p> <p>Place explorer2 in the edit box to add a menu entry "Explore2" to all submenus; left clicking on it will open a double-pane Explorer window on the selected directory and right clicking will show an *Folder Contents menu for the folder (useful with nofiles).</p>
Icons	<p>Place noicons in the edit box to omit menu icons (only works if the Folder Contents menu is not embedded in another menu).</p>

- Execution** Place ***all** in edit box to execute all commands, rather than displaying a menu.
Place ***allclose** in edit box to close all commands, rather than displaying a menu.
- Embed** Place **embed** in the edit box is used if the *Folder Contents command appears in a menu: it causes the menu entries to be embedded within that menu rather than appearing when the *Folder Contents command is selected (embed must be in lower case).
- Position** Place **center** in edit box to center menu on screen.
Place **under** in edit box to center menu under mouse.
- File date** Putting a number **n** in the work directory edit box means that only files accessed more recently than **n** days ago will be included.
Placing **sortext** in the edit box means items will be sorted by file extension.
- Command** **Stiletto** normally runs the file selected from the menu by running the associated command; you can instead specify the command and any parameters by putting cmd followed by the command and parameters:
cmd "c:\path\cmd /p1 /p2"
Note that the command and parameters must be enclosed in double quotation marks. Furthermore, if the command file path contains blanks, it must be enclosed in single quotation marks, eg:
cmd "'c:\windows path\mplayer' /play /close"
can be used to play .wav files with mplayer.
- Extension** To include files with only certain extensions, list the extensions separated by blanks including the initial period.

To exclude files with certain extensions, list the extensions to be excluded, separated by blanks, and include a - in front of the period of each extension.

Examples:

autosoftcol 2 offset -15 0 maxtext 5

Start a new column every 2 entries; limit labels to 5 characters, and offset 15 characters to the left of the cursor.

nosubdir .exe 15

Include .exe files accessed less than 15 days ago from main directory

.xls nosubmenu

Include Excel spreadsheets from all subdirectories on one menu.

-.dll -.bak

Exclude dll and bak files.

Special Folders for Folder Contents Menu

Using the built-in Folder Contents Menu command, you can display a menu of the special folders used by Win95/NT 4. To access special folders, the parameters edit box for this command can contain one or more of the following (separated by commas).

start menu	start menu entries
desktop	shortcuts on your desktop
recent	recently accessed documents

templates	standard document templates
personal	personal favourites folder
programs	menu of all program folders (current user for NT4)
programs xxx	menu of programs folder xxx (eg Accessories)
Allprograms	menu of all program folders
Allprograms xxx	menu of programs folder xxx for All Users profile (NT4 only)
control panel	explorer view of control panel entries
my computer	explorer view of my computer entries
network	explorer view of network neighborhood entries
recycle bin	explorer view of recycle bin entries
printers	explorer view of printers

With NT4 and for versions preceding 97b, **Stiletto** would use the programs associated with the **administrator** profile. Starting with 97c, **Stiletto** uses the currently logged on user, except that the All Users profile is used for AllPrograms. You can return to the 97b behavior using the UseAdmin internal option. **Stiletto** assumes these folders are under your Windows system directory.

You can use in the SystemFolderDir internal option to specify the directory.

When using Win 95 Start Menu, it is possible to create sub-menus of the contents of special folders like control panel by creating folders with names of the form Control panel {...}. It is possible to achieve a similar effect in Stiletto as follows:

Create a folder called (say) control

Use explorer to copy all the entries from the explorer control panel special folder to that new folder. You should get a series of shortcuts (links).

Create a Folder Contents Menu command and reference the control directory. Embed the folder contents menu command in a menu or submenu. You should get a menu of all your control panel applets, like the one you can get on start menu

Virtual Desktops

Use virtual desktops if you run many programs at the same time and want to reduce desktop clutter. A virtual desktop is a collection of windows which you show and switch-to as a group. Only windows on the active virtual desktop are visible.

Normally, when you shutdown **Stiletto**, all desktops are lost. But if you use the saved desktop feature, **Stiletto** will remember all the programs which make up the current desktop and you can re-launch that desktop by its assigned name later.

Configuration

You define and switch-to a virtual desktop through a menu that you access by Ctrl+right-clicking anywhere on the **Stiletto** button bar or by activating the built-in command *Virtual Desktop. You can switch desktops, create new desktops, lock/unlock windows on desktops, move windows between desktops, close and rename desktops, and create saved desktops.

See below for an explanation of the menu contents.

In addition to the menu, you can also use the *Virtual Desktop command to work with desktops, by associating this command with a button, menu item, hot key, and so on. Use the parameter field as follows:

The command

Command *Virtual Desktop

Parameter: *arrange

shows all nine potential desktops and allows you to drag/drop windows among desktops, create/delete desktops, and lock/unlock windows. You can access a control menu by right clicking. The active desktop named is shown in bold. You can also double click on the list of windows in a desktop to close the *arrange dialog and switch to that desktop or double click on the name of a desktop to switch desktops without closing the *arrange window. If you work with fewer than nine desktops, you can change the arrange dialog's height (but not its width).

While the *arrange dialog is open, you cannot access other *Virtual Desktop commands or the virtual desktop menu.

To switch to a virtual desktop named **mydesk** from a button or menu entry, include the desktop name with the *Virtual Desktop command:

Command *Virtual Desktop

Parameter: mydesk

If mydesk is running, it is activated. If mydesk is a saved desktop and it is not running, it will be launched. If mydesk is not a saved desktop and is not running, a new desktop called mydesk is created.

You can also specify the desktop by number; eg to switch to the third desktop use:

Command *Virtual Desktop

Parameter: 3

You can also use the following entries in the parameters edit box of a *Virtual Desktop command:

*new	creates a new desktop called desk n
*new xxx	creates a new desktop called xxx
*clear	closes all windows on the current desktop
*rerun	closes all windows and restarts the desktop (if it is saved)
*next	switch to next desktop
*prev	switch to previous desktop
*show	shows the See All menu
*switch	shows the See All/Switch menu
*arrange	drag/drop windows between desktops, create/delete desktops, rename
*init n	create n empty desktops called desk l;

You can put the keyword **icon** after *switch or *show to include icons on the shown menu. You can put a number n after *switch to limit text characters to n characters. You can put the keyword **mne** to have **Stiletto** generate mnemonic characters. For example

Command *Virtual Desktop

Parameter: *switch icon 8 mne

shows the *switch menu with icons and menu mnemonics and maximum menu name length of 8 characters.

Further Information

If you activate a window which is on a hidden desktop (eg via tray icon), **Stiletto** will activate that desktop. If you want to avoid this, uncheck "Show Virtual Desktop if any of its windows is activated" on Special dialog.

You can have up to 9 active desktops.

You cannot switch desktops while the configuration tabbed dialog is open.

You can show the name of the current desktop as a button label.

It is possible to show a different **Stiletto** bar for each desktop. Create new bars and start them with the desktop you want them to be associated with. Note that your main bar (the one started first) is visible on all desktops.

Be careful when using multiple button bars with virtual desktops: all desktop switching and virtual desktop configuration must be done from the same bar.

You can define a command which will display a menu which depends on the currently active virtual desktop:

Command *Show Menu

Parameter *desk

will display the menu with the same name as the currently active virtual desktop (no @ at start).

You can change to specific wallpaper, play a special sound, hide/show desktop icons or the task bar, and so on, for saved desktops by checking the "Relaunch programs in saved desktops" checkbox on the special configuration dialog. If this box is checked, each time you switch to a virtual desktop, **Stiletto** will re-execute all commands in the menu where "switch to if active" is **not** checked. So, for example, if you included a *change wallpaper command in the saved desktop with "switch to if active" checked, then **Stiletto** would execute the command and change the wallpaper each time you activated the desktop.

Saved virtual desktops are actually menus of programs which **Stiletto** launches as a group. If you want to change the contents of an existing saved desktop, you will use the menu configuration commands. **Stiletto** adds an at-sign (@) to the start of the menu name of a saved desktop.

The *init entry in the parameter edit box is often used on a startup script to create empty desktops. If you have other programs starting at the same time as **Stiletto**, you may need to put a *wait before this command to let these other programs get started and stabilized.

Explanation of Virtual Desktop Menu

List of Defined Desktops

Select one of the desktop names on the menu to show the windows on that desktop.

New Desktop

Hides all the windows on the current desktop and creates a new one. You can name the new desktop with the rename menu entry, if you want.

Arrange

Shows all nine potential desktops and allows you to drag/drop windows between desktops, create/delete desktops, rename desktops, lock windows on all desktops.

Lock

Shows a list of windows on the current desktop. Selecting a window locks it. A locked window appears on all desktops. The menu item is only enabled when there are windows on the desktop which can be locked. You can also pre-specify locked windows using the "Show on All Virtual Desktops" edit box on the Window Control configuration dialog.

Unlock

Shows a list of locked windows. Selecting one unlocks it. The menu item is only enabled when there are locked windows.

Remove From Desktop

Shows a list of windows. Selecting one removes it from the current desktop.

Move/Copy from this

Shows list of windows. Selecting one causes menu of desktops to be shown; selecting a desktop from this list moves the selected window to that desktop (hold down Ctrl to copy the window). Only enabled if there is a windows which can be moved and there is more than one desktop.

Clear this Desktop

Closes all windows on the current desktop. If the windows only appear on this desktop, the corresponding programs are closed.

Clear and relaunch saved

Closes all windows on the current desktop and restarts the programs which make up the saved desktop of the same name. If the windows only appear on this desktop, the corresponding programs are closed.

Close and move windows to

Closes current desktop and moves its windows to selected desktop. Only enabled if there is another desktop besides the current one.

Rename Desktop

Allows you to assign new name to desktop while it is active.

See All/Move/Copy to this

Shows the names and window captions of other desktops and allows you to copy/move a window to the current desktop.

The active desktop name is show in round parentheses, eg (**mydesk**); other desktop names are shown in angle brackets, eg <**otherdesk**>. Select a window name to move that window to the current desktop or hold down the Ctrl key while selecting a window name to copy it.

You can embed this menu in a menu by putting **embed** in lower case in the work directory edit box (***show** must also be in lower case).

You can change to n number of rows on the See All menu by putting DeskShowRows=n under [General] in stiletto.ini file (with **Stiletto** shut down). Use n=99 for a single column.

See All/Switch To

Shows the names and window captions of other desktops and allows you to switch to another desktop and activate a window on that desktop.

The active desktop name is shown in round parentheses, eg **(mydesk)**; other desktop names are shown in angle brackets, eg **<otherdesk>**. Select a desktop name to switch to that desktop and activate the last window which was active. Select a window name to switch to that desktop and activate that window.

You can embed this menu in a menu by putting **embed** in lower case in the work directory edit box (***switch** must also be in lower case).

You can change to n number of rows on the See All menu by putting DeskShowRows=n under [General] in stiletto.ini file (with **Stiletto** shut down). Use n=99 for a single column.

New Saved Desktop

Creates a saved desktop from the current visible windows. Saved desktops are stored as menus with a @ at the start of the name.

Start Saved Desktop

If a desktop of the specified names exists, switches to it. Otherwise creates a new desktop (unless the current desktop is empty). Then all the programs on the saved desktop are started.

Replaced Saved Desktop

Replaces a saved desktop from the current visible windows. Saved desktops are actually menus, so this is done through the menu contents tab.

Modify Saved Desktop

Select the desktop from a submenu and you can edit its entries. Saved desktops are actually menus, so this is done through the menu contents tab.

Delete Saved Desktop

Removes the selected saved desktop.

Copying Buttons on the Bar

You can copy the contents of buttons on a button bar with the built-in command

```
Command    *Copy Buttons
Parameters  n1 n2 n3
```

copies buttons n1 through n2 inclusive to buttons starting with n3. For example:

```
Command    *Copy Buttons
Parameters  22 24 8
```

copies buttons 22 through 24 to 8 through 10.

None of the buttons to be copied can be active bar buttons.

You can use this command to dynamically change the visible buttons. Use it with Virtual Desktops to show different buttons for each virtual desktop:

```
Command:    *Virtual Desktop
Parameter   desk1<*copy 20 23 5
```

switches to virtual desktop **desk1** and copies buttons 20 through 23 to 5 through 8. Presumably, buttons 20 through 23 make available special commands for desktop **desk1**.

You can make the range of buttons to be copied depend on the active window by listing the .exe file names of the programs to be checked:

```
Command    *Copy Buttons
Parameters  wordpad=25 27 8 winfile=22 24 8 *=28 30 8
```

copies buttons 25 to 27 to button 8 through 10 if explorer is active; buttons 22 to 24 to 8 through 10 if File Manager is active, and buttons 28 to 30 to 8 through 10 otherwise. Put *= last to cover the default case. If *= is omitted and none of the listed programs are active, no copying is done.

If you assign a *copy command to the middle mouse position of the 30th button of a bar positioned in the caption, then that *copy command will be automatically executed each time that the bar moves. This allows you to create a tool bar which varies depending on which window it is in.

Scripts: Launching or Closing All Commands on a Menu

It is possible to develop scripts of commands you want to run as a group. Within these scripts, you can use *wait, if, and jump commands to wait for some condition or to create loops and conditionally execute commands.

Scripts are configured using menus.

You run a script of all the commands on a menu at once by:

```
Command    *Script
Parameters  call mymenu
```

launches all commands on mymenu or shows them if they are already running and "Switch to If Active" is clicked on the command entry controls. Put the menu name in double quotes if it contains spaces.

Stiletto normally starts executing with the first command on the menu. But if you put the menu item label of a menu item after the menu name, Stiletto will start at that entry:

```
Command    *Script
Parameters  call mymenu cmd2
```

runs all commands on mymenu starting at the one labelled cmd2. Put the menu item label in double quotes if it contains spaces.

To close all programs

Command *Script
Parameters close mymenu

Programming Scripts

Stiletto normally executes all commands until the end of the menu, but you can stop execution by the command:

Command *Script
Parameter quit
You can use return instead of quit.

To make it easier to maintain batch menu commands, you can pack different commands onto one menu: start each command sequence with an identifying label and end each command with a quit.

You can call another menu by including a *Script call command in the calling menu.

You can further program batch menu commands with jumps and if-conditions.

Command *Script
Parameters jump xxx
goes to label xxx of the currently executing menu for the next command.

Command *Script
Parameters if keyword text
executes the following command only if the condition specified by the keyword and text is true. If the condition is false, the next command on the script skipped. The next command skipped refers to the remainder of the multiple commands if the *Script if is not the last command (or the only command); on the other hand, it refers to the next line in the script menu if the *Script if command is the last command or the only command in the command entry controls.

The keyword can be any of **window**, **nowindow**, **active**, **noactive**, **path**, **nopath**, **modem**, **nomodem**, **alt**, **noalt**, **ctrl**, **noctrl**, **shift**, **noshift**; these are explained in wait command. It can also be **activevdesk** which is true if the active virtual desktop name is **text** or **noactivevdesk** which is true if the active virtual desktop is **not text**. You can use **if saver/nosaver** to check if a screen saver is running/not running. In all cases, put **text** in quotes if it contains blanks.

You can also check for the existence of a file with

Command *Script
Parameters if filepath "path"
Path can contain wildcards, in which case the if checks for any file matching the path. Use **nofilepath** to reverse the logic.

To help with Script programming, **Stiletto** has a set of 32 flags which you can manipulate and test. To set a flags n1 through n2:

Command *Script
Parameters flag set n1 n2
where n1 and n2 are any numbers between 0 and 31. You can omit n2 if you only want to access one flag. To clear flags, use **flag clear n1 n2**; to toggle (reverse) the setting, use **flag toggle n1 n2**. Use **0 31** for **n1 n2** to access all flags.

You can test the flag with the **if** command; use

Command *Script
Parameters if flag n
to check to see if flag number n is set; use **if noflag n** to check to see if it is clear.
sets variable n to **string**. The variable name n must be a single digit. Use ? for "string" to prompt for input.

Stiletto has a set of 26 variables you can set in scripts and then use in any command. The variables are named by the letters a-z. Set them with the command:

Command *Script

Parameters set a "string"
sets variable a to **string**. The variable name a must be a single letter. Use ? for "string" to prompt for input.

Add a number to a script variable with:
Command *Script
Parameters add a num
where num is a positive or negative integer.

Multiply a number and a script variable with:
Command *Script
Parameters multiply a num
where num is a positive or negative integer.

Use a variable anywhere in the command or parameter edit box of any command (not necessarily in the script). You must define a script variable character using the advanced dialog; suppose you use & then the variable x can be inserting with &x. Put &x in quotes if the variable value could contain blanks. Note that you do **not** use the & in the set command.

Use {sv} to send the script variable character (eg &) when sending keys.

You can test variable for equality with if equal (notequal for inequality). For example
Command *Script
Parameters if equal "&e" "four or 4"
tests e to see if it is **four or 4**. Note use of quotation marks.

You can use the wait command in a menu started with *Script call to wait for some condition. For example, you could run a dialer, wait for the modem, then run a program which accesses the modem.

Dynamically Changing Button Color

You can dynamically change the button color with the built-in Change Button Color command. This command turns the buttons Own Colour flag off or on; (see configure button for details), unless a button has the Freeze checkbox set, in which case its colors cannot be changed dynamically.

Use *color when combining in multiple commands.

To dynamically change button colors, create a button or menu with the Change Button Color command and put a string with the following structure in the parameters edit box:

```
[r] buttons-changed new-colours
```

where

- r optionally, start with the letter **r** to reset own color to unchecked for all non-frozen buttons
- buttons button number to change; leftmost (topmost) button is 1:
 You can use a range of buttons, like 10-12.
 You can use * for all buttons.
 You can use the letter **b** to stand for the **Stiletto** button from which the Change Button Color command was launched.
- new-colours is used to set the new face and text colors:
 Use three numbers between 0 and 255 to specify the Red, Green, Blue components of the colours (eg 255 0 0 for dark red). See Control Panel|Appearance|Tool Tip|Color for a palette to view various RGB combinations).
 You can use = instead of the three numbers to leave the color to the value set with the button configuration dialog.
 You can leave out both triplets of RGB value to set both colors to those set in the configuration dialog.
 You can use the letter **t** to toggle the Own Color check box, ie to flip between own color and bar color.

You can use the letter **r** to reset the Own Color flag, ie set the colors back to the default.

Examples of parameters box

```

b          Set the own color setting of button from which command was launched..
b t       Toggle own color setting of button from which command was launched.
b r       Clear own color setting of button from which command was launched.
10-12 127 0 0   Set face of buttons 10, 11, 12 to medium red
r 1 = 255 255 255   Reset all buttons, then set text of button 1 to white

```

Dynamically Changing Button Text

Use the command
***Change Button Text**
n text
to change the button text on button n to "text"

Dynamically Changing Button Icon

Use the command
***Change Button Icon**
n filepath i

to change the icon on button n to the ith icon in the file filepath. Put filepath in double quotes if it contains blanks. For example,

```
*Change Button Icon
2 "c:\my path\my.exe" 1
```

changes the icon on button 2 to the first one in c:\my path\my.exe.

You can alternate between icons with

```
*Change Button Icon
n path1 i1 path2 i2
```

which sets the icon to the i1th one in path1 unless it already is, in which case it is set to the i2th one in path2. For example,

```
*Change Button Icon
3 c:\windows\system\moricons.dll 3 c:\windows\system\moricons.dll 8
```

alternates between the third and eighth icons in the file (file1 and file2 need not be the same in general).

Filling a Menu with All Active Windows

You can create or change a menu and fill it with the commands corresponding to all currently open windows which are visible and not minimized.

Use the built-in All Windows to Menu command to do this.

Create a button or menu entry with this command and put the name of the menu you want to create in the command parameters edit control. When you execute the All Windows to Menu command, this menu will be created and populated with the exe files corresponding to all active, visible, non-minimized windows.

If the menu already exists, you will be asked if you want to replace it. To avoid this confirmation prompt, put the word ***overwrite** after a space following the menu name in the parameters edit box.

Scrolling with the Mouse

Purpose

You can scroll windows vertical or horizontally using mouse movements. This avoids having to move the mouse to the scroll bar to scroll the window. You can scroll either automatically or manually. Automatic scrolling scrolls the window even when the mouse is not moving; manual scrolling requires mouse movement to scroll the window.

Automatic Scrolling Configuration

To scroll windows automatically, execute this command:

```
Command    *Stiletto Exec
Parameter  autoscroll
```

A small gray rectangle with the letter "s" will appear in the current window. Move the mouse above it to scroll up; the further the mouse is from the gray rectangle, the faster the window is scrolled. Move the mouse below the rectangle to scroll down; the further the mouse is from the gray rectangle, the faster the window is scrolled. Left click to stop scrolling.

Many newer programs support autoscrolling internally if you middle click on one of their windows. To automatically take advantage of them, define a hot key (say tap shift) which sends a middle mouse click to the programs which support autoscroll and executes the **Stiletto** command otherwise. To do this, define the hot key twice, and use the target window feature on the second definition. For example, to use native autoscroll in Internet Explorer:

```
Hot Key:    tap shift
Command    *Stiletto Exec
Parameter  mouse middle
Target     *Internet Explorer
```

```
Hot Key:    tap shift
Command    *Stiletto Exec
Parameter  autoscroll
Target
```

Manual Scrolling Configuration

You can start manual scrolling in one of two ways: by attaching a command to a hot key/mouse action or by the middle mouse button.

To start manual scrolling with a hot key, attach this command to the key:

```
Command    *Stiletto Exec
Parameter  scroll
```

Scrolling only occurs for the window which the mouse is over when the hot key is activated. Scrolling continues until the left mouse button is clicked.

To set up middle mouse scrolling, use the Special configuration dialog. Check the Scroll checkbox there to scroll only while middle mouse down; gray-check to scroll with middle mouse up until left button clicked.

To scroll a window, activate the *Stiletto Exec scroll hot key or hold down middle mouse and move in desired direction. For ordinary check, scrolling will continue even if the mouse stops. For gray check of the command, scrolling will pause unless the mouse is near the top or bottom of the window. You can control or disable speed of automatic scrolling with Scroll interval Advanced option.

To scroll pages (instead of single lines), click the right mouse button while the window is scrolling.

To quickly move to the start or end of the file, hold the Alt key down and move the mouse in the desired direction.

Mouse scrolling only works with applications that use standard windows scroll bars.

For ordinary check, you can use the internal option ScrollReset to have **Stiletto** return the mouse cursor to the starting point of the screen after the window scroll is completed.

Some applications, such as Microsoft Internet Explorer, already support mouse scrolling. You can disable **Stiletto** scrolling for these or any window by typing the caption of the window in the edit box beside the middle scrolling check box. Separate captions of different programs by commas. Normally, you will not type the whole caption, but rather only a part. Use ***xxx** to match all captions ending in **xxx**. Use **xxx*** to match all captions starting with **xxx**. For example, *Internet Explorer will match MS IE windows.

Moving Stiletto

Although it is generally more convenient to have the **Stiletto** button bar at a fixed place on the desktop, there are times when you want to temporarily move **Stiletto** out of the way.

The built-in Move **Stiletto** command does this. When executed, this command causes the button bar to jump to the other side of your Windows desktop. If you execute it again, **Stiletto** jumps back to its previous position.

(Or you can also have a floating button bar through the Position **Stiletto** built-in command). The Move command can be assigned to a button on the button bar or put in the Menu. The Move command is only temporary; the next time **Stiletto** starts up, it will be displayed in its standard position.

Move handles the split button bar as follows: if the move command is assigned to the left half, then the button bar will be fused on the right half of the desktop. If the move command is assigned to the right half of the button bar, then the button bar will be fused on the left half of the desktop. (Hence you may want to assign the move command to two different buttons.) When the move command is pressed from a fused button bar, it will be resplit. Although this may seem complicated, the idea is that a single mouse click near the area you want to clear will move **Stiletto** out of the way.

Drag and Drop onto the Stiletto Button Bar

You can **left** or **right** drag and drop a set of one or more file names from the Explorer/File Manager or Explorer onto the **Stiletto** bar to start a command with the file names as the parameters.

Left drag and drop starts a command with dropped file(s) as the parameter. Right-drag/dropping file(s) onto the bar activates a menu allowing you to select the button to receive the file or to be configured. Alternatively, you can specify that a right drag/drop should activate the right-button command on the selected button using a check box on the Bar dialog.

For **left** drag and drop, normally the command corresponding to the left mouse button is started. But if you hold down the **Shift** key as you drop the file name, the command corresponding to the middle button is started. If you hold down the **Ctrl** key, the command corresponding to the right mouse button is started. (As another memory aid, consider this: both "Shift" and "Middle" have "i" in them; both "Right" and "Ctrl" have "r" in them.)

(**Stiletto** does not check on the **Shift** or **Ctrl** keys until after you drop the file. If you are dropping a file from a floppy disk, you need to hold the key down for a moment after the file is dropped since **Windows** does not tell **Stiletto** that the file is dropped until after it checks the floppy drive which takes a second or two.)

You do not have to memorize this rule: you can **right** drag and drop a file onto any button and you will be able to select the command to receive the file with a menu.

Stiletto has a built-in drag and drop rename command, Show Wallpaper command for previewing wallpaper, and Play Sound command for playing wav files.

Sometimes you want to drag and drop files in the middle of the command line. To do so, put the character "|" at the point where you want the dropped files to be placed. The "|" will be replaced by the dropped files when the command is run and the text following the "|" will follow the dropped files. Do not forget a space after the "|", if needed.

Stiletto drag and drop also works with the **Norton Desktop for Windows** file manager and the **SideBar** file manager.

You can drag and drop files to the Active Task Buttons, and they will be passed to the executing program (if the program does not accept dropped files, you will hear an error beep).

Stiletto always attempts to start a new instance of a command when a file is dropped on a button.

Drag and Drop Rename

The built-in drag and drop rename command can be used to rename a file with a new name or extension. The renamed file stays in the same directory. Long file names and extensions can be used.

File wildcards are not allowed.

You can associate this command with a menu or the button bar, but if you want to use the drag and drop capability, the command must be assigned to a button.

Activating the command brings up a small dialog box where you can type the current file path (drive, directory, file name, extension), the new file name, and the new extension.

If you drag and drop a file name from Explorer/File Manager onto the button with the rename command, the current path edit box will be initialized to the dropped file name and the new file name and extension edit boxes will be initialized to the dropped file name and extension.

If you Press the "Ext to .bak" button, the new file extension is set to .bak and the new file name is set to whatever was present in the new file name exit box.

Accessing Program Manager or Other Shell Groups

For NT.351, you use the built-in Prog Mgr/Shell Groups command to run commands from your existing program manager or other desktop shell groups. (Use *Folder Contest Menu programs for Win 95/NT 4). The Prog Mgr/Shell Groups command displays a dialog box with two list boxes. The left list box shows the names of all your groups. Clicking on any one of these will fill the right list box with the names of all the commands (ie icons) in that group. You can then execute any command in the group by either double clicking on it, or by selecting it with a single click and pressing the Execute button.

The command dialog is normally closed when you run a command; however, if you uncheck the Close When Command Run check box, the dialog will stay open after a command is run.

If you single click on any command name in the right list box, you will see the file paths for the command and its work directory displayed at the bottom of the dialog box.

You can set up the command to immediately display the commands for any group when started by typing the name of the default group into the work directory field of the command entry controls. For example, if you had several groups you accessed often, you could set up several different commands, one for each group, and assign them to a menu with the menu name set to the group name. Then, when you selected that item from the menu, the Groups command would start with the right hand list box filled with the commands from the default group.

Cut and Paste Commands

For NT 3.51 you can copy program manager or other desktop shell commands directly to the **Stiletto command entry controls** used to enter commands for button configuration, menu, hot keys, timers, and alarms. (Use Capture button for Win 95/NT 4 to capture from start menu).

Cut and Paste uses the "P" button on the command entry controls. This button will be enabled whenever there is information from a program manager or desktop shell group to be pasted: Each time you select or execute a command from the groups command, the command name and work directory are made available through the "P" button. So, to paste a command, you first need to have selected the command from the program manager or desktop shell group dialog.

Follow these steps to cut and paste commands:

Start the Configuring Stiletto built-in command.

Press the Program Mgr/Shell Groups button on the Info tabbed dialog.

If needed, move the Program Manager Groups dialog box so you can access the configure dialog. Access the command you want to configure through the button configuration, menu, or alarms tabbed dialogs.

Select the program manager group and command you want to paste (single click!).

Press the "P" button on the command entry controls to copy the information.

Save the copied command on the menu, if appropriate.

Repeat the above three steps for all commands you wish to copy.

Running Multiple Stiletto Button Bars

Purpose

You can have many **Stiletto** button bars; each separate button bar uses its own configuration file. For example, if you use virtual desktops, you can have a different bar for each desktop. Or you could have a main bar and a small bar positioned in the caption.

Configuration

To set up a new configuration file, use the button on the Info dialog. This allows you to create a new bar and name it. For this example, suppose you choose the name **mybar**. Creating the new bar will show it in the bottom middle of your screen so that you can configure it.

Once you are happy with the configuration, you can re-start the **Stiletto** bar at any time with this command:

```
Command:    c:\stilpath\stiletto.exe
Parameter:  mybar.ini
```

Do not use a directory path before mybar.ini. If you want your button bar to start automatically, assign the above command to the **Stiletto** startup menu of your main bar. You can also start bars manually by assigning the above command to (eg) a button, or you can start bars with saved virtual desktops by adding the above command to the menu entry for the saved desktop.

If you wish, you can continue to create other button bars by creating a separate configuration file for each. All configuration files are stored in the same directory as the **Stiletto** .exe file.

Further Information

A default configuration file, stiletto.ini, is used when you run **Stiletto** without a configuration file on the parameters edit box. The name of the configuration file **Stiletto** is using is shown in the Info dialog.

Hotkeys, the desktop/title bar mouse buttons, and application sounds are determined by the first bar that requests them. Each bar will separately process chimes, alarms, wallpaper and sound randomization, and resource warnings. Hence, you should isolate your use of these features to one main bar. This will also conserve your **Windows** resources.

When you run multiple bars and close the initial bar, all other bars are closed as well.

If you want to start multiple bars at Windows Startup, you should put your main bar in the Windows Startup folder and start the remaining bars from the start up menu of this main bar. Starting multiple bars from the Windows Startup folder can lead to timing-related problems in **Stiletto**.

To remove a bar, delete the .ini, .bki, and .udo files in the **Stiletto** folder (eg mybar.ini).

It is possible to put .ini files in a separate folder from the exe file; see below for details.

You can also manually create new bars using the blank button bar file called blankini.def which is part of the **Stiletto** zip package. Copy this file to newbar.ini (or any other file name you wish) using a Dos copy command:

```
copy blankini.def newbar.ini
```

Stiletto Command Line

You put the name of the **Stiletto** configuration (".ini") file on the **Stiletto** command line. If you omit it, **Stiletto** assumes the configuration file is called stiletto.ini and is in the same folder as the Stiletto exe file.

To separate data from program files, you can put the configuration file in a separate folder from the exe file by putting the full path to the ini file on the command line:

```
"c:\program files\Stiletto\Stiletto.exe" "C:\My Documents\Stiletto\Stiletto.ini"
```

If you use a shortcut to start **Stiletto**, the command line can be found in the shortcut properties. Note that you must put double quotes around file paths which contain blanks.

To start the program from **Stiletto** command entry controls:

```
Command      c:\program files\Stiletto\Stiletto.exe
```

```
Parameter    "C:\My Documents\Stiletto\Stiletto.ini"
```

Stiletto puts all files which it can change into this folder as well: the timer log, alarm log, clip folder, tray icon info, saved desktop icon positions, saved explorer windows (from explorer tracking option).

Restarting Stiletto with a new Configuration

You can dynamically reconfigure any **Stiletto** bar to use a new configuration file in two ways:

Manually: by executing a command

Automatically: depending on the active program

Manually Reconfiguration of Stiletto

Use the *Reconfigure **Stiletto** built-in command to reconfigure any **Stiletto** bar to use a new configuration file. First use the procedures for multiple bars to create a set of configuration (.ini) files that you want to use. You then have the following options:

To restart the current bar from its existing .ini file:

Command: *Reconfigure Stiletto

Parameter

The existing .ini file is re-read and **Stiletto** reconfigured based on its contents. Use this command if you change the ini file using some other program.

To restart the bar with configuration file other.ini and use the new file new.ini:

Command: *Reconfigure Stiletto

Parameter (other.ini) new.ini

The running bar with which has configuration file other.ini is reconfigured to use new.ini.

To restart the nth bar (n is any digit between 1 and 8):

Command: *Reconfigure Stiletto

Parameter 3 newbar.ini

The bar which was started 3rd will be reconfigured to use newbar.ini.

To restart the other bar:

Command: *Reconfigure Stiletto

Parameter =newbar.ini

If only one bar is running, a second bar with configuration file newbar.ini is started. If exactly two bars are running, the other bar is reconfigured to use newbar.ini. If more than two bars are running, the bar to use newbar.ini is determined as follows: If the current bar was the first **Stiletto** bar started, the second bar is reconfigured. Otherwise, the first bar started is reconfigured.

Automatically Reconfiguration based on Active Window

You can create a **Stiletto** bar with a configuration which depends on the active window.

Using the procedures for multiple bars, create a set of configuration files named after the .exe files of programs you want to create tool bars for. For example, to create tool bars for Netscape Browser, Excel, and Notepad, create configuration files named netscape.ini, excel.ini, and notepad.ini. In addition, create a configuration file default.ini which will be used for all other active windows.

All configuration files must reside in your main **Stiletto** file folder.

All configuration files must have "Automatically reconfigure based on active program" checked on Bar tab of configuration dialog.

Now start the default bar by

Command: c:\yourpath\stiletto.exe

Parameter default.ini

For example, you could include this command in your Windows StartUp group or your **Stiletto** Start menu.

Each time the active window changes, **Stiletto** switch the configuration file appropriately.

Showing and Moving the Bar Temporarily

You can show a hidden or covered bar or move the bar temporarily to the mouse cursor with the built-in *Show/Move **Stiletto** command.

To show the bar only

Command: *Show/Move Stiletto

Parameters:

shows/uncovers all bars.

Command: *Show/Move Stiletto

Parameters: bar.ini

shows/uncovers the **Stiletto** program with configuration file bar.ini.

To temporarily move the bar to the mouse cursor (usually through a hot key):

Command: *Show/Move Stiletto

Parameters: *move

moves the bar which executed the command to the mouse cursor.

Command: *Show/Move Stiletto

Parameters: *move2

If two bars are running, moves the other bar to the mouse cursor.

Command: *Show/Move Stiletto

Parameters: *bar.ini

moves the bar with configuration file bar.ini to the mouse cursor.

Moving the bar also shows it.

The bar moves back to its original position after the mouse cursor is moved off it and the delay given by the "Delay for autohide/move" on the Bar configuration tab has elapsed.

Position Stiletto

You can position **Stiletto** manually anywhere on the desktop by using the Position **Stiletto** built-in command.

First assign this command to any **Stiletto** button and to any of the left, middle, or right mouse buttons. Then use the bar dialog to select any standard vertical or horizontal position, depending on whether you want **Stiletto** to be oriented vertically or horizontally. Finally, click on the button with the Position command using the appropriate mouse button and drag **Stiletto** to wherever you would like it to be positioned. After you are done dragging, **Stiletto** will record its position in the configuration file and will return to this position automatically when started.

Caps Lock and Scroll Lock

The Special GUI dialog contains check boxes to permit you to control the behaviour of the Scroll Lock and Caps Lock keys.

You can specify that pressing shift always clears caps lock, to avoid reversed mixed case like **sTILETTO**. Or you can disable the caps lock key completely.

By setting the "Shift Clears Cap Lock" check box to the gray-checked state, you specify that shift should clear caps lock only when a letter is pressed with shift.

You can also completely disable the caps lock key.

You can disable the Scroll Lock key. This key is rarely used, and when activated unknowingly, causes irritating behaviour from the arrow and other keys.

When using this feature with multiple instances, make sure than only one instance uses it and hotkeys.

Hiding Windows

You can use the Hide Window built-in command to hide windows. You might use this if you do not want a window to appear on the list of active windows or the Active Task Buttons.

When you execute a Hide Window command, the cursor changes to a cross. Left click on the window you wish to hide. This window, its top-level parent, and all the parent's children will be hidden.

You cannot hide a **Stiletto** window or the desktop window. Some other programs will also refuse to be hidden.

If you execute Hide Window but then decide you do not want to hide a window, left click the mouse on the desktop or on a **Stiletto** window to cancel the operation.

If you want to show a hidden window, configure the **Stiletto** list of active windows to show hidden windows using the Menu Setup dialog, assign this command to a menu, then execute the menu and select the hidden window from the list.

Working with Tray Icons from Other Programs

Purpose

You can use the *Tray Icon command to simulate mouse clicks on the tray icons from any other program. You can also use this command to hide these tray icons and still access the commands by simulated mouse clicks.

This lets **you** decide how to access tray icon functions and which tray icons should appear in the tray window on your taskbar.

Configuration

Before you can access a tray icon, you must train **Stiletto** on how to access the icon. See below for information on how to do this. You have to train **Stiletto** once for each icon you want to access.

Once you have trained **Stiletto**, you send mouse clicks to the icon with the following command:

Command *Tray Icon

Parameters icon_name click keystrokes

icon_name is the name you assigned to the icon when you trained **Stiletto**

click is one of lc (left click), ll (left double click), mc, mm, rc, or rr.

keystrokes is optional; if present, it is a set of keystrokes to send to a menu resulting from the click (if a normal window results from the click, use *Keys with multiple commands instead).

You can also hide the tray icon with the command:

Command *Tray Icon

Parameters icon_name hide

You can still send mouse clicks to a hidden icon.

Further Information

You can use the command wizard "Access tray icons of other programs" dialog to see the list of all valid icon names or to delete an icon name.

Training Stiletto to Recognize Tray Icons from Other Programs

To access tray icons from other programs, you must first train **Stiletto** to recognize the hidden the hidden window and internal codes that this icon uses. Follow these steps:

1. Make sure the tray icon to be accessed is visible in your tray. It is helpful to shut down other windows, but this is not necessary.
2. Use the "Add New" button on the "Access tray icons of other programs" of the command wizard, **or** Ctrl+left click on the button bar and press "Capture Tray Icon Info" on Info tab.
- 3 You will get a message box prompting you to left click on the tray icon. Press OK on this message box and then left click on the tray icon.
4. If **Stiletto** is able to capture the information, you will get another message box reporting success and asking you to help confirm that the information was correctly captured. Press OK and **Stiletto** will simulate a right click on the icon as a test.
5. If the right click test succeeds, **Stiletto** will ask you to enter a name for the icon information. This is the icon_name field used in the *Tray Icon command or selected from the drop down in the command wizard.

If **Stiletto** cannot capture the left click on the icon, or if the right click test is not successful, try again once or twice to ensure that this was not just a transient problem.

Positioning Windows with Stiletto Exec

You can position a window at a specific location or at the top, bottom, left, or right half of the screen with

Command Stiletto Exec
Parameters pos windowld p

where windowld specifies the window to be positioned. It is one of:

*	positions current active window
=File Path	positions program run from that "File Path"
Title	positions window with caption "Title"
PartTitle*	positions window with caption starting with "Part Title" (Note asterisk at end)
*PartTitle	positions window with caption end with "Part Title" (Note asterisk at start)
PartTitle	positions window with caption containing "Part Title" (Note asterisks at start and end)
autorun	positions window of last window matched by autorun menu
activebar	positions window of last window referenced by active bar button

The position is specified by p; p is one of

t	top half
b	bottom half
l	left half
r	right half

left top wid hei to position at horizontal position **left**, vertical position **top**, width **wid**, height **hei**

You can omit width and height to use the program defaults.

For example,

Command: *Stiletto Exec
Parameter pos autorun 30 100 200 200
used in autorun menu to set position of matched window to 30 100 and size to 200 by 200.

Command: *Stiletto Exec
Parameter pos *Exploring* b
positions an open explorer window in the bottom half of the screen.

To launch a command and set its position, use *sexec (lower case):

Command: Notepad.exe
Parameters: <*wait !<*sexec pos ** 10 50 100 200
waits for Notepad to be launched and then positions the window at vertical position 10, horizontal 50, width 100, height 200.

Automatic Minimization of Inactive Windows

To help keep a neat desktop, you can use **Stiletto** to automatically minimize all windows except those belonging to the active task.

Since there are circumstances where you want to have more than one window open (eg when dragging and dropping), you control auto minimize mode with a command, not a configuration switch. You can turn auto minimize mode on or off by attaching the appropriate command to a button or menu.

The AutoMinimize command can be used to turn auto minimize mode off or on. Use the parameters edit box to set the function of the command:

on	to turn auto minimize on
off	to turn it off
toggle	to reverse the current auto minimize setting

To start **Stiletto** in auto minimize mode, put the auto minimize command on the startup menu.

You can use an active window list or the Active Task Buttons or the Windows 95 task bar to switch among auto-minimized tasks.

It is possible to control which windows are automatically minimized by the Never AutoMin edit box on Window Control dialog. Put captions of windows you do not want minimized, separated by commas, in this edit box. If the entry in the edit box ends with a *, then windows with captions starting with the characters before the * will not be auto-minimized. For example

File Manager, Exploring*
prevents autominizing for both windows with the caption **File Manager** and windows with captions starting with **Exploring**.

It is also possible to turn off the autominimize process when specified windows become active; this is most useful for Help since you usually don't want the program which called help to be auto minimized. Put the captions of such windows in the Ignore For Automin edit box on Window Control dialog. If the entry in the edit box ends with a *, then windows with captions starting with the characters before the * will not cause auto-minimization. As a special case, if the entry consists of a ? alone, then no window with either the word Help or Search in its caption will cause autominimization. For example

Exploring*, ?
means that windows with captions starting with **Exploring** and windows with **Help** in their captions will not cause autominimization.

Automatically Hiding Windows

You can specify that **Stiletto** should automatically hide any windows, should they become visible and inactive.

Put the comma-separated captions of the windows you want to autohide in the Auto Hide edit box on Window Control dialog. If the entry in the edit box ends with a *, then any windows with captions starting with the characters before the * will be hidden when inactive. For example,

HideMe
in this edit box will cause any windows with a caption of Hide Me to be hidden if inactive.

Executing Dos Commands

The built-in command Dos Command can be used to execute a Dos command line command. Type the command you want to execute in the parameters box. **Stiletto** will copy the command to a bat file called stildos.bat and then use the pif file stildos.pif to execute this command. For example:

Command:	Dos Command
Parameters	copy c:\path\file1.txt c:\path2\file3.txt

executes the copy command.

You can enter multiple Dos commands by separating them with a semi-colon (use the DosSepChar internal option to change or remove the Dos separator character).

Stiletto sets the current directory to the **Stiletto** directory before executing Dos commands in order to access the stildos.pif and stildos.bat files.

You can change the pif file Stiletto uses by inserting the pif file name, prefixed by a *, as the first item in the parameters box. You must include the .pif extension and the pif file must reference the bat file stildos.bat. For example:

Command:	Dos Command
Parameter:	*mypif.pif dir *.*>output.txt;print output.txt

uses pif file mypif.pif to execute the dir and print Dos commands.

If you want to use the < command input redirector, you will need to change the CommandSepChar with an internal option.

You can execute a Dos command as part of a set of multiple commands by using the *dos command name.

The stildos.pif file shipped with **Stiletto** is intended for Windows 95; you may need to change it for Windows NT.

Browse and Run Command

The Browse and Run built-in command displays a modified version of the file open common dialog box. You can search for a file or document and select it (eg by double clicking) to have that file executed. As discussed in more detail below, you can use this command to show all files of a certain type in a directory for you to select one to execute. For example, you could have a menu entry called "My Spreadsheets" which executes a Browse and Run command showing all .xls files in a default directory that you specify.

Stiletto will follow any file associations; for example, double clicking on a .txt file will start your default text editor (often NotePad). You set file associations with the Explorer/File Manager File|Associate.

You can enter command parameters in the appropriate edit box in the dialog.

Stiletto remembers the last 25 commands you execute; you can select one of them from the drop down box.

You can set both the default starting directory for the search and the default file types displayed in the file list box using the Work Directory of the Browse and Run command entry controls. Using this feature, you can tailor Browse and Run commands to, for example, go immediately to the directory where you keep your Excel spreadsheets and list all the spreadsheets so you can execute one.

To set the starting directory, type it in the work directory box.

Set the default file extensions as follows: After the default start directory (if used), type a string that consists of a <, followed by the name of the file types, followed by another <, followed by the default extension in the form *.ext. For example:

```
c:\mysheets<Excel Files<*.xls
```

sets the starting directory to c:\mysheets and fills the file list box with all files ending in .xls. You can specify more than one file extension by separating them with semi-colons; for example:

```
c:\<Editable Files<*.txt;*.ini
```

sets the default directory to c:\ and lists all ini and txt files. You can also work with more than one type of file:

```
c:\<Word Files<*.doc<All Files<*.*
```

sets the starting directory to c:\ and fills the file list box with .doc files. You can select All Files when working with the dialog.

Tiny Type and Run Dialog

If you want an easily accessible but unobtrusive command line, use the built-in command Tiny Run Box. It creates a small window consisting of a single drop down edit box. You can type any command into this box and press enter to have the command executed. Or, if you have a three-button mouse, you can execute the command by middle-clicking on the edit box.

You can select the command from the drop down which stores the last 25 commands entered. You can enter dos commands by prefixing the command with *dos (lower case).

After you first start the Tiny Run Box, drag and resize its width to desired dimensions. **Stiletto** will remember the location and width the next time the run box is started.

You can further configure the run box by right-clicking on the edit box (not the caption). You can then:

- specify that the run box should shrink when inactive (see below for details)
- specify that the run box should close when inactive for ten seconds
- specify that the run box should/should not be always on top
- specify that all commands expect those starting with "win " should be prefixed by *dos (useful if you use the run box mainly for dos command line commands)
- or specify that all commands expect those starting with "win " should be prefixed by the ksh shell prefix *dos ksh -L -c; you can change the shell prefix with the shellprefix internal option
- specify whether or not the caption and resizing window frame should be shown
- pick a background color for the window
- browse for a file to execute
- execute the command in the run box
- save the current size to be used as the shrunk size

To keep the run box out of the way when not in use, you can specify that it should shrink when not active. Follow this sequence of steps in the order given:

1. Set the caption on.
2. Move to position so that left of window is at desired location.
3. Resize the width to desired shrunk width.
4. Select "save shrunk width" from configuration menu.
5. Resize to desired large width.
6. Select "shrink if inactive" from configuration menu.
7. Turn caption off, if desired.

If you use the keyboard extensively, you may want to configure a hot key to activate the tiny run box (by setting the hot key command to the Tiny Run Box command).

Put the command on the **Stiletto** start menu if you want the run box to appear when **Stiletto** starts.

If your command file name contains blanks, you must surround it by double quotation marks.

Copy Characters, Files, Date, Time to Clipboard

You can copy characters to the clipboard with the **Stiletto** Exec command:

Command: Stiletto Exec
Parameters clip chars

where chars is a sequence of characters to be copied to the clipboard, enclosed in quotes if it contains blanks.

The following command copies 1234 to the clipboard and then pastes this string into the currently active window:

Command Stiletto Exec
Parameters clip 1234<*wait 0<*send * c-v

Attach this command to a hot key to implement a global shortcut key.

You can also copy files to the clipboard:

Command: Stiletto Exec
Parameters clipfile filepath

copies all the characters in file filepath to the clipboard. The file is limited to a maximum size of 9K.

You can use the same technique illustrated above to create hot keys which paste paragraphs of information, eg if you are a shareware writer sending out many form letters to registered users.

To copy the current time to the clipboard, use **cliptime** in the parameters box. To copy the current date in short format, use **clipdate** or **clipshortdate**. To copy the date in long format, use **cliplongdate**. These follow the time or date format set in your Control Panel | Regional Settings preferences.

Inserting Text in a Log File

You can use the following command to write a line of text to a file:

Command *Stiletto Exec

Parameter file "filepath" datekeyword "text"

writes the **text** to the end of the file given by **filepath**. Enclose the **text** and the **filepath** in quotes if they contain blanks.

The datekeyword is optional; if present, it can be one of these three values:

date writes time and date in Windows short date format at start of line

julian writes time and date as number of seconds since 1970 01 01 at start of line

both writes both julian and Windows short date at start of line

Example:

*Stiletto Exec

file c:\logs\log1.txt both "text after date"

writes "text after date" after both dates to c:\logs\log1.txt

*Stiletto Exec

file "c:\logs path\log1.txt" nodate

writes "nodate" after to c:\logs pth\log1.txt

Using the Clipboard Contents as the Command Parameter

You can use the contents of the clipboard in the command parameter field as follows:

Use the option on the Advanced configuration dialog to set the clipboard character to c, where c is any non-alphanumeric character. Then put the character c in the parameters edit box of the command entry controls.

Example:

```
ClipBoardChar=~
```

Then attach the following to a button:

```
Command:    c:\yourpath\notepad.exe  
Parameter:  ~
```

launches Notepad to edit the file name contained as text on the clipboard.

```
Command:    c:\yourpath\netscape.exe  
Parameter:  ~
```

launches Netscape.exe to view the URL stored as text on the clipboard.

Changing Screen Display Resolution

You can change the display resolution, color depth, and refresh frequency (NT only) with the built-in Change Display Res command.

If you use this command with nothing in the parameters edit box, **Stiletto** will present a menu of valid screen resolutions to choose from. Select one to change and save the new setting in the registry. If you change the color depth or refresh frequency, you will be asked if you want to restart windows for the settings to take effect.

To set a resolution without the menu, specify:

Command: Change Display Res
Parameters: x1 y1

where x1 gives the new horizontal pixels and y1 gives the new vertical pixels. For example, to change to 1024 x 768:

Command: Change Display Res
Parameters: 1024 768

You can alternate between two settings by the following command format:

Command: Change Display Res
Parameters: x1 y1 x2 y2

When this command is executed, the display resolution is set to x1 x y1 unless it is already that value; in this case it is set to x2 x y2.

Normally, the new settings are saved in the Registry; if you do not want this to happen put the word **nosave** after the settings in the parameter field.

System Resources Window

By executing the Show System Resources command, you create a small window which shows GDI/user, memory, status, and free disk space and which can be updated periodically by **Stiletto**.

After you first start the Show System Resources, drag the window to desired position and resize it to desired dimensions. **Stiletto** will remember the location and width the next time the system resources command is started.

If you want **Stiletto** to periodically refresh the values, use the command parameters box to specify a number between 1 and 30; **Stiletto** will continually update the display after that number of seconds.

Alternatively, you can specify that **Stiletto** automatically close the resource windows after n seconds by putting **-n** in the command parameters box; eg put -3 to close after 3 seconds.

You can further configure the window by right-clicking on the edit box (not the caption). You can then:

- specify that the window should/should not be always on top
- specify whether or not the caption should be shown
- specify which of GDI/User, memory, battery, and free disk should be shown
- specify whether the resource text would appear on separate lines or on one line

Put the command on the **Stiletto** start menu if you want the resources window to appear when **Stiletto** starts.

Show Size and Position of Windows as These Are Changed

You can have **Stiletto** display a small window with the mouse screen position and the size and position of the window under the mouse. This display can be activated manually or it can be automatically shown whenever you move or size a window.

To manually show the information window, execute this command:

Command: *Stiletto Exec

Parameter: movesize show

The window will be displayed until you execute the command again (ie to stop display, execute the command again).

To show the information automatically whenever a window is moved or sized, use the check box on the Special GUI dialog. Or, you can dynamically start the automatic display by executing the following command:

Command: *Stiletto Exec

Parameter: movesize yes

The information window has four lines of text:

mouse screen coordinates, both Absolute (point 0,0 is top-left of screen) and Relative (point 0,0 is top-left of window under mouse)

window coordinates: (left,top) - (right, bottom) of window under mouse

total window size: width x height

client window size and aspect ratio

The client window excludes the border, caption, menu bar, tool bars, and status bar. The aspect ratio is the width of the client window divided by its height.

If you run more than one bar, you must execute this command or select this option from the first bar started.

The display window uses the same colors and font as the flyover (tool tip) window.

For the automatic display, to have the information in the display updated dynamically as you move or size a window, you must have the Windows option "Show Window Contents While Dragging" activated; this option is available from MS Plus!.

You can stop automatic display of size and position information with:

Command: *Stiletto Exec
Parameter: movesize no

You can toggle (reverse) the automatic display of size and position information with

Command: *Stiletto Exec
Parameter: movesize toggle

Saving and Restoring Desktop Icon Positions

For NT 4/Win 95, use the Built-in commands *Save Desktop Icon Positions and *Restore Icon Desktop Positions to save/restore the relative positions of desktop icons and optionally align the icons to a grid. Assign the commands to a button or menu, and execute them to save/restore your desktop icons positions. Or you can save and restore icon positions directly using buttons on the Info configuration dialog.

Normally, you'd only execute the save command after changing or adding a desktop icon. You can put the command in your start-up menu to automatically execute it each time **Stiletto** starts if you frequently change your desktop icons.

Positons are stored as numbers which are independent of screen resolution. If you save positions under one resolution and restore under another, the relative positions of icons on your physical screen will not change.

You can align icons to a grid before saving by putting **grid n1 n2** in the parameters box of the *Save Desktop Icon command, where n1 is horizontal grid spacing and n2 is vertical grid spacing. The top left corners of icons are moved to the nearest grid point. For example:

Command: *Save Desktop Icon Positions
Parameter grid 30 20

aligns icons so that horizontal pixel position is a multiple of 30 and vertical is a multiple of 20.

Features of NT and Win95 Versions of Stiletto

Stiletto 98 is a 32 bit implementation: there is one version of the program which runs in all 32 bit versions of Windows.

The following restrictions apply to the Windows NT version of **Stiletto**, resulting from limitations in Windows NT as compared to Windows 95:

Windows GDI and user free resources are always 99

Exit to DOS mode and restart is not available

Stiletto does not handle special font/color settings for console apps

The stildos.pif file used with Dos Commands needs to be changed to suit your system

The *Menu of Window under Mouse command is not supported in Windows NT.

Windows NT is capable of running 16 bit programs in separate Virtual Dos Machines (VDMs). To do this in **Stiletto**, set up the command and parameters as follows:

Command: cmd

Parameter : /c start /separate c:/yourpath/win16.exe commandargs

If you experience problems running 16 bit windows applications under NT, shutdown **Stiletto**, edit the Stiletto.ini file, and insert

NTEXESearch2=10

under

[Task Menu Info].

However, certain features of **Stiletto** (eg checking for active task before launching) will not work for 16 bit Windows programs in this case.

Advanced Configuration Dialog

Using the info configuration dialog, you can access the Advanced configuration dialog which lets you set many less-used **Stiletto** options:

App Bar	If checked, you must shut down and restart Stiletto for changes to the App Bar screen space option on the Special GUI dialog. This will speed other configuration changes.
Animate bar	Check to animate bar disappearance; gray to add sound.
Center configuration	Check to center configuration tabbed dialog.
Show drive letter	When display free disk space on button labels, check this to show drive letter too.
Keep hidden	Invisible bars are normally shown when a hot key is used; check this option to keep them hidden. This may interfere with mnemonic processing on menus.
Refuse focus	If checked, Stiletto will not take the keyboard focus or become the foreground window when another window is closed.
Include desk icons	If checked, desktop icons will be included on the Start Menu shown by pressing the Capture button on command configuration controls.
Vertical text	Displays text vertically on buttons; mainly used for vertical bars.
Rotate text	Displays text rotated on buttons; mainly used for vertical bars.
Drag left edge	If checked, dragging bar near upper left edge will move bar.
Drag right edge	If checked, dragging bar near lower right edge show more buttons.
Sort active buttons	If checked, active bar buttons are sorted by window number instead of window caption text so that changes in the caption do not affect the sort order
Show all windows	If checked, all windows for a task are shown whenever any window for that task is activated.
Parameter dialog	Check to center parameter entry dialog displayed by ?; gray check to move dialog to mouse cursor

Caption dialog	Check to include dialog boxes when bar is positioned in the caption
Auto redisplay	If checked, Stiletto will automatically refresh its display each time you change a your Windows configuration
Keep hidden	Avoid showing Stiletto or bring it to top when displaying hot key menus; this may cause problems with mnemonic keys on the menu
Start and end char	Specify characters to replace {} for specifying special keys when sending keystrokes.
Command separator	Use this character to separate multiple commands; leave blank for none.
Script var character	Use this character to insert script variables a - z.
Clipboard character	Use this character to insert clipboard contents in parameters of commands.
Prompt character	Use this character to prompt for command parameters.
Max menu row	Sets maximum number of rows in *Folder Contents Menu
Button held down	Button configuration dialog shown after button held down for this number of milliseconds; set to a large number to disable.
Marker size	Set size of marker strip for hidden bars.
Explorer windows	Sets number of windows shown on Explorer Windows tracking menu.
Scroll interval	Sets time in milliseconds between scroll steps when automatic mouse scroll is activated..
Hover	Sets time that mouse is over entry in command list in Menu, Alarm, and Keys before long commands are shown in flyover help window.
Send keys delay	Set delay in milliseconds for first key and subsequent keys
Visible	Keeps bar invisible unless specified window(s) are active. Put captions of window in edit box, separated by commas. Use *xxx for captions ending in xxx, yyy* for captions starting with yyy, and =path for all windows started from path.

Internal Configuration Options

Stiletto has a set of internal options to make technical configuration changes. To access these, shut down any running **Stiletto** bars, and edit the **Stiletto** configuration (ini) file to insert one of the following options in the [General] section.

Active Update: To have text on active bar buttons updated when the caption changes, put `ActiveUpdate=1` under [General]. May cause blank buttons to appear on the task bar after screen saver runs.

Autorunmaxwait: For automatic run menus, specify maximum time in milliseconds **Stiletto** waits for new windows to become visible after they are first created by putting `AutoRunMaxWait=n` under [General]

CenterAll: **Stiletto** normally positions the tabbed configuration and browse dialogs near the bar; if you prefer them to be centered on your screen, specify `CenterAll=1` in the **Stiletto** ini file [General] section. Also centers Add Reminder Msg.

CheckClose: **Stiletto** normally verifies whether you want to save any changes if you use the X box to close a dialog; to avoid this check `CheckClose=0` in the **Stiletto** ini file [General] section.

DSTCorrection: Put `DSTCorrection=xxxx` in the [General] section of the **Stiletto** ini file to set the Daylight Savings Time correction to xxxx seconds (default is 3600); only used when Windows indicates DST is in effect.

Fast Menu Appearance on Click: When you click on a button to display a menu, **Stiletto** will display the menu on the click: you can then roll to the selected item and release to select an item or release and click again on the desired menu item. If you prefer the second method, you have to be careful not to move the mouse before releasing (as this is taken as a no selection under the first method). If you always use the second method of accessing menus, you can use `FastMenu=0` in the [General] section of the ini file to turn off the first method so you don't have to be careful about moving the mouse.

HotKeyAllUpWait **Stiletto** will wait for up to 1500 milliseconds for Ctrl, Alt, and Shift to be up before sending keys; you can change the length of the maximum wait by specifying `HotKeyAllUpWait=n` under [General] where n is the new maximum wait in milliseconds (may be zero).

InfoResources: To show GDI/User resources on the info tab, put `InfoResources =1` the ini file [General] section.

Lock configuration: To disable configuration commands, put `Lock=1` the ini file [General] section.

MaxMenuSz **Stiletto** limits menus to 200 entries. You can increase this maximum to up to 1000 by putting `MaxMenuSize=n` in the ini file [General] section, where n is new maximum.

MinimizeMemory For NT, **Stiletto** tries to minimize its memory usage. This may slow response time; to disable the feature put `MinimizeMemory=0` ini file [General] section.

Menu Bias To set the horizontal left-click/right click position of menus, use `LeftMenuBias=n` or `RightMenuBias=n` in the ini file [General] section.

Owned windows in active task lists: Starting with version 1.9q, **Stiletto** no longer displays "owned" top-level windows in active task lists. This cuts down the size of such lists. But if you prefer to see these windows, specify `ShowOwned=1` in the **Stiletto** ini file [General] section.

PressDelay: Specify `PressDelay=n` to delay n milliseconds before pressing the default button on a dialog; use 0 for no delay.

RightContext: Specify `RightContext=0` under [General] to stop[display of right context menu for *Folder Contents Menu.

Scrolling Reset: For middle mouse button scrolling, you can indicate that the mouse cursor should be returned to its starting position by setting ScrollReset= 1 in the Stiletto ini file [General] section

Scrolling Double: For middle mouse button scrolling, you can indicate that double clicking the mouse should scroll to the end by setting ScrollDouble= 1 in the Stiletto ini file [General] section

Separator character for multiple Dos commands: You can change or eliminate the character used to separate Dos commands to the character "c" by specifying DosSepChar=c in the **Stiletto** ini file [General] section, where c is the new separator character, which cannot be an alphanumeric, comma, or space. To eliminate command separation, specify DosSepChar=0.

ShellPrefix: You can change the prefix used in the tiny type and run box to **text** by putting Shellprefix=text under [General] in the **Stiletto** ini file.

StartMenuDelay: Put StartMenyDelay=n in the [General] section of the **Stiletto** ini file to set the delay for moving the Win 95/NT 4 start menu to n milliseconds.

Showtoolbar: **Stiletto** normally does not show win95/NT4 toolbar-style windows on the active task buttons; if you would like to see them, put ShowToolbar=1 under [General] in the **Stiletto** ini file.

Suspended flyover help: After you launch a command, flyover help is normally suspended until you move the mouse off the **Stiletto** button bar. This is to prevent problems with full screen **Dos** commands: if flyover help is not suspended, flyover help will appear unnecessarily after you activate a full screen **Dos** command and minimize the **Dos** window. But, if you never use full screen **Dos** and want flyover help not to be suspended, you can put FlyoverSuspend=0 in the **Stiletto** ini file [General] section.

System folder location: Put SystemFolderDir=c:\yourpath to force Stiletto to always look in c:\yourpath for special folders for *Folder Contents Menu

Transparent button bar: You can ask for a transparent buttons bar by putting transparent=1 in the **Stiletto** ini file [General] section and making sure that Use Windows Button Colors in the Bar dialog is unchecked. This may cause some strange (visual only) side effects, especially with multiple **Stiletto** instances, and may lead to a **Stiletto** GPF when used with some full screen Dos programs, so it is being offered only as a "use at your own risk" internal option. An bar with buttons cannot be made transparent.

TickTimer: Starting with version 1.9s, timers are normally updated by computing the difference between the current clock and the clock setting when the timer was last calculated. This method keeps timers accurate even if another application monopolizes the CPU for an extended period. However, it does mean that timers must be manually reset if you change the system time. An alternative approach which is insensitive to clock changes is to update the timer based on assuming that timer notifications will not be lost due to other applications. You can ask for this approach by putting TickTimer=1 in the ini file [Alarm Info] section.

UChange:: Due to an internal design error, Stiletto will replace ü with a comma in command lines. To fix this, put UChange=1 in the ini file [General] section. This is default in all versions after 97g.

UseFolderIcon To force **Stiletto** to use an internal icon to represent folders in menus, put UseFolderIcon=1 in the ini file [General] section.

UseUserProfile: To force Stiletto under NT4 to look in the the environment variable USERPROFILE for program and other special folders, put UseUserProfile=1 in the ini file [General] section. UseAdmin takes precedence over UseUserProfile, so you must omit UseAdmin if you want to use UseUserProfile.

Frequently Asked Questions

Where is my configuration stored? How do I back it up? How do I keep my configuration when upgrading?

The configuration is stored in file stiletto.ini. Take a backup copy of this file to save your configuration. Installation zips of **Stiletto** do **not** include a stiletto.ini file so they do not overwrite any existing configuration when installed: to upgrade, copy all the files in your current **Stiletto** directory to a backup, shutdown all running bars, and unzip the **Stiletto** stilins zip file into your main **Stiletto** directory.

What are all the files in the Stiletto folder? Which can I delete?

See filelist.txt in the folder for an explanation. In addition, **Stiletto** creates .bki backup files and .udo undo files. To save space, once you register you can delete files with the names of the form reg*.*.

Where is my registration code stored? Do I have to re-enter it for each upgrade?

The registration code is stored in the registry. **Stiletto** automatically reads it from there. There is no need to re-enter when upgrading.

How do I resize all of the buttons without having to hit each button configuration? Why can't I change the height of the button bar?

Use the bar tab of the configuration dialog to set the base size for all buttons and to set the bar height (width for vertical bars).

For win95 and NT4, what is the best way to show a menu by right-clicking the desktop?

If you use the menu setup tab to set a desktop menu, **Stiletto** will attempt to show both the **Stiletto** menu you set and the Windows desktop or desktop icon context menu. This may not always work well; eg in NT 4, one of the menus may not close properly.

Instead of using the menu setup tab, create a right-desk hot key which executes a *Show Menu for your desktop menu. Include the following command in your menu:

Menu Item Name:	Context
Command	*Stiletto Exec
Parameter	mouse right

If you click you mouse anywhere on the desktop, only the **Stiletto** menu will be shown. To access the Windows context menu for the item under the mouse, select the Context command.

You may also want to experiment with right-hold hot keys, chord left+right hot keys, and middle mouse hot keys.

How do I create a bar in the caption so it looks like the icons of the bar are part of the caption?

On the bar tab of the configuration dialog, check (not gray) "Use flat look for bar". Select one of the three caption positions: L. Cap, M. Cap., or R. Cap. Use the Bkg Color, Border, Shadow, and Hilight buttons to set the corresponding colors to the color of the active title bar.

How do I use middle mouse button to send left double click? What else can I do with the middle mouse button?

The middle mouse button can provide many functions with **Stiletto**:

You can attach hot keys to it: for example, a mouse-all hot key and a mouse-hold hot key. Use these hot keys for direct commands, like sending a left double click with **Stiletto Exec mouse**, or for menus, such as menu of send key commands to send common shortcut keys or simulate picking menu entries.

In addition to the hot key, you can also use the middle mouse for either scrolling or for moving a window by setting the option on the Special GUI config tab.

How can I activate programs which are not files, such as printers or control panel applets?

Use Explorer to create shortcuts to these special programs and then run the shortcuts from **Stiletto**. You can create a folder of shortcuts to all your printers or other special programs, and display them all as a menu using Folder Contents Menu.

How can I start many Explorer windows at the same time? How can I set the folder that Explorer starts with?

To start many windows from Explorer (or any other program), you must uncheck "Switch to If Active" at the bottom of the command entry controls for each button or menu item which is to start the command. To learn how to use Explorer to start at any folder, see the file tips.txt that Microsoft includes in your Windows directory. Put the command parameters described there into the **Stiletto** Parameters edit box.

What do the "C" and "P" buttons on the button config screen do?

You can find out about any button or other control on **Stiletto** configuration dialogs by clicking on the ? in the upper left corner of the caption, moving your mouse to the control you are interested in, and then clicking on it. In particular, the C and P buttons can be used to copy and paste commands from command entry point to another- eg a button to an alarm.

If I configure a button bar that's NOT on the caption bar or task bar, how do I use it without minimizing everything on the desktop?

You can choose any of these options, or combine them:

- Use Bar tab on configuration dialog and check "Always on top" for **Stiletto** bar

- Select "Show Stiletto when mouse bumps screen edge" option from Bar tab; bumping screen edge will show bar

- Assign "Show/Move Stiletto" command to hot key/mouse movement: activating the key will show Stiletto

I have a caption bar, and I don't want it to go away when there is no active window, but I don't like the way it snaps to the upper right side of the screen. How do I change this behavior?

Drag the desktop bar to the exact position the bar has when it is at the left of a maximized window's caption bar. Clicking on the task bar then doesn't move it.

How much memory does Stiletto use?

The **Stiletto** program requires about 400K. Each bar including the first one will use about an additional 300K. Icon caches require further memory.

Modern operating systems like NT and Win95 (!) do not keep all components of **Stiletto** in memory if other programs you are running need memory. Pieces of **Stiletto** that are rarely used, for example the configuration dialog or members of the icon cache, will be swapped out to disk automatically. (Hence there would be little memory saved by providing a separate program for **Stiletto** configuration.)