

[Introduction](#)

[Conventions](#)

[The NoteTab Family](#)

[NoteTab Pro](#)

[NoteTab Std](#)

[Freeware and Trial Versions](#)

[Comparison Chart](#)

[Key Features](#)

[Time-saving Features](#)

[Frequently Asked Questions](#)

[Tips and How to...](#)

[History of NoteTab](#)

Introduction

NoteTab™ is an award-winning text and HTML editor for Windows 95, 98, ME, NT4, 2000, and higher. It is user friendly and also feature rich with many innovative productivity tools. Its features appeal to users ranging from novice to very demanding. Whether you create Web pages, write source code, send e-mail, take notes, analyze text, read files, or do anything related to text, you will certainly find NoteTab a worthy tool and a great time saver.

NoteTab Pro has won many prestigious awards, including the 1999 *Shareware Industry Awards* ("Best Web Enhancement" category), the 1998 and 1999 *People's Choice Award*, and the 1998 *PC Magazine Shareware Awards* (Applications category). NoteTab Light won the 2000 *Shareware Industry Awards* ("Best Application" category).

If you are not familiar with technical computer terms, open the outline file Glossary.otl in NoteTab. You will find all the important words defined in it.

Innovative Features

Unlike MS-Notepad that comes with Windows, NoteTab lets you edit documents of virtually any size. Furthermore, you can open as many documents as your system's memory will allow, each with its own button in the tab bar.

NoteTab is not your typical text editor and is currently unrivaled by the range of useful features it offers. You can, for example, open links in your default browser, calculate mathematical expressions, create outline-type documents, convert text files to Web pages, strip tags from HTML documents, send documents as e-mail, capture text copied to the Clipboard, view detailed text statistics, etc.

One of the most original features, which NoteTab was the first to implement, is the Editor Clipbook. This is a flexible tool for handling text clips, which can be anything from a single character to a large "boilerplate" chunk of text. Clips are stored in libraries with a header for identification, and retrieved by selecting the appropriate header from the list or by typing the first matching characters of the header into your document and hitting a function key. When auto-replace mode is enabled, the Clipbook tool can provide functionality similar to WinWord's AutoCorrect feature.

By default, pasting a Clipbook item will overwrite any selected text in your document. But by customizing your clip text with simple codes you can do anything from wrapping the clip around a selection (great with HTML tags), to launching other programs, to loading files, to sending the document as e-mail, to calculating the result of functions with multiple variables, etc.

Starting with version 4.5, the Clipbook is programmable and supports features such as variables, functions, commands, conditional statements, events, etc.! Furthermore, commands also provide seamless support for Perl and Gawk interpreters; scripts can be stored in Clips and applied directly to highlighted text or the whole document.

Of course, NoteTab also offers the features you expect to find in a good text editor such as Find/Replace in all open documents or disk files, read and write files in DOS/UNIX/EBCDIC/Mac format, sort/join/split/indent lines, change character case and text alignment, convert between ASCII and ANSI, spell checker, thesaurus, automatic indentation, etc.

NoteTab is very customizable. You have a selection of more than 90 commands to choose from for the toolbar and the main shortcut menu. You can even create your own toolbars from scratch, with your own icons and commands. The editor's behavior can be fine tuned to suit your needs, thanks to many options.

NoteTab can even replace Notepad by simply selecting a menu command. All these great features have earned NoteTab many awards and favorable reviews in the press.

NoteTab™ is a trademark of Eric G.V. Fookes, Switzerland

Windows is a trademark of Microsoft Corporation registered in the U.S. and other countries.
All other trademarks and service marks are the property of their respective owners.

Conventions

The name **NoteTab** is used to refer to either **NoteTab Std** or **NoteTab Pro**. The **Editor Clipbook** tool is often referred to as just **Clipbook**.

The pipe symbol "|" is used to separate imbedded menu names or dialog box elements. For example:

"Modify | Change Case | Capitalize"

refers to the **Capitalize** command in the **Change Case** submenu situated under the **Modify** menu.

"Options | Clipbook | Indent on Paste"

refers to the **Indent on Paste** setting on the **Clipbook** tab in the **Options** dialog box.

If you are not familiar with technical computer terms, open the outline file **Glossary.otl** in **NoteTab**. You will find all the important words defined in it.

The NoteTab Family

The NoteTab™ family of editors is composed of three variants (see the [comparison chart](#) for more details):

1. NoteTab Pro (Commercial)

- Very fast
- Multi-level undo/redo
- Bookmarks
- Line-count ruler
- URL and HTML-tag highlighting
- Show/Hide nonprinting characters
- Spell checker and thesaurus
- Full support for NoteTab-Outline documents
- Search/replace text in files on disk
- Options for word wrap at column, tab types, etc.
- Read and write EBCDIC files
- Supports fixed-pitch OEM fonts
- Can edit files up to 2 GB each
- Trial version from: <http://www.notetab.com>
- Cost: \$19.95 USD (Minor updates are free)

2. NoteTab Standard (Commercial)

- Spell checker and thesaurus
- Full support for NoteTab-Outline documents
- Search/replace text in files on disk
- Can display variable- and fixed-width fonts
- Cannot display fonts with OEM character sets (Terminal, MS Line Draw,...)
- Read and write EBCDIC files
- Can edit files up to 2 GB or as large as memory permits
- Cost: only \$9.95 USD (Minor updates are free)

3. NoteTab Light (Freeware)

- Same features as NoteTab Standard except those listed below:
- No spell checker or thesaurus
- Can read NoteTab Outline documents but cannot create or edit them
- No Clipbars (user-made toolbars)
- No "Reformat Lines" or "Justify" command
- Can search for text in disk files but cannot perform replace operations
- Can read EBCDIC files but cannot create or modify them
- Available from: <http://www.notetab.com>
- Cost: Free!

Fookes Software also offers several [other products](#) you may find interesting. Visit our Web site at:

<http://www.fookes.com/>

NoteTab™ is a trademark of Eric G.V. Fookes, Switzerland

NoteTab Pro

NoteTab Pro™ is one of the commercial versions of the NoteTab family of editors. It uses a more sophisticated input control that provides many options to configure its behavior (wrap to column, scroll past end of line, real/fixed/smart tabs, etc.). Furthermore, it offers multiple-level undo/redo, bookmarks, URL and HTML-tag highlighting, a spell checker, and a thesaurus. Note that, unlike NoteTab Standard, NoteTab Pro only supports fixed-width fonts. NoteTab Pro is particularly well suited as an HTML editor.

NoteTab Pro costs only \$19.95 USD. See the topic [Ordering Information](#) to find out how easy it is to purchase NoteTab Pro.

The main advantages of NoteTab Pro over NoteTab Std are:

- Speed
- Multiple-level undo/redo
- URL and HTML-tag highlighting
- Bookmarks
- Wrap to column
- Show/Hide nonprinting characters
- Choice between real/fixed/smart tabs
- Line-count ruler

See the [comparison chart](#) for the differences between NoteTab Pro and [NoteTab Std](#).

The NoteTab Pro trial version, which is freely available for downloading, has all the features of the commercial version except for the spell checker and the thesaurus, which are not included. The trial version is time limited and will stop working about one month after the moment it was first installed. The program can be uninstalled after the trial period or you can still use it to connect to the order form on the Web. If you purchase the commercial version, you can install it over the trial version and preserve all your settings.

NoteTab Pro™ is a trademark of Eric G.V. Fookes, Switzerland

NoteTab Std

NoteTab Standard™ (or NoteTab Std™) is similar to NoteTab Light™ (the Freeware version) but adds a spell checker, thesaurus, full support for NoteTab's outline documents, and a fully enabled command for search/replace text criteria in disk files. NoteTab Std is an excellent replacement for MS Notepad.

NoteTab Std costs only \$9.95 USD!!! See the topic [Ordering Information](#) to find out how you can purchase NoteTab Std.

The main advantages of NoteTab Standard over NoteTab Pro are:

- Can use fixed- and variable-width fonts
- Better support for non-Roman alphabets (double-byte characters; e.g. Chinese, Hebrew, etc.)

See the [comparison chart](#) for the differences between [NoteTab Pro](#) and NoteTab Std.

NoteTab Light is the freeware version. Unlike the commercial version, it does not have a spell checker or thesaurus. Furthermore it only lets you read [outline documents](#) and EBCDIC files, the [Search Disk](#) tool does not provide the Replace option, and [Clipbars](#) are not supported. When you install NoteTab Light for the first time, it will ask you if you want to try some of the commercial features. If you answer yes, you will have access to all the outline editing commands and the Replace option for the Search Disk tool will be available. If you answer no, you can always change your mind later on by choosing the View | Commercial Features menu command. Once enabled, these extra features are available for one month. After that period, NoteTab Light simply reverts to its original state. If you purchase NoteTab Std, you can install it over the freeware version and preserve all your settings.

NoteTab Std™ and NoteTab Light™ is a trademark of Eric G.V. Fookes, Switzerland

Freeware and Trial Versions

NoteTab Light

NoteTab Light is the freeware version. Unlike the commercial version, it does not have a spell checker or thesaurus. The new Clipbar feature (user-made toolbars) is not available either. Furthermore it only lets you read outline documents and EBCDIC files, and the Search Disk tool does not provide the Replace feature. When you install NoteTab Light for the first time, it will ask you if you want to try some of the commercial features. If you answer yes, you will have access to all the outline editing commands and the Replace option for the Search Disk tool will be available. If you answer no, you can always change your mind later on by choosing the View | Commercial Features menu command. Once enabled, these extra features are available for one month. After that period, NoteTab Light simply reverts to its original state. If you purchase the commercial version, you can install it over the freeware version and preserve all your settings.

NoteTab Pro (Trial)

The NoteTab Pro trial version, which is freely available for downloading, has all the features of the commercial version except for the spell checker and the thesaurus, which are not included. The trial version is time limited and will stop working about one month after the moment it was first installed. The program can be uninstalled after the trial period or you can still use it to connect to the order form on the Web. If you purchase the commercial version, you can install it over the trial version and preserve all your settings.

Comparison Chart

NoteTab Pro, NoteTab Std, and NoteTab Light features compared:

	<i>NoteTab Pro</i>	<i>NoteTab Std</i>	<i>NoteTab Light</i>
General speed	Very fast	Can be slow with large files	Can be slow with large files
"Replace All" speed	Very fast	Average	Average
Show nonprinting characters	Yes	No	No
Multi-level undo/redo	Yes	No	No
Bookmarks	Yes	No	No
URL and HTML-highlighting	Yes	No	No
Use OEM fonts	Yes	No	No
Use variable-width fonts	No	Yes	Yes
Spell checker + thesaurus	Yes	Yes	No
Text drag-and-drop editing	Yes	Yes	Yes
Disk Search & Replace	Yes	Yes	Only during trial mode
Clipbars	Yes	Yes	Only during trial mode
Read & write EBCDIC	Yes	Yes	Read (Write in trial mode)
Read & write outline files	Yes	Yes	Read (Write in trial mode)
Multiple tabulator settings	No	Yes	Yes
Can wrap text to column	Yes	Not in real time	Not in real time
Show line numbers	Yes	No	No
Maximum file size	2 GB	2 GB	2 GB
Maximum paragraph size	32 KB	2 GB	2 GB
Cost (in US dollars)	\$19.95	\$9.95	Free

Key Features

- Open and edit many documents at once, even huge files (only limited by system resources).
- *NoteTab Pro*: multi-level undo/redo.
- Text drag-and-drop editing.
- AutoCorrect/Auto-replace mode similar to Microsoft® WinWord.
- Supports MS-Office sound events.
- Fully configurable toolbar with flat or standard style buttons (more than 90 commands available).
- *Commercial versions of NoteTab*: Clipbars let you create your own custom toolbars.
- Fully configurable shortcut menu.
- Programmable. Add your own custom features using an easy script language.
- Web enabled: can open links and HTML documents in browser.
- Supports third-party Web tools: HTML Tidy, TopStyle, and CSE HTML Validator.
- Powerful Clipbook tool makes it easy to memorize and insert any kind of text in your document. Includes libraries for HTML tags, acronyms, smilies, etc. Can also be used for "shorthand" glossaries or to launch other applications.
- *Commercial versions of NoteTab*: Clipbars, to add your own toolbars to NoteTab.
- Innovative In-Context Clipbook reduces mouse movement and speeds up work.
- Integrates support for Perl and Gawk interpreters; scripts can be stored in NoteTab and applied directly to highlighted text or whole document.
- Can calculate (type a mathematical expression and NoteTab will calculate the result).
- Each document can have different settings such as font, tab widths, word wrap.
- Supports outline-type documents that help you organize your notes or any kind of information.
- Supports document templates.
- Can search/replace text in all open documents or disk files; accepts search tokens and regular expressions.
- Functions to convert text files to Web documents, strip HTML tags, sort lines, change case, process text in blocks, etc.
- *NoteTab Pro*: can highlight URLs, HTML tags, Clip code, and HyperLinks.
- Drag-and-drop image files from the Quick List tool into an HTML document creates image tags with the correct size attributes.
- *Commercial versions of NoteTab*: spell checker and thesaurus (requires the appropriate dictionary files).
- *NoteTab Pro*: insert up to 10 bookmarks per document.
- Automatic scrolling for hands-free reading of documents.
- Options to set margins, page numbering, headers and footers for printing jobs.
- Reformat text, including left and right justification.
- Text statistics (counts number of characters, words, and frequency of each word).
- Drag-and-drop files from File Manager or Explorer.
- Accepts multiple file names, including wildcards, passed as command-line parameters to the program.
- Quickly open files or directories from the Favorites list.
- Can capture text copied to the Clipboard from any application.
- Can read and write files in DOS ASCII, UNIX, EBCDIC, and Mac formats.
- Year 2000 compliant.
- Covers all essential features available in Notepad (including LOG).
- And lots more...

Time-saving Features

NoteTab offers many time-saving features. The most important are:

- Editor Clipbook: lets you store and paste any kind of text.
- Auto-replace mode.
- Favorites: quickly open your favorite files and directories.
- Templates: save time in creating many common types of documents.
- Paste Board: copies all text received by the Clipboard into a chosen document.
- Customizable shortcut menu: choose the commands you find most useful for the main shortcut menu.
- Clipbars: user-created toolbars with buttons that run Clips.
- Third-party products: useful programs that integrate with NoteTab.

Frequently Asked Questions

What does the NoteTab icon represent?

Fookes Software, the company that produces NoteTab, is based in Geneva, Switzerland. The white cross on the red background represents the Swiss flag.

How do I get rid of the splash screen?

You can disable the splash screen by unchecking the "Show Splash Screen" option in the Options dialog box.

Can I replace Notepad with this editor?

Yes! If NoteTab is correctly installed, under Windows 95/98/NT4 you will find a command called "Replace MS Notepad" under the Help menu. When you choose that command, NoteTab will first rename Notepad by adding the extension ".MS" to the program file. Then, it will copy a special little program called NOTEPAD.EXE to the Windows directory. Whenever this program is executed, it launches NoteTab instead of Notepad. If you want to restore MS Notepad after this operation, just use the command "Restore MS Notepad", which is also available under the Help menu.

Note that this feature is not available under Windows 2000 and ME, because it does not allow substituting MS Notepad with another program.

How can I get a print preview?

NoteTab currently does not integrate a print-preview function. However, there are two excellent utilities we can recommend for this task. Both of them provide print preview and, if you want, let you print multiple pages per sheet so you can save a lot of paper. FinePrint™, from *Single Track Software*, is the easiest to use and also the most flexible. This Shareware program can be downloaded from the following Web site:

<http://www.singletrack.com>

TxtPrint, from Örjan Råberg, is Freeware. Unlike FinePrint, it only handles text files. The easiest way to send NoteTab documents to it for printing is by using the TxtPrint library available from the Clipbook Repository at the NoteTab Web site. TxtPrint can be downloaded from the following link (download size 74 KB):

<http://www.notetab.com/ftp/txtprinten10.zip>

When I save a document as FILENAME.EXT, it becomes FILENAME.EXT.TXT

All programs using the standard Windows file dialog boxes do this. In fact, the default extension is added by Windows, not NoteTab. If the extension you are using is listed in the dialog's file-type list, then the default extension is not added. You can add any extension to the list by opening the Options dialog box and then activating the File Filters tab.

Microsoft added this behavior with the introduction of long file names. You can now have file names like the following:

MyFile.1999.txt

The .1999 is not considered as being part of the extension. And if you type

MyFile.1999

in the File dialog box and ".1999" is not listed as a file extension (or associated with another application), then Windows assumes it is part of the file name and name does not have an extension -- hence it adds the default extension.

How do I specify a default directory for opening and saving files?

There is an option in the Options dialog box on the Files tab that lets you choose the default open/save directories.

How do I associate specific extensions with this editor?

There is a tool in the Options dialog box on the Associations tab that makes it really easy to associate file extensions or undo associations. Add extensions to the list to create an association or delete an extension to undo the association.

The link is not opened in my browser when I click on it!

You must hold the Control key down and double-click on the link to open it in your browser. However, if you set the menu option "Tools | Quick Open Links", you only need to double-click on the link without holding the Control key down.

NoteTab does not load the whole file!

The file probably contains one or more null characters. By default, NoteTab treats the first null character it encounters as the end of the text. To change this behavior, you can set the "Filter Binary Codes" option which you will find on the Files tab in the Options dialog box. When this option is set, any null character NoteTab encounters when loading a file is converted to the substitute character (also defined in the Options dialog box). You will be able to load the full content of any file when this feature is enabled.

My Word processor file looks funny in NoteTab and some text is lost!

Files produced by word processors are usually not saved as plain text files. They contain all sorts of control codes that only the word processor that created them (or compatible programs) can understand. Word processors use those special control characters to store text formatting information (bold, italic, line spacing, paragraph alignment, etc.).

The NoteTab editor only understands plain text files (that's why it is so fast compared to a word processor) and will display the special control characters as funny symbols or rectangles on the screen. Some of the control characters may be interpreted by NoteTab as signaling the end of a file. This explains why such files are rarely fully loaded in the editor.

Unless you really know what you are doing, you should not edit or save such files with NoteTab. Doing this may render them unreadable to your word processor!

NoteTab Pro: where have all my fonts gone?

Unlike NoteTab Std, NoteTab Pro cannot use variable-width fonts. It therefore limits your choice of fonts to those that are fixed-width. This design is imposed by the input control used in NoteTab Pro. The benefits are better performance and a choice of commands/configuration options that are not available in NoteTab Std.

How can I set attributes (bold, underline, ...) for individual text elements?

This is not possible with NoteTab (or any "plain text" editor for that matter). This feature would require storing special control characters in the document. Since NoteTab is a "plain text" editor, it only saves documents as text without any control codes. Therefore all formatting information would be lost when you open the document again. If you want to format individual text elements, you will need a more sophisticated editor like a word processor.

Is NoteTab available in other languages than English?

At the present time we do not have plans for it. However, that is not to say that it never will be.

Are there plans to make a Linux or MAC version of NoteTab?

There is a possibility of NoteTab being ported to Linux if Borland-Delphi, a program that NoteTab is written in, adds support for it. There are no plans for Macintosh. Note, however, that you can run NoteTab on Macs using Virtual PC.

Where is the convenient "x" toolbar button for closing documents?

Version 4 uses a new type of toolbar which displays faster and consumes less system resources; this is particularly important as it now supports more than 90 buttons. Unfortunately, the little "x" button was no longer usable as it caused conflicts with the new toolbar. You can, however, add a standard "Close Document" button.

Answers to other questions...

Tips and How to...

In NoteTab, you can drag-and-drop selected text. To move text to another part of your document, first select it, then click on the selection with the left mouse button and drag it to the desired location. If you want to duplicate text rather than move it, follow the same procedure but with the Control key pressed down.

In NoteTab Pro, double clicking on a word with the mouse selects it together with all blank spaces following it. If you only want the word selected, just hold down the Alt key while you double click on the word.

When you have several documents open, you can drag-and-drop page tabs to reorder them any way you like. If the tabs are stacked, you will first have to enable drag-and-drop by double clicking on the tab before moving it.

If you use the system tray and always want the program available, don't exit the editor but minimize it instead. The program will minimize to the system tray and reappear next time you click on its icon.

If you use Explorer and would like to open a selection of documents in NoteTab, make sure you use "Send To" instead of the "Open" command. The "Open" command tries to launch a new instance of NoteTab for each selected file. Due to a feature that causes NoteTab to check for previous instances of the program, only some of the files may be loaded. Furthermore, this process is considerably slower than using the "Send To" command.

NoteTab makes the popular "The HTML Reference Library" (by Stephen Le Hunte) available from the Help menu. If you have HTMLib version 3.0 or older, you can install the package in the same directory as NoteTab. Another method that works with all versions is to create a shortcut* to the HTMLib.exe file in the NoteTab directory. The highly recommended "HTML Reference Library" is available from:

<http://www.htmlib.com/>

(* to create the shortcut: 1. open Explorer and select the HTMLib file; 2. press Ctrl+C to copy it to the Clipboard; 3. select the NoteTab directory and choose the command "paste Shortcut" from the Edit menu. The menu command will be displayed in NoteTab after you restart it.)

The ruler can display measurements in characters (only accurate with fixed-width fonts), inches, and millimeters. Click on the ruler to toggle between the different units. When you place the mouse cursor over the ruler, the measurement at that position is indicated in the hint bar. Note that the ruler measurements do not scroll horizontally with the editor window.

Use the View | Multiple Instances menu command if you want to open multiple copies of the editor in memory.

The Clipbook can be used to launch other programs, open the current document in its associated application, open the current document in the specified program, and open URLs. Clips with this capability have a little exclamation "!" symbol on their left side.

Open a local file by placing the cursor on its name in a document and pressing the F7 key (or through the menu: Tools / Open File at cursor).

The "Go to Line" command is activated when you double-click on the cursor-position pane in the status bar.

NoteTab Pro: You can toggle between insert and overwrite mode by double-clicking on the Insert/Overwrite pane in the status bar.

Double-clicking on the hint line in the status bar selects all the text in the current document.

You can copy the content of the Clipbook and Outline list to the Clipboard (see the list box shortcut menu). In the Hint bar, you can see how many items are contained in the list when you select the "Headers to Clipboard" or "Headings to Clipboard" command.

Pressing the Tab key when you have several lines of text highlighted (starting at the beginning of the first line) will now act as if you have chosen the indent command. Holding the Shift key down will unindent the highlighted lines.

When the Quick List is used to display open documents, double-clicking or pressing Enter on a name in the list will set the focus to the corresponding document.

If you are very familiar with INI files, you can edit NoteTab's own INI file like any other document. To have your changes taken into account when you reopen NoteTab, you must rename the file by adding an exclamation mark in front of the name (e.g. !NoteTab.INI). Next time you run NoteTab, the content of the renamed INI file will be used.

History of NoteTab

NoteTab has been a leader in innovations brought to the field of text editors ever since the first version. Many of these original features have now been copied by competing software. Here is a short outline of the most significant features first introduced by NoteTab:

October 1995:

Introduction of the first version of NoteTab (called Mini NoteTab) for Windows 3.x. It was one of the first text editors (perhaps even the first) to use a tabbed layout instead of the cumbersome MDI interface.

December 1996:

Favorites dialog box was introduced. This feature made it very easy to find files and directories no matter where they are located on the hard disk(s).

January 1997:

The flexible and very popular "Editor Clipbook" tool was created and integrated in NoteTab version 2.0. Although this feature is being increasingly copied, the original Clipbook remains the uncontested leader in ease of use, power, and flexibility.

Web-enabled features: lets you easily open links in your default browser by clicking on them with the mouse. Added a command to preview HTML documents in Web browser.

June 1997:

The convenient In-Context Clipbook was introduced in NoteTab version 3.0.

February 1998:

NoteTab version 4.0 introduces many original features never seen in text editors before. These include:

- Outline documents with hypertext links
- Clipbook Wizards
- Customizable shortcut menu
- Calculates mathematical expressions
- Easiest method ever to replace MS Notepad with alternative editor

September 1998:

NoteTab version 4.5 is released. This version is programmable thanks to an extension of the Clipbook syntax which supports variables, conditional statements, events, more than 90 Clip commands, and more than 80 Clip functions. Furthermore, NoteTab also integrates support for Perl and Gawk interpreters; scripts can be stored in NoteTab and applied directly to highlighted text or the whole document.

March 1999:

NoteTab version 4.6 is released. Many new features have been added since the previous update. Some, like drag-and-drop editing and an option to show nonprinting characters in NoteTab Pro, were requested by users for a long time. Auto-replace mode based on the Clipbook adds yet another significant time saver; this feature is similar to WinWord's AutoCorrect tool and the Code-Completion functionality found in some programming environments. With the addition of EBCDIC support, NoteTab has become a valuable tool for any project involving editing or converting EBCDIC files from MVS and AS/400 systems. As a result, you can now use NoteTab for those huge Y2K projects.

February 2000:

NoteTab version 4.8 is released. The Pro and Std versions let you create your own toolbars, called Clipbars. The Clip language adds over 50 new commands and functions, and for the first time NoteTab Light supports all its features. NoteTab now integrates with HTML Tidy, TopStyle, and CSE HTML Validator. HTML coders can configure NoteTab to render their tags either in upper case, lower case, or in XHTML format.

[Ordering Information](#)

[Order Form](#)

[Site Licenses](#)

[License \(Commercial version\)](#)

[License \(Freeware version\)](#)

[We are an ASP member](#)

Ordering Information

There are several ways you can purchase a registered copy of NoteTab:

1. If you own a credit card, the fastest method is to order on-line through a secure server. It takes just a few minutes to fill in the order form and for your credit card information to be processed before you reach the download link.
2. Order by phone or fax with your credit card, and download from the Internet.
3. If you prefer to pay by check or money order, you can purchase your registered copy from the NoteTab agent in the USA.
4. You can also pay the author directly by sending cash in an envelope or transferring your payment to his postal account in Switzerland.

The following prices are valid until the end of July 2001:

NoteTab Pro v4.83, cost \$19.95 USD *

NoteTab Std v4.83, cost \$9.95 USD *

** There is an extra charge if you want the software on CD-ROM.*

After that time, please check the NoteTab Web site or write to [<sales@notetab.com>](mailto:sales@notetab.com) to find out if the prices have changed. You can also find the latest prices from the following Web page:

<http://www.notetab.com/order.htm>

Site licenses: purchase orders (net 30 days) are accepted from government and accredited educational institutions and major corporations, provided that they are submitted on purchase order forms with a purchase order number. A \$10.00 USD processing fee will be added to the total. Please be sure to **include a contact e-mail address** with your purchase order, and send it to:

Fookes Software
Av. Eugène-Pittard 22 Ter
CH - 1206 Geneva
Switzerland

Fax: +41/22-789 58 44

See the topic [Site Licenses](#) for site license pricing.

Note that customer information is considered confidential and will not be shared or distributed to any third party.

Secure On-line Ordering

The "Place Order" area is secured using state-of-the-art Secure Sockets Layer (SSL) encryption, which means that no one can intercept transmissions to or from the server. All orders are delivered electronically – right to your computer. After the purchase approval process, you will be given a link to download your new software. Are you concerned about downloading your new software? No need to worry! During your download, if you experience problems such as a dropped connection or a system crash, you will be able to re-initiate your download by going to the customer service page. Go to the following page to order

NoteTab online:

<http://www.notetab.com/order.htm>

Note that if you are using NoteTab Light or the trial version of NoteTab Pro, you can also order the software by choosing the "Ordering Information" command from the Help menu.

The following credit cards are accepted: Visa, MasterCard, American Express, Discover, Diner's Club, JCB

You can also order from www.regnow.com or www.ebarn.com if the above link fails.

Order by Phone or Fax

If you prefer, you can order NoteTab from *Universal Commerce* (www.regnow.com) by sending your credit card number by phone or fax (instead of sending it over the Internet). You will need to provide the following information:

- Product you want to order (NoteTab Pro or NoteTab Std)
- Your full name
- Your full address w/ postal code and country (should match the credit card address)
- Your full phone number
- Your e-mail address
- Your credit card information (card type, number, validity)
- Your signature – if you send this information by fax

Then, once your order has been received and the credit card information processed, you will get a receipt and a download link together with registration information by e-mail.

Use any of the following numbers to make your order:

Toll Free Phone	1-877-353-7297
Toll Free Fax	1-888-353-7276
Phone	1-425-392-2294
Fax	1-425-392-0223

Pay by Check or Money Order

Send your check or money order, payable to **Eric G.V. Fookes**, along with the name of the product(s) you are ordering (NoteTab Std or NoteTab Pro), and your e-mail address so the official NoteTab agent in the USA can send you the appropriate download information. Payments must be in US dollars drawn on a US bank, or you can send international postal money orders in US dollars.

Send your check or money order to the following address:

Fookes Software
P.O. Box 998
Milton, FL 32572-0998
USA

E-mail: Jody Adair <jody@notetab.com>

Once your order has been processed, your registration information will be e-mailed to you together with a

download link to the commercial version of NoteTab. You can also order NoteTab on a CD-ROM if you do not have access to the Internet, or prefer to receive your registered software by mail; please see the [Order.txt form](#) for details.

Please do not forget to send your e-mail address! If you cannot accept downloads or e-mail attachments, you will have to adjust the cost of the program to include the CD-ROM and any handling costs.

Pay the Author Directly

If you like, you can also purchase NoteTab directly from the author. The following prices are valid until the end of July 2001:

NoteTab Pro v4.83, cost \$19.95 USD (or CHF 30.-) *

NoteTab Std v4.83, cost \$9.95 USD (or CHF 15.-) *

** Please add \$5.00 USD if you want to pay by check. If you need the software on CD-ROM, please order through our agent in the USA (see above).*

After that time, please check the NoteTab Web site or write to [<sales@notetab.com>](mailto:sales@notetab.com) to find out if the prices have changed. You can also find the latest prices from the following Web page:

<http://www.notetab.com/order.htm>

Send your payment in cash (in US Dollars or Swiss Francs) to the following address:

Fookes Software
Av. Eugène-Pittard 22 Ter
CH - 1206 Geneva
Switzerland

E-mail: [<sales@notetab.com>](mailto:sales@notetab.com)

If you live in Europe, you can also pay directly to the following Swiss postal account: CCP 12-51699-7, Geneva. Please **also send an e-mail to** [<sales@notetab.com>](mailto:sales@notetab.com) to inform us that you have paid this way (often, statements from the post-office or bank contain incomplete information and we cannot fulfill such orders unless we have your e-mail address).

Use registered mail if you feel it is safer; most customers do not bother though, and no one has lost their payment through the post so far. You can also transfer your payment to the post-office account indicated above. Please include a note to indicate which version you want (NoteTab Standard or NoteTab Pro) **and your e-mail address**; see the [Order.txt form](#) for details.. Once your payment arrives, you will receive a download link to the commercial version and a key to unlock the installation program.

Due to high processing costs, please do not send a check unless you plan to purchase multiple copies or a site license for at least \$40 USD.

Order Form

To print this order form, select **Print Topic** from the **File pull-down menu**.

NoteTab™ 4.83 Order Form/Invoice

Prices in US Dollars and guaranteed through July 2001.

I would like to order:

NoteTab Pro Single Copy: _____ copies at \$19.95 each = _____

NoteTab Std Single Copy: _____ copies at \$ 9.95 each = _____

Download link: Free
Shipping & handling Free

or

On CD: \$10.00 in the United States (*) = _____

On CD: \$12.00 for other countries (*) = _____

Shipping & handling is included in prices above.
CD includes Trial versions of our other software
and some extra help files.

Total payment = _____

(*) This service is currently only available through our agent
in the USA. International orders are accepted at:

Fookes Software
P.O. Box 998
Milton FL 32572-0998
USA

Customer information is considered confidential and will not
be shared or distributed to any third party.

Ms/Mr First name [please print] Surname

E-mail address (required for download or if order incorrect!)

Company

Address

ZIP or Postal code / City

Country

Check/money order is enclosed

Payment in cash is enclosed

Only use following methods when ordering directly from author:

[] Payment made to postal account: CCP Geneva, #12-51699-7

[] Payment made to bank:

UBS SA, CP 2600, 1211 Geneva 2, Switzerland

Branch code: 0279

SWIFT code: UBS WCH ZH 12 B

Account: 279-C0612461.0 (holder: Eric Fookes)

Site Licenses

A site license for NoteTab entitles an organization to receive one copy of the distribution package and to duplicate it for the specified number of copies, or to make it available to the specified number of users on a network. Note that a site license is for a single physical site (covers a single organization for an area of up to 100 miles or 160 kilometers in radius) unless you choose the world-wide license (covers everyone in a single organization). See the topic [Ordering Information](#) to find out how you can purchase NoteTab.

The following prices are valid for NoteTab version 4.83 and are guaranteed until the end of July 2001. You can always find up-to-date prices on the following Web page:

<http://www.notetab.com/order.htm>

NoteTab Pro

Computers	USD
1 to 4	\$19.95 each
5 to 19	\$16.00 each
20 to 49	\$14.00 each
50 to 99	\$12.00 each
100 to 399	\$10.00 each
>399	\$4'000 (total for single location)
>399	\$10'000 (total for world-wide site license)

NoteTab Std

Computers	USD
1 to 4	\$9.95 each
5 to 19	\$8.00 each
20 to 49	\$7.00 each
50 to 99	\$6.00 each
100 to 399	\$5.00 each
>399	\$2'000 (total for single location)
>399	\$5'000 (total for world-wide site license)

License (Commercial version)

NOTICE TO USERS: CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT. USE OF THE NOTETAB PRO (TRIAL) SOFTWARE PROVIDED WITH THIS AGREEMENT (THE "SOFTWARE") CONSTITUTES YOUR ACCEPTANCE OF THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT INSTALL AND/OR USE THIS SOFTWARE. USER'S USE OF THIS SOFTWARE IS CONDITIONED UPON COMPLIANCE BY USER WITH THE TERMS OF THIS AGREEMENT.

1. LICENSE GRANT. Fookes Software grants you a license to use the trial version of the NoteTab Pro (Trial) software program (the "SOFTWARE") on any computer as long as the terms of this license agreement are respected. "You" means the company, entity, or individual installing or using the SOFTWARE. "Use" means storing, loading, installing, executing, or displaying the SOFTWARE. You may not modify the SOFTWARE or disable any licensing or control features of the SOFTWARE except as an intended part of the SOFTWARE's programming features.

When you first install a **trial copy** of the SOFTWARE, you are granted an evaluation period of not more than 30 days, after which time you must pay for the SOFTWARE according to the terms and prices discussed in the SOFTWARE's documentation, or you must remove the SOFTWARE from your computer. This license is not transferable to any other hardware product or other company, entity, or individual.

2. OWNERSHIP. The SOFTWARE is owned and copyrighted by Fookes Software. Your license confers no title or ownership in the SOFTWARE and should not be construed as a sale of any right in the SOFTWARE.

3. COPYRIGHT. The SOFTWARE is protected by United States copyright law and international treaty provisions. You acknowledge that no title to the intellectual property in the SOFTWARE is transferred to you. You further acknowledge that title and full ownership rights to the SOFTWARE will remain the exclusive property of Fookes Software and you will not acquire any rights to the SOFTWARE except as expressly set forth in this license. You agree that any copies of the SOFTWARE will contain the same proprietary notices which appear on and in the SOFTWARE.

4. UNAUTHORIZED USE. You may not use, copy, rent, lease, sell, modify, decompile, disassemble, otherwise reverse engineer, or transfer the SOFTWARE except as provided in this agreement. Any such unauthorized use shall result in immediate and automatic termination of this license.

5. BUNDLING. In no case may the SOFTWARE be bundled with hardware or other non-shareware software without written permission from Fookes Software (av. Eugène-Pittard 22 ter, CH-1206 Geneva, Switzerland).

6. DISTRIBUTION. Provided that you verify that you are distributing the Freeware or trial version of the SOFTWARE (you can see this information in the window title bar when you run the software: program name should be "NoteTab Light" or "NoteTab Pro trial") you are hereby licensed to make as many copies of the Freeware/trial package of the SOFTWARE as you wish; give exact copies of the original Freeware/trial package of the SOFTWARE to anyone; and distribute the Freeware/trial package of the SOFTWARE in its unmodified form via electronic means (Internet, BBS's, Shareware distribution libraries, CD-ROMs, etc.). You may charge a distribution fee for the package, but you must not represent in any way that you are selling the software itself.

Distribution of the commercial version (i.e. neither trial nor Freeware) is forbidden.

All distribution of SOFTWARE is further restricted with regard to sources which also distribute virus source code and related virus construction/creation materials. The SOFTWARE may not be made available on any site, CD-ROM, or with any package which makes available or contains viruses, virus

source code, virus construction programs, or virus creation material.

Permission to distribute the SOFTWARE is not transferable, assignable, saleable, or franchisable. Each entity wishing to distribute the package must independently satisfy the terms of the distribution license.

7. U.S. GOVERNMENT INFORMATION. Use, duplication, or disclosure by the U.S. Government of the computer software and documentation in this package shall be subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7013 (Oct 1988) and FAR 52.227-19 (June 1987). The Contractor is Fookes Software, av. Eugène-Pittard 22 ter, CH-1206 Geneva, Switzerland.

8. LIMITED WARRANTY. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS. FOOKES SOFTWARE DISCLAIMS ALL WARRANTIES RELATING TO THIS SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER FOOKES SOFTWARE NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THIS SOFTWARE SHALL BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH SOFTWARE, EVEN IF FOOKES SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIMS. THE PERSON USING THE SOFTWARE BEARS ALL RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE.

SOME JURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS OR EXCLUSION MAY NOT APPLY TO YOU TO THE EXTENT THAT LIABILITY IS BY LAW INCAPABLE OF EXCLUSION OR RESTRICTION.

IN NO EVENT SHALL ANY THEORY OF LIABILITY EXCEED THE LICENSE FEE PAID TO FOOKES SOFTWARE.

9. SEVERABILITY. In the event of invalidity of any provision of this license, the parties agree that such invalidity shall not affect the validity of the remaining portions of this license.

10. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. IN NO EVENT SHALL FOOKES SOFTWARE OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE DELIVERY, PERFORMANCE, OR USE OF THE SOFTWARE, EVEN IF FOOKES SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL FOOKES SOFTWARE'S LIABILITY FOR ANY CLAIM, WHETHER IN CONTRACT, TORT, OR ANY OTHER THEORY OF LIABILITY, EXCEED THE LICENSE FEE PAID BY YOU, IF ANY.

11. GOVERNING LAW. This agreement shall be governed by the laws of Switzerland, excluding the application of its conflicts of law rules, and shall inure to the benefit of Fookes Software and any successors, administrators, heirs, and assigns. Any action or proceeding brought by either party against the other arising out of or related to this agreement shall be brought only in a COURT of competent jurisdiction located in Switzerland. The parties hereby consent to in personam jurisdiction of said courts. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed.

12. INSPECTION AND TESTING. You agree and acknowledge that you will thoroughly inspect and test the SOFTWARE for all of your purposes upon commencement of your use. Any suit or other legal action, claim, or any arbitration relating in any way to this agreement or software covered by it must be officially filed or officially commenced no later than 90 days after your first use of the software.

13. ENTIRE AGREEMENT. This is the entire agreement between you and Fookes Software, which supersedes any prior agreement or understanding, whether written or oral, relating to the subject matter of this license.

14. RESERVED RIGHTS. All rights not expressly granted here are reserved to Fookes Software.

License (Freeware version)

NOTICE TO USERS: CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT. USE OF THE NOTETAB LIGHT SOFTWARE PROVIDED WITH THIS AGREEMENT (THE "SOFTWARE") CONSTITUTES YOUR ACCEPTANCE OF THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT INSTALL AND/OR USE THIS SOFTWARE. USER'S USE OF THIS SOFTWARE IS CONDITIONED UPON COMPLIANCE BY USER WITH THE TERMS OF THIS AGREEMENT.

1. LICENSE GRANT. Fookes Software grants you a license to use the NoteTab Light software program (the "SOFTWARE") on any computer as long as the terms of this agreement are respected. "You" means the company, entity, or individual installing or using the SOFTWARE. "Use" means storing, loading, installing, executing, or displaying the SOFTWARE. You may not modify the SOFTWARE or disable any licensing or control features of the SOFTWARE except as an intended part of the SOFTWARE's programming features. If you choose to enable the trial features, you are granted an evaluation period of not more than 30 days, after which time you may not use those features unless you purchase the commercial version of the SOFTWARE, or you must remove the SOFTWARE from your computer. This license is not transferable to any other hardware product or other company, entity, or individual.

2. OWNERSHIP. The SOFTWARE is owned and copyrighted by Fookes Software. Your license confers no title or ownership in the SOFTWARE and should not be construed as a sale of any right in the SOFTWARE.

3. COPYRIGHT. The SOFTWARE is protected by United States copyright law and international treaty provisions. You acknowledge that no title to the intellectual property in the SOFTWARE is transferred to you. You further acknowledge that title and full ownership rights to the SOFTWARE will remain the exclusive property of Fookes Software and you will not acquire any rights to the SOFTWARE except as expressly set forth in this license. You agree that any copies of the SOFTWARE will contain the same proprietary notices which appear on and in the SOFTWARE.

4. UNAUTHORIZED USE. You may not use, copy, rent, lease, sell, modify, decompile, disassemble, otherwise reverse engineer, or transfer the SOFTWARE except as provided in this agreement. Any such unauthorized use shall result in immediate and automatic termination of this license.

5. BUNDLING. In no case may the SOFTWARE be bundled with hardware or other non-shareware software without written permission from Fookes Software (av. Eugène-Pittard 22 ter, CH-1206 Geneva, Switzerland).

6. DISTRIBUTION. Provided that you verify that you are distributing the Freeware or trial version of the SOFTWARE (you can see this information in the window title bar when you run the software: program name should be "NoteTab Light" or "NoteTab Pro trial") you are hereby licensed to make as many copies of the Freeware/trial package of the SOFTWARE as you wish; give exact copies of the original Freeware/trial package of the SOFTWARE to anyone; and distribute the Freeware/trial package of the SOFTWARE in its unmodified form via electronic means (Internet, BBS's, Shareware distribution libraries, CD-ROMs, etc.). You may charge a distribution fee for the package, but you must not represent in any way that you are selling the software itself.

Distribution of the commercial version (i.e. neither trial nor Freeware) is forbidden.

All distribution of SOFTWARE is further restricted with regard to sources which also distribute virus source code and related virus construction/creation materials. The SOFTWARE may not be made available on any site, CD-ROM, or with any package which makes available or contains viruses, virus source code, virus construction programs, or virus creation material.

Permission to distribute the SOFTWARE is not transferable, assignable, saleable, or franchisable. Each entity wishing to distribute the package must independently satisfy the terms of the distribution license.

7. U.S. GOVERNMENT INFORMATION. Use, duplication, or disclosure by the U.S. Government of the computer software and documentation in this package shall be subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7013 (Oct 1988) and FAR 52.227-19 (June 1987). The Contractor is Fookes Software, av. Eugène-Pittard 22 ter, CH-1206 Geneva, Switzerland.

8. LIMITED WARRANTY. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS. FOOKES SOFTWARE DISCLAIMS ALL WARRANTIES RELATING TO THIS SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER FOOKES SOFTWARE NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THIS SOFTWARE SHALL BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH SOFTWARE, EVEN IF FOOKES SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIMS. THE PERSON USING THE SOFTWARE BEARS ALL RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE.

SOME JURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS OR EXCLUSION MAY NOT APPLY TO YOU TO THE EXTENT THAT LIABILITY IS BY LAW INCAPABLE OF EXCLUSION OR RESTRICTION.

IN NO EVENT SHALL ANY THEORY OF LIABILITY EXCEED THE LICENSE FEE PAID TO FOOKES SOFTWARE.

9. SEVERABILITY. In the event of invalidity of any provision of this license, the parties agree that such invalidity shall not affect the validity of the remaining portions of this license.

10. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. IN NO EVENT SHALL FOOKES SOFTWARE OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE DELIVERY, PERFORMANCE, OR USE OF THE SOFTWARE, EVEN IF FOOKES SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL FOOKES SOFTWARE'S LIABILITY FOR ANY CLAIM, WHETHER IN CONTRACT, TORT, OR ANY OTHER THEORY OF LIABILITY, EXCEED THE LICENSE FEE PAID BY YOU, IF ANY.

11. GOVERNING LAW. This agreement shall be governed by the laws of Switzerland, excluding the application of its conflicts of law rules, and shall inure to the benefit of Fookes Software and any successors, administrators, heirs, and assigns. Any action or proceeding brought by either party against the other arising out of or related to this agreement shall be brought only in a COURT of competent jurisdiction located in Switzerland. The parties hereby consent to in personam jurisdiction of said courts. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed.

12. INSPECTION AND TESTING. You agree and acknowledge that you will thoroughly inspect and test the SOFTWARE for all of your purposes upon commencement of your use. Any suit or other legal action, claim, or any arbitration relating in any way to this agreement or software covered by it must be officially filed or officially commenced no later than 90 days after your first use of the software.

13. ENTIRE AGREEMENT. This is the entire agreement between you and Fookes Software, which supersedes any prior agreement or understanding, whether written or oral, relating to the subject matter of this license.

14. RESERVED RIGHTS. All rights not expressly granted here are reserved to Fookes Software.

We Are an ASP Member

What is Shareware?

Shareware distribution gives users a chance to try software before buying it. If you try a shareware program and continue using it, you are expected to register. Individual programs differ on details -- some request registration while others require it, some specify a maximum trial period. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

Copyright laws apply to both shareware and commercial software, and the copyright holder retains all rights, with a few specific exceptions as stated below. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. (In both cases, there are good programs and bad ones!) The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all and sundry or to a specific group. For example, some authors require written permission before a commercial disk vendor may copy their shareware.

Shareware is a distribution method, not a type of software. You should find software that suits your needs and pocketbook, whether it's commercial or shareware. The shareware system makes fitting your needs easier, because you can try before you buy. And because the overhead is low, prices are low also. Shareware has the ultimate money-back guarantee -- if you don't use the product, you don't pay for it.

The Ombudsman Statement:

Fookes Software is a member of the Association of Shareware Professionals (ASP). ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with an ASP member by contacting the member directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provide technical support for members' products. Please write to the ASP Ombudsman at 157-F Love Ave., Greenwood, IN 46142 USA or send an e-mail message to omb@asp-shareware.org.

The Ombudsman may be contacted by FAX at the ASP FAX number: (317) 888-2195. In communication with the Ombudsman, please include a telephone number and/or FAX number if available.

[Find Help](#)
[Edit Documents](#)
[Use the Editor Clipboard](#)
[Create Clipboard Libraries](#)
[Use Web and Document Links](#)
[Calculate in NoteTab](#)
[Manage Documents](#)
[Manage File Formats](#)
[ASCII Documents](#)
[EBCDIC Documents](#)
[Use Templates](#)
[Create Outline Documents](#)
[Convert a Normal Document to Outline](#)
[Create Web Pages](#)
[Apply Perl and Gawk Scripts on Text](#)
[Keep a Log](#)

Find Help

Almost every menu item, button, and input field in the program offers hint information. Whenever you highlight a menu item or place the mouse cursor over a dialog box element, its hint text is displayed in the program's status bar (at the bottom of the main program window).

If you want to make the most out of NoteTab, read the tips that are displayed in the Tip-of-the-Day window. The tips can be selected and copied for future reference.

If the documentation provided with NoteTab does not answer your question, you may find the appropriate information by visiting the NoteTab Web site at:

<http://www.notetab.com> (main site)

or

<http://www.notetab.ch> (Swiss site)

You will find links to an FAQ list, a [mailing list](#), patches, Clipbook libraries submitted by other users, and of course information about the latest updates.

Edit Documents

NoteTab lets you open many documents at once and makes them available to you through the tab bar. You can configure the tab bar to show or hide the extension part of the name. When you choose the "File | New" command, NoteTab automatically creates a name for the new document. You can change the format of this name by editing the field "Options | General | New Name Format".

When you modify a document, its little disk icon in the tab turns red. This visual cue makes it easy to identify documents that need saving if you want to preserve the changes. When you save the document, its disk icon becomes gray again.

You will find many useful commands in the Modify menu that help you process your text. These commands make it easy to join or split lines, sort lines, indent or unindent lines, change alignment and character case, insert bullets or line numbers, strip HTML tags or convert text to HTML, etc.

NoteTab offers many features and tools to help you write documents efficiently. If used properly, the Editor Clipboard is probably the feature that will save you the most time. It lets you store and reuse boilerplate text, which can be anything from a single character to complete multi-megabyte documents. Document templates are another time-saving feature. A template is a special kind of document that provides content for shaping a new document. You can use several templates to create many common types of documents such as letters, Web pages, etc.

Use the "Tools | Text Statistics" command to get a detailed count of the whole document or highlighted text. The dialog box will indicate how many characters and words it found and by clicking on the More button, you get the count for individual words.

People often organize their information/notes in individual documents. If you have many such documents, they can become hard to manage and occupy more space than necessary on your disk. Outline documents are the ideal solution in this case. They make it really easy to organize your information in a structured format. You can copy the content of each file under its own Outline heading. The result is a single document containing all your notes organized into clear and easily accessible categories.

NoteTab lets you view two documents side-by-side by using the "Document | Second Window" command (the active document is placed in the right window). With NoteTab Std, you can drag-and-drop highlighted text between the two windows.

You can customize the editor's main shortcut menu by choosing the commands you find most useful from the "Options | Shortcut Menu" list box. Your important commands will be just a mouse-click away. See the topic on Keyboard Shortcuts for a detailed list of keys you can use to speed up your work.

Use the Editor Clipbook

You can think of the Clipbook as a visible clipboard with multiple text items that can be pasted in any document. The pasteable text item can be anything from a single character to thousands of lines of text. Each item is stored in a [Clipbook library](#) and is identified by a header which is displayed in the Clipbook window. The header can either represent the actual text to be pasted, or a brief description of the text item. A simple [Clip syntax](#) can be used to produce specialized Clips. Thanks to this syntax, the Clipbook can also be used as a "Launch Pad" to open URLs and execute other programs (with or without the active document), send e-mail, and perform file management tasks. The topic [Create Clipbook Libraries](#) explains how you can edit your own libraries.

To paste a Clip, first position the cursor in your document, then go to the Clipbook window and double-click (or press Return) on the item header. You can also drag-and-drop items from the Clipbook to your document. If [Auto-replace mode](#) is enabled (either by opening an "auto-replace" library, or through the Tools | Auto-replace Mode menu command), NoteTab will check for a matching item in the active library whenever you press any of the following characters (Clip activation keys):

TAB SPACE RETURN !" , . : ; = > ?] }

If a matching Clip is found, the typed item in the document is automatically replaced according to the Clip instructions. For a demonstration of this feature, try the included libraries AutoCorrect and HTML-ar.

You can copy an item to Windows' clipboard by selecting its header and pressing the Ins key or by using the shortcut menu. This feature makes it easy to paste Clips into other programs.

There are several options that let you configure the Clipbook to best suit your needs. For example, you can dock the Clipbook on the left or right side of the document frame. You can also change the font and color used for the Clipbook list box. If the Default Font option is enabled, the list box font will match the font used by the active document. You can access all the Clipbook options by invoking the shortcut menu from the list box or by opening the Clipbook Properties submenu from the Tools menu.

You can resize the Clipbook list horizontally by dragging the splitter bar (between the Clipbook and the document frame).

If you have a long list of Clips or you would like to reduce mouse movement, you can invoke the [In-Context Clipbook](#). This original feature lets you easily pick Clips directly within your document at the cursor position.

The Clipbook is also great as a "shorthand" glossary tool. For example, if you add the text "Yours sincerely" in the Clipbook and give it the header "ys" (without the quotes), then next time you type "ys" in a document and hit the F2 function key, "ys" will be replaced by "Yours sincerely". You do not need to type the full Clip header; just type enough text to match the item you want to use. For example, if you use the HTML library and type the letter "c" followed by the F2 key, then the "Center" Clipbook item will be pasted in your document. If you frequently use this glossary feature, you may prefer to hide the Clipbook list so it doesn't clutter your desktop. Open the Clipbook shortcut menu and choose Hide Clipbook. Just press F4 if you want to make the Clipbook visible again. If you close the Clipbook instead of hiding it, you will not have access to the feature until you reopen it.

The list-box shortcut menu lets you add or edit items, sort the list, save or undo changes, and modify configuration options. You can also edit the content of the library by using the commands under the Tools | Clipbook menu.

You can toggle between the available Clipbook libraries by clicking on one of the library buttons (above the status bar) or by pressing the *Alt* key and a number from 1 to 0 (0 = 10); the number corresponds to the library's position in the Clipbook's combo box.

If you visit one of the NoteTab sites, you will find a large selection of Clipbook libraries for all kinds of tasks (HTML, words for writing shopping lists and recipes, PGP, a maze game, COBOL, envelope printing, LaTeX, Perl, Turbo Pascal, Java/Javascript, Cascading Style Sheets, VRML, Mathematics, etc.). At this time of writing, there are more than 100! You can download them from either of the following Web pages:

<http://www.notetab.com/libraries.htm> (main site)

or

<http://www.notetab.ch/libraries.htm> (Swiss site)

Once you have downloaded a library, just unzip it into the Libraries folder under the NoteTab program directory to install it. Or even easier, you can use the "Install Libraries" Clip in the Utilities library (click on the button labeled "Utilities" at the bottom of the NoteTab window to open it in the Clipbook).

Useful Mouse and Keyboard Shortcuts

- *Insert* copies the selected Clipbook item to Windows' clipboard.
- *Delete* deletes the selected Clipbook item.
- *Enter* or *Double-click* launches a command or pastes the item into the document.
- *Ctrl+Space* pastes or executes the currently selected Clip.
- *Ctrl+Shift+C* adds text selected in the document to the current Clipbook library.
- *Ctrl+Shift+E* opens the Clip editor.
- *Ctrl+F2* adds a new Clip
- *Shift+Ctrl+F2* adds Clip from Clipboard
- *Shift+Ctrl+X* cuts selection
- You can *drag-and-drop* items to change their order in the list.
- *F4* opens or closes the Clipbook.
- *Ctrl+G* shifts the focus from the Clipbook to the editor.
- *Alt+1* displays the first Clipbook library in the combo box list; *Alt+2* displays the second, and so on...

Create Clipbook Libraries

There are two ways to create libraries and add items to the Clipbook. The easiest way is to use the Clipbook window and Clip editor. The fastest method, however, is to create them manually.

1. With the Clipbook Tool

First you will have to open the Clipbook tool; press F4 or click on one of the library buttons at the bottom of the NoteTab window. The easiest way to create a library is by typing a new name in the library combo box which is just above the Clipbook list. Once you've typed a name for your library and pressed the Enter key, NoteTab will ask you if you want to create a new library. When you answer yes, it will add the new library name to the combo box. The Clipbook list window will be empty at this stage. Next, open the Clipbook shortcut menu by clicking in the empty list window with the right mouse button. Choose the Add New Clip command. This opens the "Clipbook Item Editor" dialog box. Type a name for your first Clip and press Enter or click on the OK button. This will add the new item to the list and open the Clip editor. Use the Clip editor to add or edit the content of the current Clip. When you are ready to add a new Clip, use the Clipbook shortcut menu again or press Ctrl+Shift+A. You can also create a new item by pressing the page down key once the cursor has reached the end of the text.

NoteTab lets you define two special types of headers. If you want to group headers under a main topic, just place a semicolon character ";" in front of the Clip name. These headers don't have a left margin icon and are colored in red by default. If you want to create a header that cannot be activated with the mouse or the ENTER key, just place an underscore character "_" in front of the Clip name. These headers are colored in gray by default.

Changes made to the content of a Clip are automatically saved in memory when you move to another item. Once a different Clip is displayed in the editor, changes made to the previous item cannot be undone with the Undo/Redo commands. Also, bookmarks set with NoteTab Pro will not be remembered. When the Clip editor is open, you can save a modified library to disk just like any other document, with the Save menu commands or buttons.

If you want to edit an existing item and the Clip editor is not open, choose the Tools | Clipbook | Clip Editor menu command or press the shortcut key Ctrl+Shift+E to open it.

If you don't want to edit Clipbook libraries directly, you can add new items by selecting text in your document and then pressing Ctrl+Shift+C (or selecting Add to Clipbook from the editor's shortcut menu) to add it to the current Clipbook library. You will then be prompted to enter a header for the text, after which it is added to the Clipbook.

You can also add text from other applications by copying it to Windows' clipboard. Open the Clipbook's list box shortcut menu and select Add from Clipboard.

2. Manually

NoteTab also lets you edit a Clipbook library like a normal document. Use the menu command Tools | Clipbook | New Library to create a new library or use Tools | Clipbook | Open Library to open an existing one. The structure of a Clipbook library is quite simple. It starts with a header on the first line followed by Clip headers and their content. For more information on editing them, see the [Clipbook Libraries](#) topic.

Use Web and Document Links

If your text contains an Internet link, holding the Control key down and double-clicking on the link will open it in your default Web browser. Also hold the Ctrl key down if you want to open a link in a new Web browser window (this feature may not work with all browsers). If this feature does not work, you will have to fill in the path and file name of your browser in the "Options | Internet | Default Browser" field. If the editor cannot resolve the URL from the text you double-clicked, it will open a dialog box so you can complete the address if necessary.

NoteTab also supports links between documents, Outline headings, and even words within a document. These types of links are referred to as "Hyperlinks".

Hyperlinks must be placed between square brackets "[]". You can jump to a hyperlink topic by holding down the Control (or Ctrl) key while you double-click on the text between the brackets. Or, if you prefer, place the cursor anywhere between the brackets and press Ctrl+H to go to the matching topic. If you set the menu option "Tools | Quick Open Links", you do not need to hold down the Control key when clicking on a hyperlink.

Hyperlinks can be inserted in any kind of document. They can represent a heading in the active Outline document, or another Outline document, or the name of a file, or an Internet link, or a program to launch, or a line number, or a text search criteria. Character case is not important unless you use the search criteria feature with case sensitivity enabled.

Examples:

- * If you activate the [introduction] hyperlink, the content of that Outline topic will be displayed.
- * If you have another Outline file called "Important.OTL", you can create a hyperlink that will display it and select the topic you specify. For example, [important.otl::meetings] will display the "meetings" heading. Note the use of the double colon to separate the main link criteria from the secondary criteria.
- * The accompanying "WhatsNew.txt" file can be opened with this hyperlink: [WhatsNew.txt].
- * The following command [explorer.exe] will launch Windows Explorer. You can also specify parameters as long as you place the program's file name between double quotes ["explorer.exe" /e,/root,C::]. Note that the program name must include the extension too (usually .exe).
- * The following hyperlink [readme.txt::outline] will display the "readme.txt" file. As it is not an Outline file, NoteTab assumes the second criteria "outline" is text criteria to search and will select the first occurrence of the word "outline". Note that you will have to specify the full path name if the file is not in the same directory as the document containing the hyperlink.
- * If you want to apply the search criteria type of hyperlink on the active document, just omit the file name. For example, [::NoteTab] will find the next occurrence of the name NoteTab. Since the double colon does not necessarily look very nice in this case, the following format with a blank space on each side of the search criteria is also valid: [NoteTab].
- * By default, the search feature is case sensitive, only matches single words, and searches forwards. However, you can change this behavior by using extra parameters: "I" indicates that the search should not be case sensitive, "T" indicates that any text is acceptable, and "B" tells NoteTab to search backwards. Use the "^" character to separate the search criteria from its parameters. For example, [tab^ITB] will select the "Tab" part of the name NoteTab preceding the hyperlink. And [readme.txt::NoteTab^B] will find the last occurrence of the word "NoteTab" in the Readme.txt file.
- * To jump to a specific line number in a document, the format of the hyperlink is similar to the search

criteria. You use the line number instead of the search text and use the parameter "L" so that NoteTab knows it is the line number you want and not text to search. For example, [readme.txt::5^ L] will display the Readme.txt file with the cursor at the beginning of the 5th line. If you specify "-1", the cursor will be placed on the last line: [readme.txt::-1^L].

* URLs are also valid. However, if the link does not start with the typical "www.", then you must fully qualify the URL (add "http://", or "ftp://", etc.). The following will connect to the NoteTab site through your default Web browser [www.notetab.com]. Hold the Ctrl key down if you want to open the document in a new Web browser window (this feature may not work with all browsers).

Calculate in NoteTab

NoteTab can also calculate mathematical expressions in a document. Expressions can have up to 40 imbedded parentheses. Note that you may need to break down very long expressions into groups of parentheses if you get an error message.

For example, if you enter the following expressions in a document:

$\text{Sin}(3)*5/\text{Pi}=\text{$

and then press Ctrl+E (or use the menu command "Tools | Calculate Expression"), the result will be added after the equal sign. If there is no equal sign, or if the document is Read-Only then it copies the result to the Clipboard and displays it in a small window. You can also add numbers arranged in a column; just select all the numbers in the following four lines and press Ctrl+E:

23
-5
42
40

Supported mathematical operators and functions are listed below.

Predefined value:

PI

Accepted operators:

+ , - , * , / , ^ , MOD, DIV

The following functions are supported; it doesn't matter if you use lower or upper case:

COS, SIN, SINH, COSH, TAN, COTAN, ARCTAN, ARG, EXP, LN, LOG10, LOG2, LOGN, SQRT, SQR, POWER, INTPOWER, MIN, MAX, ABS, TRUNC, ROUND, INT, CEIL, FLOOR, HEAV (heav(x) is 0 for $x < 0$, 1 for $x \geq 0$), SIGN (sign(x) is 0 for $x = 0$, 1 for $x > 0$, -1 for $x < 0$), ZERO (zero(x) is 0 for $x = 0$, 1 for $x \neq 0$), RND (rnd(x) = int(x) * Random) RANDOM (random(X) = Random; the argument X is not used)

Functions like MIN and MAX take two values; you must use a semi-colon ";" to separate them. For example: MAX(2;3)

Trigonometric functions take values in radian units.

Acknowledgments:

The mathematics feature was made possible thanks to the superb freeware parsing function developed by the following people: Renate Schaaf (schaaf@math.usu.edu), Alin Flaidar (aflaidar@datalog.ro), and Stefan Hoffmeister (Stefan.Hoffmeister@Uni-Passau.de)

Manage Documents

You can quickly reopen recently used documents by using the "File | Reopen" command. It stores up to 16 files with the most recently used ones listed at the top.

You can also organize groups of files into Favorites categories. You can configure NoteTab to open these automatically when you launch the program. The Favorites menu offers many commands that let you maintain lists of files and directories.

Use the Quick List tool to display files stored on your disk or to show the contents of your current Favorites category. You just need to double-click on the file or directory name to open it.

NoteTab can be configured to reopen your documents as you left them when you last used the editor. Just set the "Options | General | Reload Open Documents" checkbox.

On the "Options | Files" tab, you have a number of settings that you can adjust to determine where the File Open/Save dialog box should point when you open it. For example, if "Match Directory" is enabled, the main File Open dialog box will point to the same directory as the active document.

Manage File Formats

NoteTab automatically detects the format (DOS/UNIX/Mac) of a text file when you open it. During the loading process, UNIX and Mac files are converted to the DOS/Windows format. Through the "Options | Documents | Save As" setting, you can determine what format should be used when the document is saved. If you choose "Original", the original format is restored.

You can open and edit DOS ASCII files in NoteTab. As there is no way the program can automatically detect whether a file contains ASCII or ANSI characters, you must specify the format yourself when you open the file. By default, NoteTab assumes your files use the standard ANSI character set. See the [ASCII Documents](#) topic for more details.

NoteTab also lets you open and edit (commercial and trial versions only) [EBCDIC files](#). As there is no way for NoteTab to automatically detect such files, you must open them by choosing a file filter in the File Open dialog box that contains the text EBCDIC. NoteTab will prompt you for a line width if it detects a fixed-width text that does not use the standard widths of 80, 132, and 133 characters; use a value of zero if you want to specify a variable-width file with line breaks.

ASCII Documents

ASCII is the character set used under DOS and ANSI is used under Windows. If you don't use extended characters (such as äèüö etc.), there is no difference between the two. However, if you create a document under DOS that uses extended characters and want to display them correctly under Windows, the text must be "translated" from ASCII to ANSI in order to display the accented characters correctly. Also, if you create a Windows text document (ANSI) and want to use it in a DOS application, it must be converted to ASCII if it contains extended characters.

This editor has a built-in filter that makes it easy to edit files of either format. Note, however, that a conversion between the two formats may not always translate all characters correctly. There are mainly two reasons for this: 1) the extended ANSI characters are not necessarily supported by all font types; 2) DOS has many different character sets available, which means that extended characters are often quite different from one set to the other. The filter used in this program is based on ASCII 850 Multilingual (Latin I).

By default, the file filter list has an item for opening and saving DOS Ascii files. Since you can change the order of the filters and edit them, the program just checks for the word ASCII (case does not matter) in the selected filter when you use the file dialog box. If it finds the word ASCII, it assumes that you want to open or save a file in the DOS Ascii format. So if you want to change the default file filters but also want to use ASCII documents, remember to put the word ASCII in the description part of at least one of the filter items! You can edit the filters by opening the Options dialog box and selecting the File Filters tab.

If you convert a document from ASCII to ANSI and want it to keep its ANSI format permanently, you will have to open the Document Properties dialog box and uncheck the Dos ASCII option.

EBCDIC Documents

With the addition of EBCDIC support, NoteTab has become a valuable tool for any project involving editing or converting EBCDIC files from MVS and AS/400 systems. As a result, you can now use NoteTab for those huge Y2K projects.

EBCDIC files that do not contain packed-decimal fields can be opened, edited, and saved in the commercial and trial versions of NoteTab. Such files can either be opened through the new ^!OpenEbcadic Clip command, or by using a file filter containing the text "EBCDIC" (based on the same principle as opening Dos ASCII files). The Export command can also be used to save text in EBCDIC format. Although NoteTab supports EBCDIC files with line breaks and those with fixed-width text, it cannot correctly read EBCDIC files with variable-length records (i.e. those without line breaks).

NB: If you are using NoteTab Light, you will only be able to read EBCDIC files. It will not let you edit, save, or copy the text to the Clipboard. However, if you enable the trial mode (available for one month), you will be able to edit and save EBCDIC files just like the commercial versions of NoteTab.

How to add an EBCDIC file filter

If you have upgraded from a NoteTab release prior to version 4.6, you probably need to add an EBCDIC file filter yourself, as the default settings will have been taken from the INI file of your previous version. Just open the Options dialog box (from under the View menu) and activate the File Filters tab. Click on the New button to add a new entry in the list, then add the text "EBCDIC files" (without the quotes) in the Description field, and the wild cards that match your EBCDIC file names (the following are typical values: "*.cbl;*.cob;*.cpy;*.ddl;*.bms") in the Wild Cards field. Finally, use the Move Up/Down buttons, or drag-and-drop the added item to the position in the list that you prefer, and then click on OK to close the dialog box. Now, you will see your new file filter in the File Open dialog box when you next choose to open a file.

How to open an EBCDIC file

1. Choose the Open command from the File menu or the toolbar. Select the file filter for EBCDIC files (if you do not have this filter, follow the procedure described in the previous paragraph), choose the file you want to open (you can select multiple files), and click on OK. At this stage, NoteTab will try to determine the format of the file. If it detects line breaks, it will assume that the file contains variable-length lines. If not, NoteTab checks if the file size matches fixed-width lines of 80, 132, or 133 characters. If it does, NoteTab will open the file according to the matching format. If not, NoteTab will prompt you for the line length used in the file you are opening.
2. Open the "Utilities" Clipbook library – you can do this by clicking on the button labeled "Utilities" at the bottom of the NoteTab window. You will see a Clip labeled "Open EBCDIC document" under the "EBCDIC files" section. Double-click on the Clip to activate it. You will then be presented with a Clip wizard with a field for the file name and the LRECL format. Complete the fields and click on OK to open the EBCDIC file.

Next time, when you want to reopen such a file, NoteTab remembers its original format (in the properties file), so the text will be correctly displayed if you use the Favorites tool or the File|Reopen submenu. However, if you change the format with another program, you should use NoteTab's File Open dialog box with the appropriate filter so that the text is not interpreted as being EBCDIC.

How to save a document in EBCDIC format

NoteTab saves documents based on the "Save As" format setting defined either in the general options (if default properties are used), or in the document properties. The general options are configured through NoteTab's Options dialog box (under the View menu) on the Documents tab, and the document's properties are set in the Document Properties dialog box available from the Document menu.

If the "Save As" format is set to "Original", a file opened as EBCDIC will automatically be saved in EBCDIC format. If the option is set to EBCDIC, then all the documents to which this setting applies will be saved in EBCDIC format. Any other format will result in a non-EBCDIC file.

If you want to create a new EBCDIC document, use the New command from the File menu or the toolbar to create a new document. Then open the Document Properties dialog box (from the Document menu) and change the "Save As" setting to EBCDIC. As a result, the next time you save this document, it will be stored on disk in EBCDIC format.

You can also create an EBCDIC copy of any document (irrespective of the original format) or selection of text by using the Export command under the File menu.

Using alternative EBCDIC conversion tables

The default conversion table is stored in a file called "Default.ebc" in the NoteTab program directory. It is based on the US character set. This is a plain text file that can be edited in NoteTab. Characters are represented as hexadecimal numbers. The values are placed in chronological order, one set of values per line, starting at 00h and ending at FFh. You can create as many conversion tables as you like, each one in its own file.

When an EBCDIC file is opened or saved, NoteTab uses the last loaded table, or the default internal table if none was loaded, to perform the character conversion between EBCDIC and ANSI. You can either load the conversion table when NoteTab is started, by using the "/T" command-line parameter, or use the Clip command "^!LoadEbcDicCharTable", or use the Clip labeled "Load EBCDIC conversion table" in the Utilities library.

EBCDIC related Clip commands

^!LoadEbcDicCharTable FileName

Loads the specified file containing the EBCDIC character tables for conversion to and from Windows ANSI. The NoteTab program path is added if you do not specify a fully qualified path name. The extension .ebc is added if you do not specify a filename extension. You can reset the default character table by specifying the Default.ebc file.

^!OpenEbcDic FileName [/W=nnn]

or

^!OpenEbcDic FileName1;FileName2;FileName3

Opens and translates the text from EBCDIC format to ANSI, or selects the specified document "FileName". You can use wild cards with this command. The optional parameter "/W=nnn" can be used to define a line length for fixed-width EBCDIC files. "nnn" represents the width in characters; a value of 0 indicates variable line lengths delimited with carriage returns. Example:

^!OpenEbcDic /W=132

^!SaveEbcDic FileName [/W=nnn]

Similar to the ^!Save command (and accepts the same parameters) but saves the document in EBCDIC format. The optional parameter "/W=nnn" can be used to define a line length for fixed-width EBCDIC files. "nnn" represents the width in characters; a value of 0 indicates variable line lengths delimited with carriage returns. Example:

^!SaveEbcidic /W=132

Use Templates

A template is a special kind of document that provides content for shaping a new document. You can use several templates to save time in creating many common types of documents.

Templates are slightly different from normal documents in that their first line contains the template header definition. The header informs NoteTab that the file being opened is a template; it can also tell NoteTab what name it should generate for the new document. For example, the default header looks like this:

```
= DocTemplate "Notes%0.3d.txt"
```

Use the menu command "File | Template | New" to create a new template. You can also convert your current document to a template by inserting a template header with the menu command "File | Template | Add Header". When you save such a template, make sure you give it the extension .tpl.

Headers must always start with an equal (=) sign as the first character on the line followed by the word "DocTemplate" (case is not important). After that first part, you can add the text format that NoteTab should use to generate the names of documents based on the template. This information is optional and must be enclosed in double quotes (") when used. The code "%d" indicates where the incremental numbering is inserted in the name. In the above example, the "0.3" between the % and d tells NoteTab to format numbers smaller than 3 digits with leading zeroes. If you omit the extension, the currently defined default extension will be added when the name is generated. You can also include a path name if you always want the documents saved in the specified directory.

Add this type of header on the first line of any document and it will become a template once you save it. Then, whenever you open it through the "File | Open" or "File | Template | Open" menu command, a new document based on your template will be created. If you need to modify the content of a template, use the "File | Template | Edit" menu command.

Starting with version 4.6, NoteTab template files accept Clip commands and functions embedded in the text. When such a file is opened, the text contained in it is interpreted and the resulting text is displayed in the document. To enable this feature, you will have to include the keyword "ClipCode" in the header line of the template file. See the template file HTML.tpl for an example of embedded functions. Example of header:

```
= DocTemplate ClipCode "Home%0.3d.htm"
```

Note that you cannot close a template document through its own embedded Clip code.

Since version 4.8, template headers accept the ^\$GetDate()\$ function in order to use the current date in the file name. Example:

```
= DocTemplate ClipCode "Notes-^$GetDate(yyymmdd)$-%0.3d.txt"
```

Since version 4.82, template headers can use Clip directory functions to represent special path names (e.g. ^\$GetScriptPath\$, ^\$GetTmpPath\$). Example taken from FTP.tpl:

```
= DocTemplate ClipCode "^$GetScriptPath$FTP%0.3d.scr"
```

Create Outline Documents

Outline documents make it easy to organize your information in a structured format without using a database application or splitting your text into multiple files. An Outline document is composed of two parts: the Headings list (or table of contents) on the left and the Contents editor window on the right. When you select a heading, its content is displayed in the editor window. Open the accompanying `Readme.OTL` file in NoteTab to see an example of an Outline document.

You can have up to about 5400 headings per Outline document and a heading can contain many megabytes of information. An Outline file can theoretically have a maximum capacity of 2GB.

You can create your own outline documents by choosing the "New Outline" command from the File menu. Unlike standard documents, you must choose a name before it is created. You will also be prompted to enter a name for the first heading. Once you have completed those steps, you are ready to enter content for your first heading.

To create a new heading, just press `Shift+Ctrl+A`. You will then be prompted to enter the name for the new heading. Note that you do not need to save the content of modified headings as you move through the list – this is automatically taken care of by NoteTab. `Shift+Ctrl+R` will remove the current topic. If you can't remember these shortcuts, just place the mouse cursor over an empty part of the Headings window and look at the hint in the status bar. You will find more commands and shortcuts listed in the "Documents | Outline Heading" submenu. You can also invoke the shortcut menu from the Headings window.

You can add hyperlinks to other headings. When activated, the content of the linked heading is displayed. Use the "Documents | Outline Heading | Copy Heading" menu command (or the keyboard shortcut `Shift+Ctrl+B`) to easily create a new heading based on selected text in your outline document. This command automatically inserts square brackets around the selection, adds the heading to the list, and opens it for input.

The `PageUp/PageDown` keys let you easily scroll from one heading to the next. If the cursor is at the beginning of the heading's content, pressing `PageUp` will display the previous heading. Pressing `PageDown` when the cursor is at the end of the heading's content will display the next heading. If the last heading is already selected, NoteTab will ask you if you want to add a new heading. You can reorder headings by simply dragging them with the mouse to a new position.

Just like any other document, you can have multiple Outline documents open simultaneously. Outline files are very similar to Clipbook libraries and do not contain any control codes other than plain text characters (i.e. no binary characters). As a result, you can easily open them in any text editor or word processor (although you won't benefit from the outline layout), or send them in e-mail messages.

Convert a Normal Document to Outline

There is a relatively quick way to convert an existing document into an outline. Basically, an outline file uses the same structure as a Clipbook library. To start off, create a new library file by using the Tools | Clipbook | New Library menu command. This will create a new document with the appropriate Clipbook header. Add the keyword Outline to the header. Next, copy the document that needs converting into the new library file (You can use the Document | Insert File command). Then, change all the topic headings so they use the following format:

```
[Blank Line]
H="Topic Heading"
[Topic Content]
```

You can use the Clipbook tool to help you with this task if you have many headings. Create a Clip with the following text (note that it should not start or end with a blank line; otherwise the Clipbook tool will be confused):

```
H="^&"
```

Once you've added this Clip, all you need to do now to convert a heading is select it and double-click on the Clip.

The following is an example of an outline file with three headings:

```
= V5 Outline MultiLine NoSorting TabWidth=30
```

```
H="Heading #1"
Content #1
```

```
H="Heading #2"
Content #2
```

```
H="Heading #3"
Content #3
```

Once you've finished converting the document, choose Save As and rename the file so that it uses the ".otl" extension. Close and then reopen the document... That's it.

Create Web Pages

NoteTab has many features that make it easy to create or convert documents to Web pages. Note, however, that only [NoteTab Pro](#) offers HTML-syntax highlighting. Support tools include the [Editor Clipbook](#), the [Quick List](#), and the Modify menu which offers powerful commands to convert characters, selected text, or entire documents to HTML.

You can produce Web pages from [Outline documents](#), complete with a table of contents, headers for each topic, and links between them. To see for yourself, open the Readme.OTL document and use the menu command "Modify | Document to HTML | With Paragraph Tags". Once the new HTML document is created, hit the F8 shortcut key to view it in your default browser. Edit the settings on the "Options | Outline" tab to configure the format of the Web page. Creating this type of HTML document has never been easier. Webmasters maintaining FAQ lists will find this feature particularly useful.

The [Editor Clipbook](#) is a powerful tool for adding HTML tags to your text. Most HTML editors let you add tags through toolbars loaded with buttons or cascading menus. This approach is awkward to say the least. With the Clipbook, tags are clearly identified by meaningful names, the list is always in view once you open it, and it can contain a virtually unlimited number of items (tags or frequently used text). You can easily customize the Clipbook HTML libraries to suit your needs. A little icon next to each item indicates whether it contains a tag that can wrap around selected text (a "+" symbol) or not (a circle). Some Clips even accept user input before they are pasted in the document (see the topic [Clipbook Wizards](#) for more information). You can insert Clips by double-clicking on them, by dragging them into the document, by typing the first matching characters and hitting the F2 function key, or by invoking the [In-Context Clipbook](#) using the ESC key. Use Ctrl+Spacebar as a shortcut to insert the highlighted Clip.

Adding links to other Web pages or images stored on your computer is really easy when you use the [Quick List](#) tool. When it is configured to display disk files or [Favorites](#), file names can be dragged from the list into the document. If the extension of the document is listed in the "HTML File Extensions" list box (Options dialog box), the appropriate link tag is inserted at the point where you drop the file. Tags for image files (GIF, PNG, JPG, and BMP) will even have the correct size attributes completed. You can also insert links to other Web documents or images by using the "Document | Insert HTML Link/Image" menu command (Shift+Ctrl+F5). Color codes can be inserted by using the "Document | Insert HTML Color" menu command (Ctrl+F5).

Changing a URL or any other type of text (even multiple lines) across many Web pages is also easy with NoteTab. Two methods are available: either you open all the files that are likely to contain the searched text and then use the "Search | Replace" command (Ctrl+R) to scan through them, or use the "Search | Search Disk" command (Ctrl+D) to scan your disk files.

Use [Document Templates](#) to create new documents that already contain the framework of your Web page layout. This feature will save you a lot of typing when you add Web pages to your project. A sample HTML template is provided with NoteTab.

You can view your HTML document in your main Web browser by using the F8 function key. Hold the Ctrl key down if you want to open the document in a new Web browser window (this feature may not work with all browsers). There is no need to save your changes when you use this feature. If NoteTab detects that your document has been modified, it stores the changes in a temporary file and sends it to the browser. Temporary files are automatically deleted when you exit NoteTab.

You can organize your HTML documents in multiple categories by using the flexible [Favorites dialog box](#) (Shift+F2). The Favorites feature can exploit an unlimited number of categories. Items in the list can represent file names (including wildcards), directories, and the names of other categories.

NoteTab makes the popular "The HTML Reference Library" (by Stephen Le Hunte) available from the

Help menu. If you have HTMLib version 3.0 or older, you can install the package in the same directory as NoteTab. Another method that works with all versions is to create a shortcut* to the HTMLib.exe file in the NoteTab directory. The highly recommended "HTML Reference Library" is available from:

<http://www.htmlib.com/>

(* to create the shortcut: 1. open Explorer and select the HTMLib file; 2. press Ctrl+C to copy it to the Clipboard; 3. select the NoteTab directory and choose the command "paste Shortcut" from the Edit menu. The menu command will be displayed in NoteTab after you restart it.)

Apply Perl and Gawk Scripts on Text

NoteTab integrates seamless support for Perl and Gawk interpreters; scripts can be stored in NoteTab's Clipboard and applied directly to highlighted text or the whole document.

If you don't have a Win32 Perl or Gawk interpreter installed on your computer, there are a number of good ones you can download from the Internet. The following Freeware 32-bit interpreters have been tested with NoteTab and work very well:

For Perl:

<http://www.ActiveState.com/pw32/>

For Gawk:

<ftp://oak.oakland.edu/pub/simtelnet/gnu/gnuish/gawk304x.zip>

All you need to do is install them wherever you want on your hard disk. Then, the first time you try to run a script, NoteTab will prompt you for the file name of the interpreter. NoteTab will only prompt you again if you delete or move the interpreter to another location, or delete the NoteTab INI file.

The easiest way to use Perl or Gawk is to store each script in its own Clip. Then, create an executable Clip with the command to launch the interpreter that corresponds to the script language you have written. For example, if you have stored a Perl script for reversing the order of characters in a Clip labeled "Reverse", all you need is the following instruction to apply it to the text in the active document:

`^!RunPerl Reverse`

If text was selected in the document, then the script will only be applied to the selection. If no text was selected, then the script is applied to the whole document (with outline-type documents, only the current topic is used). The output from the Perl script then replaces the text to which it was applied.

You will find several examples in the "Samples.clb" library that accompanies the NoteTab package. See the topic "[Create Clipboard Libraries](#)" for more information on creating libraries.

Commands:

`^!RunPerl ScriptName`

Applies specified Perl script to selection in current document or whole document. ScriptName can either represent the name of a Clip or a script file. The stderr stream is saved in file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdError Name command.

`^!RunGawk ScriptName`

Applies specified GAWK script to selection in current document or whole document. ScriptName can either represent the name of a Clip or a script file. The stderr stream is saved in file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdError Name command.

For more information, see the topics [Using Perl, Gawk, and Other Scripts](#).

Keep a Log

You can keep a log by using the same method as Notepad. Just add ".LOG" on the first line (without the quotation marks) and every time you open such a document, the editor will append the current date and time (as specified by your computer's clock). There is an option that lets you configure the format for the date and time text that is pasted with the corresponding command. The [Date and Time filters](#) topic gives you all the codes you can use to build a filter.

Menu Commands
Dialog Boxes
Tools
Other Features

File Menu
Template Submenu
Edit Menu
Search Menu
View Menu
Modify Menu
Lines Submenu
Block Submenu
Text-Case Submenu
Spaces Submenu
email Submenu
Document Menu
Outline-Headings Submenu
Favorites Menu
Organize Submenu
Tools Menu
Clipbook Submenu
Clipbook Properties Submenu
Quick List Properties Submenu
Help Menu

File Menu

New

Creates a new document. The new name format can be customized by changing the "New Name Format" field on the General tab of the Options dialog box.

New Outline

Creates a new Outline-type document. This command is not available in the Freeware version of NoteTab Std (unless the "View | Commercial Features" menu command has been activated).

Open

Opens an existing document. If the file uses the UNIX or Mac format, it is automatically converted to the Windows (DOS) format. You can specify through the "Save Format" field in the Options dialog box on the Documents tab how such files should be saved. Choosing the "Original" option restores the original format when you save the document.

Reopen

Maintains a list of recently used files. Quickly reopen a recently used file. The most used ones are at the top of the list and the least used ones get gradually pushed to the end of the list. When they are pushed beyond the 16th position, they are automatically removed from the list.

Template

Save

Saves the active document. The file format used will depend on the setting in the "Save Format" field in the Options dialog box on the Documents tab. Choosing the "Original" option restores the original format when you save the document.

Save As

Saves the active document with a new name. The file filter does not influence the output format. It is only useful for displaying the corresponding files in the Explorer window.

Save All

Saves all the modified documents. This command also applies to modified Clipbook libraries when the [Clip] editor is open.

Export

Saves the text selection or the active document in a different format: ASCII/ANSI, DOS, UNIX, EBCDIC, or Mac. Using this command on an Outline document will produce a normal text file from it. By default, headings are underlined. This setting can be turned off through the "Underline Headings" field in the Options dialog box on the Outline tab.

Close Document

Closes the active document.

Close All Documents

Closes all the documents.

Printer Setup

Changes printer and/or printer options.

Print

Prints the active document or selected text.

Print All

Prints all the open documents.

Send

Sends the document through electronic mail (MAPI). If your e-mail program supports MAPI (for example Eudora), you can configure it to act as your MAPI server. Check your software's Help file or manual to see how you can make it the default server. If you get an error message when you try to use the Send command, check that you have "Microsoft Exchange" properly installed from the Windows install CD.

Exit

Quits the application; prompts you to save modified documents.

Template Submenu

This submenu can be found under the File menu.

New

Creates a new document template.

Open

Creates a new document based on the chosen template.

Edit

Opens a document template for editing.

Add Header

Adds a template header to the current document. This makes it easy to convert any document into a reusable template.

Edit Menu

Undo

Reverses the last edit action. Undo is not available across Outline headings and only applies to text operations in a document.

Redo

Redo the previously undone action. This command is only available in [NoteTab Pro](#).

Reload

Reloads last saved version of the current document. This operation cannot be undone!

Reload All

Reloads last saved version of all modified documents. This operation cannot be undone!

Cut

Cuts the selection and puts it on the Clipboard.

Copy

Copies the selection and puts it on the Clipboard.

Paste

Inserts Clipboard contents.

Paste New

Creates a new document and inserts the Clipboard contents.

Delete

Erases the selection.

Clear All

Erases the selection or the entire document if no text is highlighted.

Select All

Selects the entire document or the content of the current Outline heading.

Copy All

Copies the entire document and puts it on the Clipboard. With Outline documents, you are prompted if you want to copy the whole outline or just the current topic.

Search Menu

Find

Finds the specified text. The search criteria can represent multiple lines.

Replace

Replaces specific text with different text. The search and replace criteria can represent multiple lines.

Count Occurrences

Counts occurrences of search-text criteria.

Search Again

Repeats the last search.

Search Reverse

Repeats the last search in reverse direction.

Quick Find

Finds the next occurrence of the highlighted text. Note that this command uses the current settings of the Find dialog box (search direction/match case/whole word/scope)

Search Disk

Finds/replaces the specified text in files on the disk. The replace feature is not available in the Freeware version of NoteTab Std (unless the "View | Commercial Features" menu command has been activated).

Set Bookmark

Sets or turns off bookmark at cursor position. This submenu is only available in NoteTab Pro. Currently, bookmarks cannot span across Outline topics and are lost when you move to another topic.

Go to Bookmark

Moves cursor to the specified bookmark. This submenu is only available in NoteTab Pro.

Go to Tab

Activates the specified document tab.

Go to Line

Jumps to the specified line. In NoteTab Pro you can jump to a paragraph by preceding the number with a letter "p".

Go to Hyperlink

Opens the document or heading corresponding to hyperlink under the cursor.

Match Brackets

Finds the matching bracket. Place the cursor inside the bracket area and NoteTab will select the brackets and all the text between them. The algorithm that performs the search has been optimized for speed. Under some rare situations it will not find a matching bracket. However, if you select one of the brackets before selecting this command, the search will always return the correct result.

View Menu

Toolbar

Shows or hides the main toolbar.

Clipbar

Submenu commands to create, open, or hide the [Clipbar](#).

Ruler

Shows or hides the document ruler.

Line Numbers (*only NoteTab Pro*)

Shows or hides the line-number ruler.

Libraries Bar

Shows or hides the Clipbook libraries button bar above the status bar.

Status Bar

Shows or hides the Status bar, which displays hint information and the cursor coordinates.

Nonprinting Text (*only NoteTab Pro*)

Shows or hides nonprinting characters like paragraph marks, spacebar and tab characters, and page breaks.

Stack Document Tabs

Shows document tabs stacked or unstacked.

Multiple Instances

Allows multiple instances of NoteTab to run.

Icon in System Tray

Minimizes NoteTab to the system tray instead of the task bar.

Stay-on-Top

Makes NoteTab stay on top of other applications.

Commercial Features (*only NoteTab Light*)

Use this command to enable some of the commercial features for a [trial period](#) of one month.

Printing Options

Sets printing options (margins, font, etc.).

Spell-Checker Options

Sets spell-checker options (dictionaries, correction preferences, etc.). This command is only available in the commercial version of NoteTab. To keep the distribution file size small, the commercial package does not include any dictionaries. However, you can download the thesaurus file and dictionaries for several languages for free by visiting one of the [NoteTab Web sites](#).

Options

Sets program options.

Modify Menu

Lines

Alignment

Changes the alignment (left/center/right/justify) of selected paragraphs based on the size of the document window or column wrap. This feature only works correctly with fixed-width fonts. The "justify" command is not available in the Freeware version of NoteTab Std (unless the "View | Commercial Features" menu command has been activated).

Block

Text Case

Spaces

Strip HTML Tags

Use this command to remove all HTML tags from selected text or the whole document. Control and other special characters are converted during this process. This command also attempts to format text in tables by separating each cell with a tab character and by placing each row on a separate line. Use the submenu command "Preserve URLs" if you want to keep links.

Change HTML Tags

Use the submenu commands to convert HTML tags in your text selection or the whole document to uppercase, lowercase, or XHTML format.

Document to HTML

Converts highlighted text or the active document to HTML format. Special characters are converted to the HTML equivalent tag. Choose the "No Paragraph Tags" command if you do not want NoteTab to insert <P></P> and
 tags when it encounters new lines. Choose the "No Character Conversion" command if you do not want extended and special characters converted to HTML tags. If you apply either of these commands to the whole document, NoteTab will create a new HTML page with the header and footer defined in the Options dialog box on the Internet tab. Outline documents are converted complete with a table of contents and links between the topics.

Characters to HTML

This command will convert all extended and special characters (e.g. NoteTab Std) it encounters in the selected text (or whole document if no text is selected) to their corresponding HTML syntax. Choose "Extended Characters" if you do not want tag elements such as <, >, and " to be converted.

E-mail

Convert to Ascii/Ansi text

Converts document text from ASCII to ANSI character set or reverse. Only extended characters are affected by this command.

Lines Submenu

Line commands affect entire lines or word-wrapped paragraphs. This submenu can be found under the Modify menu.

Sort

Sorts the selected lines or the entire document in ascending or descending alphanumerical order. Use the settings "Case Sensitive Sorting" and "Sort Removes Duplicates" on the Tools tab in the Options dialog box to control the result. Depending on the number of lines to sort and the amount of available RAM, this procedure may take quite long!

Join Lines

This command joins the current line to the next, or all selected lines together into a single line. Tabs and spaces at the beginning or end of such lines are removed. Blank lines are left untouched.

Split Lines

This command splits a word-wrapped line into individual lines. If word wrap is not enabled, this function will determine the length of each line based on the width of the editor window.

Reformat Lines

This command lets you define a left margin and text width (in characters), and reformats the selection or the current line by inserting hard breaks at the line boundaries and indenting the whole block. An option lets you determine whether the text is left and right justified. This command is not available in the Freeware version of NoteTab Std (unless the "View | Commercial Features" menu command has been activated).

Adjust Line Breaks

Use this command on documents (for example UNIX or Mac files) that do not display lines with standard *carriage return + line feed* characters.

Compress

Documents received from Web browsers often have a blank line between every normal line, producing a double-spacing effect. Use this command to remove this double spacing.

Trim Blanks

This command removes trailing blank and tab characters from the end of the current line or selected lines.

Indent

The indent command adds a specified number of blanks or tabs at the beginning of the current line or all selected lines. By default, one blank character is added each time you invoke this command. You can choose to use blank characters or tabs, or specify a larger number of indent characters by changing the settings on the Tools tab in the Options dialog box.

Unindent

Unindent does the exact opposite of the Indent command.

Insert Bullets

Inserts bullets at the beginning of each line. The bullet character can be changed by editing the "Bullet Character" field on the General tab in the Options dialog box.

Number Lines

Inserts numbers at the beginning of each line. NoteTab will prompt you for the starting number. The number separator can be changed by editing the "Number Separator" field on the General tab in the

Options dialog box.

Block Submenu

Although NoteTab is not capable of drawing rectangular selections, it determines the block coordinates based on the selection's start and end coordinates. These commands are not designed to operate on word-wrapped lines. Currently, NoteTab treats tabs as a single character, so tab size is not taken into account – this feature may be improved in a future version of the program. This submenu can be found under the Modify menu.

Cut

Cuts the block selection and puts it on the Clipboard. A dialog box will prompt you for the block dimensions if no text has been selected. Select the Modify | Block | Paste menu command to paste the block of text back into NoteTab (using the normal Paste command will not insert the block in columnar format).

Copy

Copies the block selection and puts it on the Clipboard. A dialog box will prompt you for the block dimensions if no text has been selected. Select the Modify | Block | Paste menu command to paste the block of text back into NoteTab (using the normal Paste command will not insert the block in columnar format).

Paste

Inserts Clipboard contents at the column position (determined from the cursor position).

Delete

Erases the block selection. A dialog box will prompt you for the block dimensions if no text has been selected.

Sum

Adds numbers in the selected columns.

Text-Case Submenu

Except as noted below, case commands only apply to selected text. This submenu can be found under the Modify menu.

Invert Case

Inverts the case of all the selected characters. (NoteTab Pro: if no selection, the character at the cursor position.)

Upper Case

Converts all the selected characters to upper case. (NoteTab Pro: if no selection, the character at the cursor position.)

Lower Case

Converts all the selected characters to lower case. (NoteTab Pro: if no selection, the character at the cursor position.)

Capitalize

Converts the first character of all selected words to upper case and the rest of the word to lower case. If no selection, the word at the cursor position.

Sentence

Capitalizes the first character of each sentence. All other characters are changed to lower case. If no selection, the word at the cursor position.

Spaces Submenu

Spaces commands affect either selected text or the whole document if no text is selected. This submenu can be found under the Modify menu.

Single Space

Use this command to replace groups of multiple adjacent blank spaces to single spaces.

Single Tab

Use this command to replace groups of multiple adjacent blank spaces to single tabs.

Tabs to Spaces (*NoteTab Pro only*)

Converts tab characters to space characters. The number of characters added depends on the size of the tab.

E-mail Submenu

These commands affect either selected lines or the current line if no text is selected. This submenu can be found under the Modify menu.

Quote Text

Adds e-mail quote character(s) at the beginning of lines. The default quote character is ">". It can be changed by editing the "E-mail Quote" field on the Internet tab in the Options dialog box.

Unquote Text

Removes all ">" e-mail quote characters from the beginning of lines. This command does not take into account the setting in the "E-mail Quote" field on the Internet tab in the Options dialog box!

Document Menu

Properties

Sets properties for the active document. Changing these properties turns off the "Default Properties" setting. As a result, changes made to the Document settings in the Options dialog box will not affect the current document.

Default Properties

Use default settings in active document as defined in the Options dialog box on [Documents tab](#).

Disk File Properties

Shows the Windows "File Properties" dialog box for the current document.

Auto Indent

Enables or disables automatic indentation. Changing this option turns off the "Default Properties" setting.

Word Wrap

Sets or turns off word wrap in active document. Changing this option turns off the "Default Properties" setting.

Update Column Wrap

Updates lines in current document so that they wrap at defined column. This option is only available in [NoteTab Std](#) and is enabled when the column wrap option is set for the current document. [NoteTab Pro](#) does not need this feature because it handles column wrap in real time.

Font

Change the current document font, or increase/decrease its size. Using these commands turns off the "Default Properties" setting.

Insert File

Inserts the content of the specified file at the cursor position.

Insert Date/Time

Inserts current date and time at the cursor position. The date format is determined by the Date filter in the Options dialog box on the [Tools tab](#).

Insert Document Name

Inserts active document name at the cursor position.

Insert HTML Color

Inserts chosen HTML color code at the cursor position.

Insert HTML Link/Image

Insert HTML links and images by way of the File Open dialog box. If text is highlighted when the command is chosen, the tags wrap around it. Note that the File Open dialog box always points to the last used directory.

Insert Page Break

Inserts a page break character at the current line.

Outline Headings

This submenu is not available in the Freeware version of NoteTab Std (unless the "View | Commercial Features" menu command has been activated).

Second Window

Shows active document in a second fixed window pane. You can resize the windows by dragging the splitter bar between them. The tab representing the left window is labeled "1." and the one for the right window "2.". Clicking on the tabs of other documents will display them in the left window. Pressing Ctrl+Shift+W in the left window swaps them around. Pressing Ctrl+Shift+W in the right window toggles off the Second Window feature. Note that showing Outline documents in the second window does not change the position of the Outline-headings list.

Use as Paste Board

Makes the active document capture all Clipboard text cut and copy operations. New text is appended to the end of the document.

Rename

Renames the active document. The disk file is also renamed if it exists.

Delete

Deletes the active document. Erases corresponding disk file if it exists (the deleted file goes to the recycle bin).

Read-Only

Sets or turns off the read-only status of the active document. When a document is "read-only", you cannot modify its content. As a result, many menu and toolbar commands are not available.

Outline-Headings Submenu

This submenu can be found under the Document menu. It is not available in the Freeware version of NoteTab Std (unless the "View | Commercial Features" menu command has been activated).

Add Heading

Adds a new outline heading.

Make Heading

Creates a new outline heading based on the selected text in the active document.

Remove Heading

Deletes the active outline heading.

Edit Heading

Lets you edit the current outline heading.

Sort Headings

Sorts outline headings in ascending order.

Move Next

Moves to the next heading.

Move Previous

Moves to the previous heading.

Move First

Moves to the first heading.

Move Last

Moves to the last heading.

Favorites Menu

Categories

The "New Category" command creates a new Favorites category list. All available categories are listed below that command and a checkmark highlights the current category. Choose the "Make Default" command to open the category in the main Favorites menu and to make it the default in the Quick List. You can rename or delete the active category by choosing the command "Rename Category" or "Delete Category" respectively.

Organize

Items in the Favorites list can represent document names (including wild cards: * and ?), directory names, and other categories. Documents are opened and selected when you pick them from the list. If you choose a directory name, the File Open dialog box is opened with the directory's content displayed.

You can also use the Favorites menu to save documents in a specific directory, or with a specific name (no wildcards). Just hold the Ctrl key down when you select a Favorites item; this will open the Save As dialog box set to the specified directory and document name.

Organize Submenu

This submenu can be found under the [Favorites menu](#).

Add Document to Favorites

Adds active document name to the active Favorites list.

Add all Documents

Adds all open documents to the active Favorites list.

Add Path Name to Favorites

Adds path name of active document to the active Favorites list.

Go to Favorites

Shows the [Favorites dialog box](#).

Tools Menu

Clipbook

Clipbook Properties

Expand Text

Replaces text at the cursor position with matching Clipbook replacement text. This command can save you a lot of time when you often reuse same bits of text.

Auto-replace Mode

Replaces typed text with matching Clipbook item. This option is automatically set when you open an "auto-replace" library. When this mode is enabled, NoteTab will check for a matching item in the active library whenever you press any of the following characters (Clip activation keys):

TAB SPACE RETURN !" , . : ; = > ?] }

If a matching Clip is found, the typed item in the document is automatically replaced according to the Clip instructions.

Quick List

Shows or hides the Quick List window.

Quick List Properties

Open File at Cursor

If you place the cursor on a file name, NoteTab will try to open it. If NoteTab doesn't have enough information to find it, you will be prompted to complete the name.

Launch Document

Launches active document in application defined in the "Custom Launch" field on the Tools tab in the Options dialog box. This command is disabled if no application is defined. You can, for example, use this feature to easily launch the current document in a compiler or syntax checker.

Tidy HTML Code

Sends the text from the current document to HTML Tidy -- a free utility that cleans HTML code and fixes errors. The resulting code is then inserted back into the document.

HTML Tidy accepts a variety of switches to control the way it processes HTML tags. NoteTab will try to determine the best switches based on the active document type and the HTML Tag option settings. For example, if Auto-indent mode is enabled in the current document, the HTML code will be formatted with indenting. The tag format (uppercase/lowercase/XHTML) is determined by the settings on the HTML Files tab in the Options dialog box. If you want more control over the settings, you can create a configuration file for Tidy, which you should save in the NoteTab folder with the name Tidy.cfg (see the program's instructions for details). You can learn more about HTML Tidy and download the latest version by visiting the following Web page:

<http://www.w3.org/People/Raggett/tidy/>

The easiest setup is to copy the Tidy.exe program file to the NoteTab folder, but you may place it anywhere else if you prefer.

Validate HTML

This command is only available if NoteTab finds CSE HTML Validator Pro (v2.00A and higher) -- a program for checking HTML syntax errors. When you click on this command, it will send the current document text to CSE HTML Validator. Once the checking is completed, NoteTab will create a Clipbook library containing all the messages produced by the CSE program. Executing the Clips in the library will select the offending line and display the corresponding error message. Note that you should avoid

removing or adding lines while correcting errors so that the line numbers referenced by the Clips remain valid. You can learn more about CSE HTML Validator and download the latest version by visiting the following Web page:

<http://www.htmlvalidator.com/>

View in Browser

Shows the active document in the default Web browser.

Other Browser

Shows the active document in user-defined Web browser. The browser application is defined in the "Second Browser" field on the Internet tab in the Options dialog box. This command is disabled if no browser is defined.

Open Link

Opens a link in your default Web browser.

Quick Open Links

When set, you do not have to hold down the Ctrl key to activate a hyperlink with the mouse.

Auto Scroll

Starts automatic scrolling for hands-free reading of the active document.

Calculate Expression

Calculates the value of the selected mathematical expression. If the expression ends with an equal sign (=), the result is pasted just after it. If the document is read-only or if the expression does not end with an equal sign then the result is displayed in a message box.

Text Statistics

Shows detailed character and word count for the selected text or whole document.

Spelling

Checks the spelling in the active document. Checking starts from the cursor position unless it is at the end of the document. If text is selected, spell checking is applied to the selection only. You cannot spell check a document that is read-only. To keep the distribution file size small, the commercial package does not include any dictionaries. However, you can download dictionaries for several languages for free by visiting one of the NoteTab Web sites. The spell checker is only available in the commercial versions of NoteTab.

Thesaurus

Checks the thesaurus for synonyms of the word at cursor. This command is only available in the commercial versions of NoteTab. Due to its size, the commercial package does not include the thesaurus dictionary file. However, you can download it for free by visiting one of the NoteTab Web sites.

Clipbook Submenu

This submenu can be found under the [Tools menu](#).

Open Clipbook

Opens or closes the [Editor Clipbook](#). May remain hidden if configured that way.

Add New Clip

Adds a new Clip to the current library. If the Clipbook is closed, you will be prompted to choose which library should receive the new Clip.

Add from Clipboard

Adds a new Clip from the Windows Clipboard to the current library. If the Clipbook is closed, you will be prompted to choose which library should receive the new Clip.

Paste Current Item

Pastes the content of the currently selected Clip into your document.

Copy Selection

Copies the selected text in your document to the Current Clipbook library.

Cut Selection

Cuts selected text in your document to the Current Clipbook library.

Clip Editor

Shows the content of currently selected Clipbook item in the clip editor (**[Clip]** document tab). Note that you cannot use the Clipbook to paste or launch Clips when the Clip editor is the active document.

New Library

Creates a new Clipbook library and opens it as a standard document. Check the [Clipbook Libraries](#) topic for information on the format.

Open Library

Opens a Clipbook library as a standard document. Check the [Clipbook Libraries](#) topic for information on the format.

Clipbook Properties Submenu

This submenu can be found under the Tools menu.

Dock Left Side

Docks Clipbook window on the left side of document.

Dock Right Side

Docks Clipbook window on the right side of document.

Hide Clipbook

Hides the Clipbook window. This gives more space for the document window but keeps Clip items accessible through the In-Context Clipbook and the "Expand Text" function.

Font

Sets the font and text color for the Clipbook window.

Background Color

Sets the background color for the Clipbook window.

Match Document Font

When enabled, the Clipbook font changes to match the font used in the selected document.

Show Headers

Set this option to display Clipbook headers in the Clip editor window.

Quick List Properties Submenu

This submenu can be found under the Tools menu.

Show Full Name

Shows file names with their path or without. The full file name is displayed in the hint bar when you move the mouse cursor over the list.

Font

Sets the font and text color for the Quick List window.

Background Color

Sets the background color for the Quick List window.

Documents

Shows a sorted list of all open document names in the Quick List window.

Favorites

Shows the content of the active Favorites list in the Quick List window.

Disk Files

Shows disk files matching specified criteria in the Quick List window.

Clip Assistant

Shows the Clip Assistant window. When you activate this command, a list of ready-to-use syntax elements is displayed in place of the Quick List. You can paste the items into your document by double-clicking on them or by dragging them into your document. You can edit the ClipHelp.CLH file to customize the content of the Clip Assistant list.

Help Menu

Help Topics

Opens the Help Topics window.

Help on Clip Programming

Opens the Help topic on Clip programming.

Search for Help On

Searches for a help topic.

Show Tip of the Day

Shows the Tip-of-the-Day dialog box.

NoteTab Glossary

Shows NoteTab's "Glossary.otl" file. Provides definitions for all the technical vocabulary used in the documentation of NoteTab.

Readme File

Shows NoteTab's "Readme.txt" file.

What's New

Shows NoteTab's "WhatsNew.txt" file. This file contains information about the latest changes made to the NoteTab editor.

HTML Help

NoteTab makes the popular "The HTML Reference Library" (by Stephen Le Hunte) available from the Help menu. If you have HTMLLib version 3.0 or older, you can install the package in the same directory as NoteTab. Another method that works with all versions is to create a shortcut* to the HTMLLib.exe file in the NoteTab directory. The highly recommended "HTML Reference Library" is available from:

<http://www.htmlib.com/>

This menu command is only available once the "HTML Reference Library" is correctly set up.

(* to create the shortcut: 1. open Explorer and select the HTMLLib file; 2. press Ctrl+C to copy it to the Clipboard; 3. select the NoteTab directory and choose the command "paste Shortcut" from the Edit menu. The menu command will be displayed in NoteTab after you restart it.)

HTMLLib Color Wizard

This command activates the "HTMLLib Color Wizard" utility developed by Stephen Le Hunte. This menu command is only available once the "HTML Reference Library" is correctly set up (see above).

Replace MS Notepad

This feature is not available under Windows 2000. Once this command has been activated, it causes NoteTab to execute whenever MS Notepad is invoked. When you choose that command, NoteTab will first rename Notepad by adding the extension ".MS" to the program file. Then, it will copy a special little program called NOTEPAD.STB to the Windows or System directory. Whenever this program is executed, it launches NoteTab instead of Notepad. If you want to restore MS Notepad after this operation, just use the command "Restore MS Notepad", which is also available under the Help menu. Note that Windows 2000 does not allow substituting MS Notepad with another program.

Restore MS Notepad

Reverses the changes made by the "Replace MS Notepad" command. This command is only available after you have used the "Replace MS Notepad" command.

Order NoteTab

This command connects you to the order page on the Web. It is only available in NoteTab Light and the trial version of NoteTab Pro.

About

Displays the NoteTab "About" dialog box.

Export Dialog Box

Print Dialog Box

Find & Replace Dialog Boxes

Regular Expressions

Search Disk Dialog Box

Block Dimensions Dialog Box

Reformat Lines Dialog Box

Favorites Dialog Box

Text Statistics Dialog Box

Document Properties

Customizing the Toolbar

Options Dialog Box

Export Dialog Box

Use the Export dialog box to save the current document or a selection of text to a new file in a different format.

The Output field is only available when some text is selected in the document. You can choose to save just the selection or the entire document to the new file.

The format field lets you choose how line markers are defined. You have a choice between the standard Windows format (DOS), UNIX, EBCDIC, or Mac. If you choose the EBCDIC format, you can define through the LRECL field whether you want to produce a file with variable line width or fixed width.

Use the ASCII character set if your document contains extended characters (e.g. accented letters) and you want to read it with a DOS program.

Check the "Hard Wrap" checkbox if you want long lines that are word-wrapped saved with carriage returns where the lines break.

Click on the Continue button to open the File Save As dialog box and export the text.

Print Dialog Box

Most settings in the Printer dialog box are saved and restored when you run the program again. However, each time you move to another document, the Title text is replaced by the document's name and Print Word Wrap is set.

If you choose to print a document to a file, the lines will be wrapped at the same point as if they were printed on paper and page breaks are inserted accordingly. The resulting file has the same name as the document but has the extension PRN; it is created in the same directory as the document.

By default, the document font will be used for printing. You can, however, choose a printer font that is used instead of the document font. This option is available on the Font tab.

Find & Replace Dialog Boxes

If the selected text is between 2 and 255 characters long and the "Find Word at Cursor" option is set, it is placed in the Find field of the dialog box when it is opened (special characters are automatically replaced by their respective tokens – see below). Otherwise, the last searched item is displayed in the field. Searching always starts from the cursor position, unless you specify "All" in the Search field! If the scope is limited to the current document, selected text is more than 30 characters and at least twice as large as the search criteria, then Find/Replace will only search within the selected text.

Double-clicking in an empty area of the dialog box will move the cursor to the beginning or the end (depending on the search direction) of the current document. The Find/Replace tool will ask you if you want to resume a search from the beginning of a document if no match is found. Note that you can set the font of the Find/Replace fields in the corresponding dialog boxes: right click on the Find or Replace field in the dialog box to open the shortcut menu with the Font command.

Put a checkmark in the Regular Expression checkbox if you want to define the search criteria as a regular expression.

Tip: You can quickly copy text from the Find field to the Replace field by double-clicking on the "Replace with" label.

The Find and Replace dialog boxes accept tokens to represent special characters:

- ^T = Tab (\$09)
- ^B = Page break (\$0C)
- ^P = Paragraph (carriage-return/line-feed pair)

The following are not available in NoteTab Pro:

- ^C = Carriage-return (\$0D)
- ^L = Line-feed (\$0A)

If you actually need to search characters that correspond to one of the tokens, just precede it with an extra ^ symbol (for example ^^T to search for ^T).

Note that these tokens are not applicable when you perform regular expression searches.

When the editor is configured to "stay-on-top", the Find and Replace dialog boxes are modal. In other words, you cannot return to the editor until you close the dialog box. When the program is not "stay-on-top", they are modeless (you can edit documents while the dialog boxes are open).

You can increase the width of the dialog box if necessary and save the new size by opening its Control (or System) menu and clicking on Save Window State.

Put a checkmark in the "Count Occurrences" checkbox if you don't want to replace text but just want to count the number of occurrences of the specified criteria.

Regular Expressions

The Find and Replace dialog boxes support search criteria with basic regular expressions. To enable this mode, just put a checkmark in the "Regular Expression" checkbox. You can also specify regular expressions with the Find and Replace Clip commands by using the "R" option.

Those familiar with UNIX text utilities will find that the engine is somewhere between EGREP and AWK with regard to the complexity of problems that it can solve.

Find Patterns

Regular expression patterns are composed of the following:

Period (.)

Matches any single character except newline. A newline (internally) is really two characters in a specific order -- <carriage return> followed by <linefeed>. To match a newline, you must always explicitly specify a newline.

Caret (^)

Matches at the beginning of a line only. A ^ occurring ANYWHERE in the match expression (except within a character class) is interpreted in this manner. This allows meaningful use of ^ in combination with grouping or alternation (see below).

Dollar sign (\$)

Matches at the end of a line only. As with ^ the \$ character retains its special meaning anywhere within the expression (except in a character class).

Backslash (\)

Followed by a single character matches that character. For example, * matches an asterisk, \\ matches a backslash, \\$ matches a dollar sign, etc.

The following sequences have special meaning

\s	space (ASCII #32)
\t	tab (ASCII #9)
\r	return (ASCII #13)
\l	linefeed (ASCII #10)
\n	newline (#13 followed by #10)
\f	formfeed (ASCII #12)
\p	pipe character
\w	any word delimiter. Matches any of \t\s!"&()*+,-./:;<=>?@[\\]^`{ }~
\W	any nonword delimiter. Equivalent to [^\t\s!"&()*+,-./:;<=>?@[\\]^`{ }~]
\h	any hex character. Matches any of 0123456789ABCDEF
\H	any nonhex character. Equivalent to [^0123456789ABCDEF]
\a	any character, including carriage return, line feed, form-feed, etc.
\b	any blank (white) space including space, tab, form-feed, etc. Equivalent to [\s\t\f\n\r]
\B	any nonwhite space character. Equivalent to [^\s\t\f\n\r]
\d	any digit character. Equivalent to [0-9]
\D	any nondigit character. Equivalent to [^0-9]

Case is ALWAYS significant when using the special characters. Thus \s matches a space while \S matches a capital letter S.

A single character not otherwise endowed with special meaning matches that character. Thus z matches

a single instance of the letter z.

A string enclosed in brackets [] specifies a character class. Any single character in the string is matched. For example, [abc] matches an a, b, or c. Ranges of ASCII letters and numbers can be abbreviated as, for example, [a-z0-9]. If the first symbol following the [is a caret (^) then a negative character class is specified. In this case, the string matches all characters EXCEPT those enclosed in the brackets. For example, [^a-z] matches everything except lower case characters (and newlines).

The special characters defined above may be used inside of character classes with the exception of \n, \w and \h, which are shorthand for their own character classes. If the characters - or] are to be used literally inside of a character class, they should be preceded by the escape character \. Note that *?+(){}!^\$#& are not special characters when found inside a character class.

Seeking Closure

A regular expression followed by * matches zero or more matches of the regular expression. This is referred to as a closure. Thus ba*b matches the string bb (no instances of a), bab (one instance), or baaaaaab (several instances).

A regular expression followed by a + matches one or more matches of the regular expression. This is another type of closure. In this case ba+b will not match bb, but it will match bab, or baaaaaab.

A regular expression followed by a ? matches zero or one matches of the regular expression. This is another closure. Here, ba?b will match bb or bab, but not baaaaaab.

Concatenated Expressions

Two regular expressions concatenated match a match of the first followed by a match of the second. Thus (abc)(def) matches the string abcdef.

Alternation

Two regular expressions separated by | match either a match of the first or a match of the second. This is referred to as alternation. Any number of regular expressions can be strung together in this way. Alternation matches are tested in order from left to right, and the first match obtained is used. Then the remaining alternate expressions are skipped over.

Grouping Expressions

A regular expression enclosed in parentheses () matches a match of the regular expression. Parentheses are used to provide grouping, and may be nested to arbitrary depth. Open and close parentheses must be balanced. For example, the following two expressions are not equivalent, and the second probably expresses what was intended:

PROCEDURE|FUNCTION

(PROCEDURE)|(FUNCTION)

The first expression is equivalent to

PROCEDUR(E|F)UNCTION

The second expression matches either of the two words.

Tagged Matches

A regular expression enclosed in curly braces {} forms a tagged match word. Whatever was matched within the braces may be referred to by a Replace expression in a manner to be described. Tagged match words may not be nested. Open and close braces must be balanced. A maximum of nine tagged match words can be referenced by the Replace expression. For example, consider the expression

`b{a*}b`.

If the string being tested is "bab", then the tagged match word contains a single "a". If the string being tested is "baaaaaab", then the tagged match word contains "aaaaaa". If the string tested is "bb", then the tagged match word is empty.

Order of Precedence

Regular expressions are interpreted from left to right. The order of precedence of operators at the same parenthesis level is [], then *+!, then |, and then concatenation.

Tag braces are interpreted strictly from left to right and do not control precedence in any way. The first tagged match word found is given a tag of 1, the second a tag of 2, and so on up to a maximum tag of 9. The tag number that each word receives is based on when it is encountered in the line. If tags are skipped over as a result of alternation, then any remaining tags in a line receive shifted tag numbers. For example, consider the expression:

`(FUNCTION)|(PROCEDURE)\s+{[^s(]+}`

If a line contains the word PROCEDURE then the word following PROCEDURE has a tag number of 2. If a line contains the word FUNCTION, then the word following FUNCTION has a tag number of 1. It is up to the user to take advantage of this behavior. Generally, it is good practice to surround an entire set of alternates with tag markers:

`{(FUNCTION)|(PROCEDURE)\s+{[^s(]+}`

Replace-with Patterns

Replace regular expressions are constructed the same way as Find regular expressions, but the number of operators is reduced. The replacement process occurs in the following manner:

The Find expression finds a string of text that starts at the leftmost position in the input line that matches, and continues to the rightmost position that matches. The string of matched text is operated upon by the Replace expression.

Replace expressions are composed of the following:

Single character

A single character not otherwise endowed with special meaning is treated literally.

Backslash (\)

Followed by a single character matches that character. For example, "*" matches an asterisk, "\\\""

matches a backslash, "\\$" matches a dollar sign, etc. A "\" followed by a single character treats that character literally. In this way a "&" writes an ampersand and "\\" writes a backslash.

The following sequences have special meaning:

\s	space (ASCII #32)
\t	tab (ASCII #9)
\r	return (ASCII #13)
\l	linefeed (ASCII #10)
\n	newline (#13 followed by #10)
\f	formfeed (ASCII #12)
\z	null expression

Another special case occurs when "\" is followed by a single digit in the range of 1 through 9. In this case the tagged match word found by the Find expression is used in the resulting replacement text. If a tagged match word for that tag number was not defined, or if the tagged match word doesn't match anything, then nothing is output. The tagged match words can be used in any order and can be repeated any number of times.

An ampersand ("&") appearing in the Replace expression causes all text matched by the match expression to be sent to the output. The ampersand can appear in the Replace expression as many times as desired.

Examples:

The following examples use the NoteTab Replace dialog box (press Ctrl+R to open it). Make sure you tick the "regex box" before trying them.

Changes all H2 tags to H3:

Find: `<H2>{.*}</H2>`

Replace with: `<H3>\1</H3>`

Strips trailing blanks from each line:

Find: `^{\.+[\s]}s*`

Replace with: `\1`

Places each encountered word on a single line (Replace All can take quite long on big files!):

Find: `\w*{["$#A-Z0-9+]}\w*`

Replace with: `\1\n`

Converts all encountered e-mail addresses to HTML Mailto links:

Find: `[A-Z_.-0-9]+@[A-Z_.-0-9]+`

Replace with: `&`

To learn more about regular expressions, go to <http://www.perl.com> and click the "documentation" link.

Search Disk Dialog Box

Use the "Search Disk" command (Ctrl+D) to do a Search/replace operation on your disk files instead of the documents in NoteTab. The advantage of this tool is that you do not need to open all the files you want to process – you save both time and system resources.

The dialog box has two tabs that let you choose whether you want to do a search or a replace operation. Note that the "Replace in Files" tab is not available in the Freeware version of NoteTab Std (unless the "View | Commercial Features" menu command has been activated).

The Find and Replace fields accept tokens to represent special characters. These are:

- ^T = Tab (\$09)
- ^B = Page break (\$0C)
- ^P = Paragraph (carriage-return/line-feed pair)
- ^C = Carriage-return (\$0D)
- ^L = Line-feed (\$0A)

If you actually need to search characters that correspond to one of the tokens, just precede it with an extra ^ symbol (for example ^^T to search for ^T). Note that you can set the font of the Find/Replace fields: right click on one of the fields to open the shortcut menu with the Font command.

Note that these tokens are not applicable when you perform regular expression searches.

Use the File Masks field to specify the directory that contains the files you want to search. Use standard wildcards for the file name to limit the search to matching files; you can specify multiple file criteria by separating them with a semi-colon ";" (e.g. C:\NoteTab*.txt;*.otl;*.htm*). Put a checkmark in the "Include Subdirectories" field if you want to expand the search to all the subdirectories. By default, the procedure will not search through hidden files unless you put a checkmark in the "Hidden Files" box.

Put a checkmark in the "Find First" field if you want the search operation to stop as soon as a match has been encountered. This option is only available if you are doing a "Find in Files" operation.

Put a checkmark in the "Backup" field if you want backup copies of files modified by the replace operation.

When you enable the Confirm option, matching files are displayed in a list box. You can then select which files you want to open in NoteTab when you are doing a find operation. If you are doing a replace operation, you can select which files should be processed; you can also open the selected files in NoteTab if you prefer. If the replace operation encounters binary files, it will display them in the list box even if you haven't set the confirm option.

Put a checkmark in the Reg. Expr. checkbox if you want to define the search criteria as a regular expression. **Caution:** The regular expression tokens that refer to line breaks expect these to correspond to the Windows text-file format (i.e. carriage-return/line-feed pair). These tokens will not work correctly when you search through UNIX and Mac file.

You can increase the size of the dialog box if necessary and save the new size by opening its Control (or System) menu and clicking on Save Window State.

Note that read-only files are ignored when the Replace operation is chosen.

Block Dimensions Dialog Box

This dialog box is only displayed when you choose one of the "Modify | Block" commands and no text has been highlighted. It lets you determine the size of the block that you want to process. The line that will contain the top left corner of the block is defined by the cursor position. Measurements in this dialog box represent character units.

Note that the block commands are not designed to operate on word-wrapped lines. Currently, NoteTab treats tabs as a single character so tab size is not taken into account – this feature may be improved in a future version of the program.

Reformat Lines Dialog Box

This dialog box is displayed when you choose the "Modify | Lines | Reformat Lines" command. It has three fields:

- **Left Margin:** Determines the size of the margin to create on the left side of the formatted text. The size is measured in number of blank characters.
- **Text Width:** Determines the maximum width of the formatted block of text. The size is measured in characters.
- **Justify Text:** The resulting text is left and right justified when this option is set, and left justified when not set.

Currently, NoteTab treats tabs as a single character so tab size is not taken into account – this feature may be changed in a future version of the program.

This feature is only available in the commercial and trial versions of NoteTab.

Favorites Dialog Box

The Favorites dialog box makes it easy to quickly open any file or a directory (folder) that you have added to the list. You can easily add new files and directories to the list using the integrated editor or by clicking on the Add button. The Add button opens the file dialog box from which you can select multiple files. The item editor also lets you add file names with wild cards. This is handy if, for example, you want to open all your HTML files in the editor to perform a global search and replace operation:

```
C:\Web\Homepage\*.html
```

Precede the path name with a + character to also open all matching files contained in the specified directory and all its subdirectories:

```
+C:\Web\Homepage\*.html
```

By clicking on the New button, you can enter file names or directories in the input field (below the list box) without using the File dialog box displayed by using the Add button. You can also use that method to specify file names with wild cards.

When you add a directory name to the list, make sure it ends with a backslash "\" (otherwise it will be interpreted as a file name). When you open a directory item, the File Open dialog box is displayed pointing to the selected directory. If several items are selected in the list, directory items are ignored when you choose the Open or Open All button. For example, when opening the following item, the File Open dialog box will pop up and show the content of the Homepage directory:

```
C:\Web\Homepage\
```

You can create separate Favorites categories for all kinds of different tasks. The names of all available categories are displayed in the combo box above the list. Favorites categories are always stored in the program directory and have the extension FVR. These are text files that can also be edited in NoteTab like any other file.

You can also specify other Favorites categories by clicking on the New button in the dialog box, then typing a colon character (:) followed by the name of the category. For example:

```
:Favorite
```

Once you have created such an item, double-clicking on it will open all the files listed in the category. This is very useful if you want to simultaneously open files that are stored in different directories or with different wildcards.

You can reorder the content of the list by doing drag-and-drop or sort the list by invoking the shortcut menu. You can select multiple files in the list and open them by clicking on the Open button. If you want to open all the files in the list, just click on the Open All button (in this case, directory items are ignored).

The shortcut menu also lets you undo any changes you have made to the list. Changes are permanently saved when the dialog box is closed or when the category is changed. You can create new categories by invoking the combo box shortcut menu. This menu also lets you rename categories or delete them.

NoteTab can be configured to automatically open all the files listed in a Favorites category whenever you launch the program. If you want to open files from a category that is different from the one that was last used, you can specify its name from the command line.

The dialog box size and position can be saved by opening its Control (or System) menu and clicking on Save Window State.

Text Statistics Dialog Box

The Text Statistics dialog box provides information on the number of characters, words, and the number of occurrences of each different word within a text. If you select text, that information will be extracted from that portion of the document. If no text is selected, then the whole document is analyzed.

The first part of the dialog box displays the following fields:

- **Characters** The figure is based on the total number of characters encountered in the text, including spaces and tabs. Each tab counts as one character only, not the space it occupies.
- **Words** Only words composed entirely of alphabetic characters are counted here.
- **Others** Any word that is not counted in the previous field is added here. For example, numbers or words that are a mix of alphabetic characters and numbers.

To display the frequency of occurrences of individual words, click on the More button. Depending on the size of the text to analyze, this process may take some time to display the result. At the end of the list, you can read the total number of different words encountered in the text. Use the list's shortcut menu if you want to change the sorting order or copy the content to the clipboard or save it to a file.

Document Properties

This dialog box lets you set properties for individual documents. You can define word wrap, tab width, font name and size, and character set for the currently selected document. If the Default option is checked, the settings defined in the Documents tab of the [Options dialog box](#) are applied.

Word Wrap: Enable word wrap at window edge or wrap column.

Auto Indent: Use same indentation level as the last line of the previous paragraph (this is generally useful only when word wrap is off).

Trim White Spaces: Spaces at the end of paragraphs are removed when the document is saved.

Wrap to Column: Words wrap at the specified column (NoteTab Std: column wrap is not enforced when editing but is reproduced in saved file or when you choose the "Document | Update Column Wrap" menu command).

Scroll Past End of Line (only available in NoteTab Pro): Caret can move beyond the end of the current line. This option is ignored if word wrap is on.

Tab Size: Text Tab size. You can set multiple tab sizes in NoteTab Std by separating each value with a semi-colon (;). Values must be ordered in increasing sizes and represent the distance from the left margin.

Tab Units (only available in NoteTab Light/Std): Defines units used for tab size.

Type Tab (only available in NoteTab Pro): You can change the behavior of the editor by changing this option. If the tab type is set to Real, pressing <Tab> inserts a "real" tab character into the document. If the tab type is set to Fixed, pressing <Tab> inserts enough spaces to move the cursor to the next tab stop. (This is called "fixed tabs" because the position of existing text will not change if the tab size is later changed, as it does if real tabs are used.) If the tab type is set to Smart, spaces are also inserted into the document, but the positions of the tab stops are a function of the positions of the words in the previous line.

Font: Opens the Font dialog box so you can change the font used to display your document(s). NoteTab Std can show fixed- and variable-width fonts but cannot display fonts with OEM character sets (Terminal, MS Line Draw,...). On the other hand, NoteTab Pro can display fonts with OEM character sets but does not support variable-width fonts.

Save As: Saves document using original or specified file format. When "Original" is selected, the format (DOS/Windows, UNIX, EBCDIC, Mac) detected when a document is opened is used when it is next saved. Newly created documents use the DOS/Windows format when the option is set to "Original".

LRECL: This setting only applies to EBCDIC files. Use it to define or change the line record length. You can use values other than those listed.

Dos ASCII: Opens/saves document using ASCII character set. This option is only available in the Document Properties dialog box. Changing it will affect how the document is saved. If you want to convert the current text between ASCII and ANSI, use the "Modify | Convert to Ascii/Ansi Text" menu command. For example, if you convert an ASCII document to ANSI and want it to stay this way, just uncheck the DOS ASCII option; next time you save the file, it will keep its ANSI format. See the [ASCII Documents](#) topic for more details.

Use Default Settings: Uses settings defined in preferences dialog.

Document Properties are saved whenever the document or the editor is closed. All settings for up to the 256 (default value) most recently opened files are stored in a file called NOTETAB.FPR or NOTEPRO.FPR, which is placed in the program directory. You can change the number of settings that can be remembered by editing the field "Store Properties" on the Advanced tab. You can clear all memorized settings by clicking on the Clear Properties button, which is also located on the Advanced tab.

Customizing the Toolbar

Important: the toolbar buttons may not display correctly if you have an old version of Windows 95. If this is the case on your system, you will have to install a newer version of the ComCtl32.DLL file. You can do this by downloading and installing the following update file from Microsoft's Web site (about 500 Kb):

<http://msdn.microsoft.com/developer/downloads/files/40Comupd.exe>

You can configure many aspects of the toolbar. There are over 90 commands you can display in it and you can reorder any button to suit your way of working. The toolbar properties are available through the Toolbar shortcut menu (right-click with your mouse on the toolbar) or by opening the Options dialog box and selecting the Toolbar tab.

Show in Toolbar: This list shows all the available toolbar buttons. By clicking in the check box in front of each button description, you can make it visible or remove it from the toolbar. Use the Separator items to group related buttons. You can easily reorder buttons and separators by clicking on their caption and dragging them to a new position. Alternatively, you can select a button item and use the Move Up and Move Down buttons to change its position. The list's shortcut menu offers commands to move all selected items to the top of the list, to sort them in alphabetical order, and to check/uncheck them all.

Regular Buttons: When checked, the toolbar shows the standard "extruding" buttons. Otherwise, the new flat buttons (Office 97) are displayed.

Show Caption: Use this option to display captions in the buttons. The width of these buttons is automatically determined by the size of the longest caption.

Show Tool Tips: Tool Tips appear when you place the mouse cursor over a toolbar button. You can disable this feature by turning this option off.

Show Separators: If you have set the Regular Buttons option, separators add blank space between the buttons where they are placed. With the flat buttons, separators are displayed as chiseled vertical lines. If you do not want to display separators, just disable this option.

Wrap Toolbar: If this option is set and the editor window is too small to display all the buttons, the toolbar will resize itself and wrap the buttons to the next line.

Click Sound: Plays the chosen WAV file or makes PC speaker beep when buttons are clicked.

Play Sound: Sound file to play when the Click Sound option is enabled.

Button Width: Sets button width in pixels. Use 0 or leave blank to adjust width automatically.

Caption Font: This button opens the Font dialog box and lets you change the default font used for the toolbar button captions.

Options Dialog Box

This dialog box lets you configure the main program settings. Use the Options command under the View menu to open it. The settings are automatically saved when you quit NoteTab.

[Options | View](#)

[Options | General](#)

[Options | Files](#)

[Options | Documents](#)

[Options | Internet](#)

[Options | HTML Files](#)

[Options | File Filters](#)

[Options | Associations](#)

[Options | Colors](#)

[Options | Toolbar](#)

[Options | Shortcut Menu](#)

[Options | Tools](#)

[Options | Clipboard](#)

[Options | Tab Bar](#)

[Options | Outline](#)

[Options | Quick List](#)

[Options | Advanced](#)

Options | View

Show Toolbar: Displays or hides the program toolbar.

Show Nonprinting Text: Displays or hides nonprinting characters like tabs, line/page breaks, and spacebar characters. This setting is only available in NoteTab Pro.

Show Ruler: Displays or hides the document ruler.

Show Libraries Buttons: Displays or hides the Clipbook button bar (the buttons give access to the available libraries).

Show Status Bar: Displays or hides the program status bar.

Show HTML Features: Turn this setting off if you do not want to see the options and commands related to the creation of HTML documents (useful for simplifying the interface). When this setting is unchecked, it can be re-enabled later from the View menu.

Show Advanced Features: Turn this setting off if you do not want to see the more advanced options and commands in NoteTab (useful for simplifying the interface). When this setting is unchecked, it can be re-enabled later from the View menu.

Ruler Units: Sets units for measurement tick marks.

Application Icon: Changes the program icon. Click on the "Change Icon" button to display the available icons. This affects the icon displayed in the task bar, system tray, and associated files displayed in Explorer. Some of the changes will only be visible after you restart Windows.

Stay On Top: You can make NoteTab stay on top of other applications by enabling this option. When you choose this configuration, the Find and Replace dialog boxes cannot be modeless. Note that even the Help file will be covered by the application window.

Icon in System Tray: Puts NoteTab's icon in the system tray instead of the task bar. Some programs keep an icon in the system tray when you close them. When they do this, many of these programs just hide the application window but remain fully loaded in memory – this is very misleading to most users! With NoteTab, however, instead of "closing" it to the system tray you minimize it by clicking on the Minimize button. This will hide the application window and leave the icon in the system tray for easy access.

MS-Office Sounds: Enables corresponding sound events if the MS-Office (Microsoft) sounds have been installed.

Show Menu Icons: Shows the corresponding toolbar icon next to the menu item (if available).

Show Splash Screen: Shows the splash screen when you start the program.

Show Tip-of-the-Day: Shows the Tip-of-the-Day window each time you start NoteTab.

Options | General

Save Position and Size: When you enable this option, the editor will always restore its previous position and size when you start it.

Reload Open Documents: Use this option if you want the documents that were open when you closed the editor to be reopened next time you start the editor. Files that no longer exist are ignored by this feature.

Load Favorites on Startup: Loads the documents listed in the Favorite Files list (from current list or an alternative list indicated through the command line) whenever you start the editor.

Allow Multiple Instances: Set this option if you want to allow multiple instances of the editor in memory.

Check Changes on Activate: Checks if documents have been modified by other applications when focus returns to NoteTab. Turning this option off may significantly speed up the readiness of NoteTab after activation when it has floppy disk or network files open. The downside is that you are no longer warned if files have been modified by other programs.

Full Names in Favorites: Set this option to display full file names in the Favorites menu.

Clip Event Warnings: This field offers a choice of three options that determine what happens when a Clipbook library containing event headings is opened. When "Disable Clip Events" is set, Clip events such as OnOpen, OnClose, and OnClipboard are never triggered automatically. You have to launch them manually. With the "Prompt First" option, the OnOpen and OnCloseClip events are only executed if you allow them by responding to a prompt. When "No Prompting" is set, Clip events are automatically triggered without prompting you.

Close to System Tray: Minimizes NoteTab to the System Tray when the [X] button in the title bar is clicked. This option only applies when the "Icon in System Tray" is enabled.

Display Prompts: Show warning prompts when you choose the Reload and Delete commands, and when you choose the Search Disk command when NoteTab contains unsaved changes.

Blind User Mode: Autoscroll hides dialog when active and message dialog boxes make a sound when displayed. The Find dialog box will automatically close after a search operation and the Replace dialog box will close after a "Replace All" operation. When an item is selected from the Quick List, Outline, or Clipbook list, the focus remains on the list control instead of jumping back to the document. This makes it easier to use NoteTab with screen readers.

New Name Format: The name format is used when new documents are created. The code "%d" indicates where the incremental numbering is inserted in the name. In the following example, the "0.2" between the % and d tells NoteTab to format numbers smaller than 2 digits with leading zeroes: **NoName%0.2d.txt**

Bullet Character: Defines character used by the "Modify | Lines | Bullet Insert" function.

Number Separator: Defines characters used by the "Modify | Lines | Number Lines" function to separate the numbers from the text.

Options | Files

Make Backups: Makes a backup copy before updating a file on disk.

Save Files on Close: Automatically saves modified files before closing them. Note, however, that the File Save As dialog box is displayed if the document does not exist on disk or if the disk file has the Read-Only attribute set.

Automatic Save every: Set this option to save modified documents at regular time intervals.

Filter Binary Codes: By default, if you try to load a file that contains null characters (a character with a decimal value of 0), NoteTab will treat the first occurring null as the end of the file. This option lets you override this behavior by allowing NoteTab to substitute null characters with another character – defined in the "Substitute Binary Character" field. When set, you can load the full content of any type of file (including binaries). **Important:** saving files which have had characters substituted will not restore the control character! This option slows the loading of large files, so do not set it if you use only true text files.

Warn Partial Loading: Warns if a file has only been partially loaded due to the presence of null characters. This option has no effect if "Filter Binary Codes" is set.

Warn Binaries Substituted: Displays a warning message if binary characters have been substituted in the loaded document.

Backup Type: This setting is ignored if "Make Backups" is disabled. 1) ".bak extension" replaces the extension of the backup copy with ".bak" extension for backup files when checked. 2) "Tilde after dot" inserts a tilde "~" between the dot and the file's extension. 3) "Incremental" inserts an index number between the file name and its extension; the advantage of this method is that all backup copies are preserved (up to 999 copies), and the file type stays unchanged and therefore remains associated with its application.

For example, with a file called "NoName.txt", the following backup formats are produced depending on the chosen setting:

".bak extension" -> NoName.bak
"Tilde after dot" -> NoName.~txt
"Incremental" -> NoName.~001.txt, NoName.~002.txt, etc.

Undo After Save: Allows an undo operation after a modified document has been saved.

Match Directory: The working directory matches the directory of the active document when you use the main Open dialog box.

Use Default Directories: Set this option to always use fixed directories for opening and saving documents.

Open Directory: Default directory for opening files when the "Use Default Directories" option is set.

Save Directory: Default directory for saving files when the "Use Default Directories" option is set.

Options | Documents

The options you set on this tab define the default document properties that will apply to all newly opened documents and those that have the Default option checked. You can define word wrap, automatic indentation, tabulator width, font name and size, etc.

Dos ASCII: This setting is only available in the Document Properties dialog box.

Options | Internet

This tab is not displayed if the "Show HTML Features" has been unchecked!

Main Browser: When left blank, NoteTab can detect if the default browser (Internet Explorer, Netscape, or Opera) is already open and use that instance. You can specify another browser if you do not want to launch the default one.

Other Browser: Web browser path and file name used by "Other Browser" command.

HTML Header: HTML Header used by the "Document to HTML" command.

HTML Footer: HTML Footer used by the "Document to HTML" command.

HRef Tag: HRef tag format used when a file name is dragged from the Quick List to an HTML document.

Img Tag: Image tag format used when a GIF/JPG/BMP file name is dragged from the Quick List to an HTML document.

E-Mail Quote: Quote character(s) used by the "E-mail: Quote Text" command.

Highlight URLs: Highlight fully qualified URLs in text. Turning this option off will increase the speed of screen updating (only noticeable on slower systems). This setting is only available in NoteTab Pro.

Underline URLs: Underlines the highlighted links. This setting is only available in NoteTab Pro.

Options | HTML Files

This tab is not displayed if the "Show HTML Features" has been unchecked!

HTML File Extensions: This list is used by NoteTab Pro to identify which documents require HTML tag highlighting when this feature is enabled. If a document has an extension that matches one of the entries, then highlighting is performed. Dragging file names from the Quick List will also automatically insert the appropriate HREF or IMG tag. Use the Add button to insert new extensions and the Delete button to remove the selected item.

Server Name and Physical Path: These fields enable previewing of Web pages, such as ASP and Cold Fusion, through a local Web server (e.g. MS Personal Web Server). The relevant fields are "Server Name" (e.g. localhost) and "Physical Path" (e.g. c:\inetpub\wwwroot\html). When these fields are set, any file under the Physical Path will be displayed through the local Web server. Based on the example values, the following file name:

c:\inetpub\wwwroot\html\index.asp

is converted to:

<http://localhost/index.asp>

when sent for previewing to the browser.

Highlight HTML Tags: When enabled, any document with an HTML extension (listed in the HTML File Extensions list) will highlight HTML tags. Turning this option off will increase the speed of screen updating (only noticeable on slower systems). If the default extension (see File Filters tab) corresponds to an HTML extension, then new documents will also automatically highlight HTML tags. This setting is only available in NoteTab Pro.

Create XHTML Tags: When checked, produces tags based on the W3C recommendations for XHTML.

Create Uppercase Tags: When checked, produces tags with element and attribute names in uppercase. When unchecked, they are rendered in lowercase.

Create Lowercase URLs: Uncheck this option if you want NoteTab to preserve the character case of URLs inserted into HTML documents. When checked, URLs are converted to lowercase.

Options | File Filters

This tab provides a little editor that lets you control the file filters that appear in the File dialog boxes. The filters appear in the order in which they are listed in this dialog. You can edit, delete, and add new descriptions and wildcards. Use a semi-colon ";" to separate multiple wild cards (e.g. *.txt;*.otl;*.ini). You can reorder items by using the Move Up / Move Down buttons or by drag-and-dropping items in the list.

By default, the file filter list has an item for opening and saving DOS Ascii files. The program checks for the word ASCII (case does not matter) in the selected filter when you use the file dialog box. If it finds the word ASCII, it assumes that you want to open or save a file in the DOS Ascii format. So if you want to change the default file filters but also want to use ASCII documents, remember to put the word ASCII in the description part of at least one of the filter items!

The default extension field is used when creating a new document name. It is also used by the file dialog box, when you type a file name without extension.

Note that if you use a default extension with more than three characters, the Windows File Open/Save dialog box seems to ignore the extra characters. This behavior is controlled by Windows, not NoteTab. Enter the full extension in the File Open/Save dialog box to avoid this problem.

Options | Associations

This tab is not displayed if you have disabled the "Show Advanced Features" option, or if your registry does not allow changes to file associations!

This tab provides an editor that lets you define the types of files associated with NoteTab. You can edit, delete and add new extensions to the list. The order is of no importance.

Options | Colors

You can set the background color of documents and the color of text by clicking on the Background and Normal Text buttons respectively.

Notetab Pro: To change the color of selected text, use the Highlight Back and Highlight Text buttons. To change the color used to highlight HTML Tags and URLs, click on the HTML Tags and URL Text buttons respectively. The color of `<!-- -->` comment tags can be changed through the Comment button. Use the Highlight Tags button to change the color of HTML tags and URLs when they are selected.

To reset all colors back to the Windows standard colors, just click in the Use Default Colors check box.

Options | Toolbar

See the topic [Customizing the Toolbar](#) for an explanation of the options.

Options | Shortcut Menu

You can customize the content and order of commands in the main shortcut menu (the one that opens when you right click in the editor window). The choice of commands is the same as those available for the toolbar.

Just put a check mark in front of the commands you would like to show in the shortcut menu. You can drag-and-drop to reorder commands. Use the list's shortcut menu to sort items or group those you selected, or to check/uncheck all items.

Options | Tools

Find Word at Cursor: If the cursor is placed within a word, opening the Find or Replace dialog box will insert it in the Find field. If this option is not enabled, you have to select the word yourself to insert it in the dialog box.

HTML Convert Blanks: Converts leading blanks to non-breaking space tags (used by Document to HTML command).

Indent Size: Number of blank characters or tabs inserted by the Indent/Unindent function.

Case Sensitive Sorting: Defines whether the "Sort" command should be case sensitive or not.

Sort Removes Duplicates: Removes duplicate lines encountered by the sorting function.

Indent with Tabs: Set this option to use tab characters instead of blank spaces.

Date Filter: The date filter is used by the Insert Date command and the [Log feature](#). If you leave this field blank, the short date format defined in your Windows setup is used. The [Date and Time filters](#) topic gives you all the codes you can use to build a filter.

Custom Launch: Command string used by the "Launch Document" command (use ^** to indicate the position of fully qualified document name).

Paste-Board Divider: Define text or blank lines used to separate pasted Clipboard items when using the ["Paste Board" feature](#).

Paste Sound: Sound file to play when "Paste Board" receives text (uses PC speaker if left blank).

Options | Clipbook

Dock Left Side: Set this option to display the Clipbook on the left side of the editor window.

Dock Right Side: Set this option to display the Clipbook on the right side of the editor window.

Prompt for Header: Prompt for header name when a new item is added to the list. If this option is turned off, the Clip text or its first line is used to determine the header name.

Single Click to Paste: Setting this option will make the Clipbook insert items on a single mouse click instead of a double-click.

Indent on Paste: Set this option to automatically adjust the indent of pasted items with multiple lines.

Save Without Prompt: Automatically saves changes to Clipbook libraries without prompting.

List Color: Background color used for the Clipbook list box.

Header Color: Color used to indicate items in the Clipbook list that are executed instead of pasted in the document.

Icon Color: Color used to draw the Clipbook item icon.

Clipbook Font: Font used for the Clipbook list box.

Match Document Font: Sets Clipbook font to active document font.

Show Clip Icons: Set this option to display little icons next to wrappable or executable Clip items.

Show Headers: Set this option to display outline headers in the editor window (useful for editing the header).

Show Button Icons: Set this option to display the little icons in the buttons.

Stack Buttons: Set this option to stack the buttons if they don't all fit in the button bar.

Options | Tab Bar

Show File Extension: Show file extensions in the document tabs.

Show Tab Icons: Set this option to display the little icons in the tabs.

Stack Tabs: Shows document tabs stacked or unstacked.

Show Name as Tooltip: Set this option to display document names at popup tooltips when cursor passes over the tabs.

Play Sound: Plays the chosen WAV file when you click on a document tab button. No sound is produced if the tab is already active.

Active: Defines color of tab when it is selected.

Inactive: Defines color of tab when it is not selected.

Modified: Defines color of tab matching a modified document (only used when icons are not displayed).

Paste Pad: Defines color used to identify the Paste Board tab.

Font: Defines the font used for text in the tab bar.

Options | Outline

Font: Font used for the Outline list box.

Color: Background color used for the Outline list box.

Show Heading in Editor: Set this option to display outline headings in the editor window (useful for editing the heading).

Underline Headings: Underlines headings when exporting or printing outline documents.

Highlight Links: Highlight hyperlinks in text. Turning this option off will increase the speed of screen updating (only noticeable on slower systems). This setting is only available in NoteTab Pro.

Show Table-of-Contents: Creates a table-of-contents in the HTML file produced by the "Document to HTML" command.

Number Headings: Numbers headings in order of appearance. This setting is used by the "Document to HTML" command.

TOC Title Format: Defines the format and text used to create the table-of-contents. This setting is used by the "Document to HTML" command.

Title-Section Format: Defines the format used to create the title section of the HTML document created by the "Document to HTML" command. The code "**^1**" is used to mark the position of the Title element and "**^2**" marks the position of the table-of-contents element.

Topic Format: Defines the format used to convert Outline headings to HTML. The code "**^1**" is used to mark the position of the Heading element and "**^2**" marks the position of the Content block.

Options | Quick List

Font: Font used for the Quick List window.

Color: Background color used for the Quick List window.

Type List: Defines the content displayed by the Quick List window.

Show Full Names: Set this option if you want to display the full file name in the window.

Options | Advanced

Support Files Location: Only use this field if you have placed the NoteTab program file in a different directory from its support files (Help, dictionaries, settings, etc.).

Store Settings in Registry: Set this option if you want to store the NoteTab settings in the Windows registry instead of an INI file. Note that using the registry instead of an INI file slows down the loading and closing of NoteTab. Use the INI file for best performance.

Clear Properties: Clear all stored document properties. Next time you open documents, the default settings defined in the Options dialog box are used. This command has no effect on the documents that are currently open.

Store Properties: Defines the maximum number of document properties that should be remembered between sessions.

Text-Size Limit: You can configure the maximum document size the editor open. The value is in bytes. The value cannot exceed 2'147'483'632 (2 GB). This option will not change the buffer size of documents that are currently open.

Undo-Size Limit: You can configure the size of the undo buffer through this field. The value is in bytes and cannot exceed 65'532 (64 KB). By setting a smaller value you save system resources. This is only useful if you intend to open a very large number of documents at once. This option will not change the buffer size of documents that are currently open. This field is only available in [NoteTab Pro](#).

Paragraph Limit: You can configure the maximum paragraph length allowed in documents. When word wrap is turned off, a paragraph represents a single line. The length is measured in characters and cannot exceed 32'767 (32 KB). Choosing a lower value will save system resources. This option will not change the paragraph size limit of documents that are currently open. This field is only available in [NoteTab Pro](#).

Key Mapping: You can choose between default Windows or WordStar key mapping. Note that conflicting NoteTab shortcuts are not available when you choose WordStar key mapping. This field is only available in [NoteTab Pro](#).

Highlight Clip Code: When set, instruction, label, and comment lines in Clipbook libraries are highlighted. This option is only available in [NoteTab Pro](#).

Quick List

Paste-Board Feature

Auto-replace Mode

Automatic Scrolling

Using and Creating Clipbars

Editor Clipboard

Quick List

You can open the Quick List by pressing F6. When you use this feature for the first time, it will display the list of documents currently open in NoteTab. You can quickly move to another document by selecting it from that list. This is particularly useful when you have many documents open.

The Quick List window can be used to display four types of content:

1. Opened documents (Documents)
2. Favorites list (Favorites)
3. Files in a directory (Disk Files)
4. Clip Assistant

The simplest way to change the type of list is by invoking the shortcut menu in the Quick List window. Choosing Disk Files will display all the files matching the wildcard criteria in the current directory. You can change directory or file criteria by editing the content of the combo box and then pressing the ENTER key. Double-clicking in the combo box will open a dialog box that makes it easy to change directories. To open one of the files listed, just double-click on its name.

The Favorites list contains the same items as the Favorites menu. When you use NoteTab for the first time, the Favorites list will be empty. Once you have added documents to the list, just double-click on a document name to open it.

When the Favorites or Disk Files are displayed, you can drag-and-drop the names into your document. If the document has an HTML extension (as listed in the Options | HTML Files dialog box), dragged file names are converted to HTML links: image files (GIF, PNG, JPG, JPEG, and BMP) will be inserted as `img` tags and other files as `href` tags. The `img` tags are complete with the size attributes of the image. You can change the format of the default tag by editing the appropriate fields in the Options | Internet dialog box. By default, URLs are automatically converted to lowercase. However, you can preserve the original case of a file name by holding down the `Ctrl` key while inserting it in HTML documents. Alternatively, you can change the setting of the "Create Lowercase URLs" in the Options | HTML Files dialog box.

The Clip Assistant contains all the commands and functions available in the Clip syntax. You select and complete commands and function through the Clip Wizards (which are themselves created through Clip syntax).

The Quick List's shortcut menu lets you close, delete, or shift the focus to the selected document, or close the window.

Useful Mouse and Keyboard Shortcuts

- *Delete* erases the selected document when the list shows the opened documents.
- *Enter* or *Double-click* opens and shifts the focus to the selected document.
- *Ctrl+G* shifts the focus from the Quick List to the editor.

Paste-Board Feature

Any document in the editor (as long as it is not Read-Only) can be configured to capture text that is cut or copied to the Clipboard from any Windows application. First select the document you want to use as a "paste board", then activate the "Use as Paste Board" option in the Document menu. As long as this option is active, the document will capture and append all text sent to the Clipboard. The computer will beep or play a sound each time the paste-board document captures some text.

The paste-board feature, for example, makes it easy to create a list of links or store important information as you surf the Web.

Note that only one document at a time can act as a paste board. When you activate this option on another document, the previous paste-board document stops capturing Clipboard text.

Auto-replace Mode

When Auto-replace mode is enabled (either by opening an "auto-replace" library, or through the "Tools | Auto-replace Mode" menu command), NoteTab will check for a matching item in the active library whenever you press any of the following characters (activation keys):

TAB SPACE RETURN !" , . : ; = > ?] }

If a matching Clip is found, the typed item in the document is automatically replaced according to the Clip instructions.

Note that if the Clip contains the cursor-positioning code **^&**, the activating key is automatically discarded. You can also use the activating key in Clip code through the new predefined variable **^%AR_KEY%**.

Usually, when using libraries for auto-replace mode, it is not necessary to see the list of Clips. You can hide the Clipbook window by using the "Tools | Clipbook Properties | Hide Clipbook" menu command. This way, the Clipbook tool is still available but you have more screen space for your document.

For a demonstration of this auto-replace feature, try the included libraries AutoCorrect and HTML-ar. See topics below for more information.

AutoCorrect library

Auto-replace mode is automatically enabled when this library is opened in the Clipbook window (this window may be hidden). The library contains over 900 entries corresponding to the most common typos and spelling mistakes in the English language. To open this library, click on the button labeled **AutoCorrect** in the button bar at the bottom of the NoteTab window.

This library provides a feature similar to AutoCorrect available in WinWord and other word processors. As you edit your document, NoteTab checks the word preceding the cursor whenever you press an activation key. If this word matches a Clip name in the list (case insensitive), it is automatically replaced according to the Clip instructions. For example, if you type **acn** in your document followed by the spacebar key, NoteTab will automatically replace the typo with the word **can**.

You can add your own entries to the library by choosing the menu command Tools | Clipbook | Add New Clip (or the keyboard shortcut Ctrl+F2). You will then be prompted to enter the Clip name – this should correspond to the typo or misspelling. Next the new Clip is added to the list and the Clip editor opened. Just type the correct spelling or text you want to be used by the auto-replace feature, save the library (Ctrl+S) and close the Clip editor (Ctrl+F4).

You can also easily remove an auto-replace item by selecting it in the Clipbook window and pressing the Delete key.

HTML-ar library

Auto-replace mode is automatically enabled when this library is opened in the Clipbook window. The library contains all the common HTML tags. To open this library, click on the button labeled **HTML-ar** in the button bar at the bottom of the NoteTab window.

You should not really use these Clips the traditional way (i.e. by dragging or clicking on the Clip item). As long as Auto-replace Mode is enabled, you just type normally in your document. When NoteTab, for example, detects that you have typed **<font** followed by one of the activation keys, it automatically

activates the matching Clip item. Type, for example, **** and you will automatically get the closing tag after the cursor: **|**.

Automatic Scrolling

Automatic scrolling allows hands-free reading of any document. You can set the speed in characters per second and the program will scroll through the text one line at a time until it reaches the end. The Auto Scroll command is available in the Tools menu.

The Auto Scroll "command center" dialog has five buttons and an input field that lets you set the scroll speed in characters per second. This dialog is modal, so you cannot return to the editor unless you close it.

For left to right, the buttons are...

- Play: Starts scrolling at the speed defined by the Speed field.
- Quit: Stops scrolling and closes the dialog box.
- Pause: Stops scrolling.
- Rewind: Same as Page Up (pressing this button pauses scrolling).
- Forward: Same as Page Down (pressing this button pauses scrolling).

You can also use the keyboard to control scrolling...

- Spacebar: Toggles between Play and Pause mode.
- Page Up: Same as Rewind button.
- Page Down: Same as Forward button.
- Esc: Stops scrolling and closes the dialog box.

If you edit the scroll speed while the document is scrolling, the pause button will be activated. Press the Play button to continue scrolling at the new speed.

Using and Creating Clipbars

The Clipbar is located below the standard Toolbar when in view and was added to NoteTab Pro and Std in version 4.8 (this feature is not available in NoteTab Light). It is a powerful tool allowing you to add your own clips and icons. The number of Clipbars you can have is only limited to your hard disk space. See our Web site for links to icons and also to download icons created by other NoteTab users. NoteTab currently provides four Clipbars: HTML-1, HTML-2, TopStyle, and Utilities. You can open these by choosing the Clipbar command under the View menu.

Tip: if you use the HTML-1 and HTML-2 Clipbars, the "Bold" and the "Italic" buttons can be configured to either insert the typical and <I> tags respectively, or the logical-style tags and . To toggle the style used, just hold down the Ctrl key when you click on either button. The last-used style is remembered even when you restart NoteTab.

The Clipbar acts a little differently from the clips in a Clipbook Library. Instead of NoteTab reading the clip in the actual opened Library, it reads it from the disk file. For that reason, if you edit a clip, you must save it before it will work on the Clipbar. The Clipbar's information (Library, clip, and button) is stored in a .ctb file. There will be one of those files made for every Clipbar made. They are found in the Libraries folder.

There are features on the Clipbar's content menu. To see them, right click over any part of the Clipbar. If an item is dulled, it means there is nothing for NoteTab to do with that particular feature. Notice the difference if you right click over a button and over a blank space out to the right. Also note the difference if you right click over a separator and a button in the first section of the menu. You can view the Tooltip by holding your mouse over the icons and also on the Status bar at the bottom of your screen.

The icons (also known as images, buttons, or glyphs) on the Clipbar must be either a .bmp or .ico format. The .bmp format is preferred because NoteTab has to change the format internally (not the file) in order to show it on the Clipbar. However, most users should not notice any difference in performance if .ico files are used.

Images are made of many very tiny squares called pixels. The transparency is read from the bottom left pixel of the image. That means whatever color that pixel is, the same color in the image will not be seen on the Clipbar. You may have noticed some images you see on computers are square like a photograph and have a background. That is because there is no transparency set which may be due to the file type. Think of transparent as invisible. The size of the icon should be no larger than 18 by 18 pixels for best results. NoteTab will shrink bigger icons to those dimensions; therefore, buttons that are not that size may get distorted. NoteTab's icons are available for you to use on the Clipbar. Other different icons must be kept in NoteTab's Library folder. The Clipbar uses the Toolbar's Options to determine whether the buttons are stacked or scroll out to the right when the bar becomes full.

How to Add a Clip to the Clipbar

1. Choose the Library that contains the clip you want; make it active by pressing on the button at the bottom of your screen on the Libraries Bar.
2. Make sure you do not have "Single Click to Paste" checked in Options under the Clipbook tab. ([View | Options | Clipbook](#)).
3. Find the clip you want to add and left click once on it holding your mouse button down. Drag and drop it to the Clipbar by moving your mouse up to the Clipbar and dropping (let go of the button) the clip on it.

How to Change an Icon on the Clipbar

1. Right click over the icon you want to change and choose "Button Properties..."
2. The clip's Button Properties dialog will come up. You may choose one of NoteTab's buttons on the first tab (Toolbar icons) or if you installed your own in the Libraries folder they will be on the second tab (Custom icons).
3. Click on the button you want to use and then click OK.
4. Right click over the Clipbar and choose "Save Clipbar."

Note: If you change the name of the clip in Button Properties you must also change it in the Library as well. It is not done automatically for you. The same goes for the Library name.

Tips

You may consider having separate Libraries for each Clipbar you make so it will be easier to keep track of them all.

Some users might find it easier to edit the .ctb file for the Clipbar by right clicking over the Clipbar and choosing "Edit Clipbar". The format is:

"Library Name", "Clip Name", "Icon Name (or NoteTab icon number)"

[Clipbook Libraries](#)
[In-Context Clipbook](#)
[Clipbook Wizard](#)
[Clip Basics](#)
[Clip Programming](#)

Clipbook Libraries

You can easily edit or create your own Clipbook libraries directly in the editor. Choose Open Library from the Tools | Clipbook menu to edit an existing library, or New Library if you want to create a new one. The best way to learn how to create your own libraries is to look at those packaged with the NoteTab program.

NoteTab Libraries

You will find a brief description of all the libraries distributed with NoteTab below. Just click on the desired library button in the bar at the bottom of the NoteTab window to open it in the Clipbook tool. You may add your own Clips to these libraries, delete items, and modify them as much as you wish. You can also use Explorer to delete libraries you may not find useful. Their file extension is .clb and they are normally stored in the Libraries subdirectory.

1. AutoCorrect

This is an "auto-replace" library. When it is opened in the Clipbook window (this window may be hidden), Auto-replace mode is automatically enabled. It provides a feature similar to AutoCorrect available in WinWord and other word processors. As you edit your document, NoteTab checks the word preceding the cursor whenever you press an activation key:

`TAB SPACE RETURN ! " , . : ; = > ?] }`

If this word matches a Clip name in the list (case insensitive), it is automatically replaced according to the Clip instructions.

2. CaptureLinks

This library demonstrates how text can be captured from the Windows Clipboard and used in a document. It optionally uses spoken messages to indicate what it is doing in the background. NoteTab will build a Web page for you in the background as you copy link information to the Clipboard.

3. Euro

Convert between any currency using the Euro as reference. All currencies that are part of the 1999 monetary union are listed, as well as a few other major currencies.

4. FormatEmail

This library will format an e-mail message by quoting the text for reply. When activated, it will capture all text sent to the Windows Clipboard. The way the Clip has been designed, you should first copy the name of the person who sent you the e-mail, then copy the part of the message you want to reply to. The formatted reply text is then stored in the Windows Clipboard ready to be pasted into a new e-mail message.

5. FTP

Use this library to download and edit files from remote servers. The commands use the Windows FTP.exe program that is distributed with Windows 95, 98, 2000, and NT4.

6. HTML

This is the perfect tool to help you create Web pages. All the important tags are readily available from the Clipbook list. Tags are wrapped around selected text. Clip wizards help you complete some of the more complex tags and HTML structural elements.

Tip: The "Bold" and the "Italic" Clips can be configured to either insert the typical and <I> tags respectively, or the logical-style tags and . To toggle the style used, just hold down the Ctrl key when you click on either Clip. The last-used style is remembered even when you restart NoteTab.

7. HTML-ar

If you do not like using the mouse, this "auto-replace" library will help you save a lot of time by completing tags for you as you type. For example, if you type the HTML tag "" in your document, NoteTab will automatically complete it with the closing tag placed after the cursor: "|".

8. PasteClips

To see how this library works, open it in the Clipbook and double-click on the "Start" Clip to activate the capture mode. Then, bring another application forward (e-mail, IRC, etc.). Next, type one of the Clip names listed in this library (for example "ys") in your application, select it and copy it to the Clipboard. If all works correctly, the selected text should be replaced by the content of the Clip you chose ("Yours sincerely" if you used the example). Of course, you can add your own shortcut text to the library (such as passwords, boilerplate text, etc.) and use them in other applications while NoteTab is working in the background.

9. RemindMe

This library shows how you can use the timer Clip event to create a scheduler. The timer event executes the "Remind Me" Clip every 30 seconds. The script in that Clip then checks the system clock, and if one of the tested time ranges is matched, it jumps to the specified label and displays an appropriate message.

10. SampleCode

This library contains various items that illustrate how certain Clip commands and functions can be used. Most of these Clips do not really provide useful functions, but they can help you understand how to create your own Clips. The best way to learn is to take a look at the Clip code and compare it with what happens when you activate it.

11. Smilies

A long list of different types of smilies you can use in your text to express your mood.

12. TopStyle

This library requires Bradsoft's [TopStyle CSS editor](#), version 1.51 and above -- a program for editing Cascading Style Sheets. You can learn more about this useful program and download the latest free version by visiting the following Web page:

<http://www.bradsoft.com/redir.asp?id=130>

The Clips in this library make it easy to edit style sheets.

13. Utilities

As the name suggests, this library contains all kinds of useful utilities. You'll find Clips to manage your Clipbook libraries (install or remove them), insert popular Internet acronyms in your text, underline titles, compare two files, create Web pages based on images in a directory, strip tags from all open HTML files, convert all open text files to Web pages, convert between units, calculate mortgage rates, etc.

Library File Format

The library file format is very simple; take a look at some of the samples provided with NoteTab to see how they are structured. Basically, there are two possible formats:

The simplest format has one text item per line and the header just duplicates the content. Empty lines are ignored.

The second format can have multi-line text items and each item can be identified by a descriptive header. This format must start with an equal sign (=) on the first line, followed by the text "MultiLine" (without quotes, character case does not matter). Each pasteable text item must be separated by an empty line. If you want to specify a header, add "H=" (without the quotes) followed by your header. The header must be directly on the line above the item it describes. Although it is not required in the current version, it is recommended that you place double quotes ("") around each header.

By default, headers are sorted in the Clipbook window. If you don't want them to be sorted, place the text "NoSorting" (without quotes, character case does not matter) in the descriptive header (this command applies to both library formats).

When you save a Clipbook library, the Clipbook window is immediately updated. You can create as many libraries as you like; they will all appear in the Clipbook combo box.

Clip headers can be "disabled" by placing an underscore "_" in front of the name. The color of the name changes to gray and the user cannot execute the Clip with the mouse or the keyboard. Disabled Clips can, however, be executed from other Clips. Just add another underscore in front if you want such a character displayed in the name but don't want the Clip to be disabled.

If you have created Clipbook libraries that may benefit other users, why not send us a copy! We will be glad to add them to the Clipbook repository at the NoteTab Web site.

Each library can have up to about 5440 Clipbook items, and you can have up to 5440 libraries.

For details on more advanced features, see the topic on Clip Syntax.

In-Context Clipbook

You can invoke the "In-Context Clipbook" by hitting the ESC key while you work in a document. Doing so will display a combo box (filled with the content of the active Clipbook library) at the cursor position. Selecting an item from the combo box will then paste the Clip text in your document. Typing in the combo box will automatically scroll to the first item matching the characters entered. Press ENTER or double-click to confirm your choice. You can choose a different Clipbook library by invoking the combo box shortcut menu. To cancel the "In-Context Clipbook", just hit the ESC key or click with the mouse in another part of the editor.

Clipbook Wizard

Clipbook Wizards are fairly easy to create using the special Clip syntax. Each field can either represent a plain input box or a combo box with a list of predefined values. These can have a descriptive label and contain predefined values.

Many examples are provided in the sample Clipbook libraries included with NoteTab. Open the HTML library and double click in the list on the item "HTML begin" to see an example of a relatively complex Wizard. You will find another type of example by double-clicking on the "Multiple fields" in the SampleCode library.

Clip Basics

You will find information about the following topics:

- *Wrapping a Clip around selected text*
- *Adding prompts in a Clip*
- *Using the Clipbook to launch other programs*
- *Program commands*
- *Clipbook Programming*

For details on using the Clip editor, see the topic [Create Clipbook Libraries](#). To help you with the Clipbook syntax, a "Clip Assistant" is available under the Tools | Quick List Properties menu. When you activate this command, a list of ready-to-use syntax elements is displayed in place of the [Quick List](#). You can paste the items into your document by double-clicking on them or by dragging them. You can edit the ClipHelp.CLH file if you want to customize the content of the Clip Assistant list.

For those of you who plan to develop many libraries, make sure you take a look at [ClipWriter](#). This program, written by an expert Clip programmer, is a powerful alternative to the NoteTab Clipbook editor. A free trial version is available for evaluation. By visiting the NoteTab Web site, you will also find other useful resources that can help you get to grips with developing Clipbook libraries.

Wrapping a Clip around selected text

Normally, if you select text in your document and then paste an item from the Clipbook, the document selection is replaced by the Clip text. However, if the Clip text item includes the two-letter code `^&`, the highlighted text in the document is inserted at the code's position and the resulting text is pasted in the document. This feature is particularly useful when producing HTML documents. For example, if you have defined a Clipbook item under the title *Italic*:

```
<I>^&</I>
```

Selecting text in your document (for example *NoteTab*) and then double-clicking on the *Italic* item in the Clipbook will surround the text with the italic tags:

```
<I>fabulous</I>
```

Adding prompts in a Clip

Sometimes you may want to complete a Clip just before it is pasted in your document. HTML tags are a good example where this could come in handy. If you use the sample HTML library that comes with this program, you will notice that when you double-click on the *Font* item, the program will display a dialog box prompting you to enter a value for the font size. You have a choice to type a value or pick it from a combo box.

Adding this type of behavior to a Clip is very easy. Simply place the two-letter code and brackets `^?[]` in your Clip at the position where user input should be inserted. For example, when you double-click on a Clipbook item with the following text:

```
<FONT SIZE=^?[]>
```

a dialog box with a plain input field will prompt you to complete the Clip. The user input replaces the two-

letter code, and then the completed tag is copied to the document. If you would like the prompt to be a bit more descriptive, you can specify the text that will be displayed in the dialog box by placing it between square brackets:

```
<FONT SIZE=^?[Font size]>
```

You can define a default value that will be displayed to the user. Just add an equal sign after the prompt followed by the default text. The following example will show the value 12 in the input field of the dialog box:

```
<FONT SIZE=^?[Font size=12]>
```

You can produce more sophisticated prompts by offering the user a list of choices. When the program detects this type of prompt, it displays a dialog box with a combo box. To specify a list of choices, use an equal sign "=" after the prompt text followed by the values. Use a "|" (Alt+0124) as a separator between the values. For example, the following item will show a combo box labeled "Font size" with the values 8, 10, 12, 14, 18, 24, and 36 in the list:

```
<FONT SIZE=^?[Font size=8|10|12|14|18|24|36]>
```

The user can also enter values that are not listed. If, however, the input should be limited to the values available in the list, just double the equal sign as in the following example:

```
<FONT SIZE=^?[Font size==10|14|18]>
```

If you would like to select a default value when the dialog box is displayed, just place an underscore character "_" in front of the desired value. In the following example, the value 10 will be used as the default value:

```
<FONT SIZE=^?[Font size=8_10|14|18]>
```

You can display in the combo box list different text from what you actually want to use. Use the ^= code to separate the text to display (on the left side) from the text to use (on the right side). The example below will show a combo box with three items in the list: Small, Medium, Large. The medium item will be selected as the default value since it starts with an underscore character.

```
<FONT SIZE=^?[Font size=Small^=6_ Medium^=10|Large^=14]>
```

A Clip can have as many prompts in it as you like. If you repeat a prompt within the same Clip, the initial answer entered by the user is inserted in its place (the dialog box is not displayed again). In the following example, the user will only be prompted once:

```
Dear ^?[First Name=Lisa|Bill|Jane|Ted],  
How are you ^?[First Name]?
```

Notice that it is not necessary to duplicate the whole prompt item, but only the prompt header. Suppose the user chose Jane, this is what will be pasted in the document:

```
Dear Jane,  
How are you Jane?
```

Using the Clipboard to launch other programs

Any Clipboard item that starts with the two-letter code ^!, will be launched when invoked. For example, double-clicking on a Clip with the following text:

^!calc.exe

will execute the calculator provided with Windows. Note that if the application is not in the search path, you will have to add the fully qualified path to the application you want to use.

Use the DOS command to launch some DOS applications if the method described above does not work:

^!DOS edit.exe

A Clip can behave a bit like a batch file in that it supports multiple commands – each on a separate line. For example, if you have a Clip with the following instructions:

```
^!calc.exe  
^!write.exe "c:\My Documents\MyFile.Doc"
```

Executing it will open both the calculator and the word processor. You can control the timing for opening consecutive applications. The instruction **^!WAIT** will wait until the previously launched application has closed before executing the next command. You cannot use the editor when it is in Wait mode (the editor's window caption indicates it is waiting for the application to close). The instruction **^!PROMPT** will display a dialog box prompting the user to click on the OK button before passing on to the next instruction(s). In the following example, the Write program will only be launched once the Calculator program has been closed:

```
^!calc.exe  
^!WAIT  
^!write.exe "c:\My Documents\MyFile.Doc"
```

Note that the **^!WAIT** command may not work with all applications (during testing, it did not work with the MS-Write program).

You can specify what message should be displayed in the prompt window. The following example will display a message box with the text "Click on OK when you are ready!":

```
^!PROMPT Click on OK when you are ready!
```

Use the **CONTINUE** command to give the user the possibility to cancel the execution of all Clip commands following it. The syntax is similar to **PROMPT** but the message box shows an OK and a Cancel button. If the user presses cancel, the Clip execution is interrupted. The following example will display a message box with the text "Click on OK to continue or Cancel to stop!":

```
^!CONTINUE Click on OK to continue or Cancel to stop!
```

If you want to open the current document in another application, use the three-letter code **^**** to indicate where the document name should be inserted in the command line. For example, if you want to open the current document in MS-Write, you would use the following command:

```
^!write.exe ^**
```

To specify a document name without its extension, use the following two-letter code instead: **^***

Documents that have not been saved are stored in a temporary file when this type of command is used. The temporary file is then substituted for the document name. If you do not want NoteTab to create a temporary file in this case, then substitute the asterisk character **^**** with a **^#**. For example:

```
^!write.exe ^##
```

You can also specify additional parameters that your application recognizes. Here is an example that will do a syntax check on a Perl script (assuming it is the current document in the editor):

```
^!perl -c ^**
```

If you are editing a document that is associated with another application, you do not need to specify the application name. For example, if you are using an HTML document (with the appropriate extension!), the following command will open the current document in the browser that has been associated with HTML extensions:

```
^!^**
```

Note that the **^!WAIT** instruction only works with launch commands that include the program name. It will not work with the example above!

You can also open URLs in the associated browser. To do so, the two-letter code **^!** must be followed by the following text **"URL"** (without the quotes). For example, the following command will open the link to the NoteTab Homepage:

```
^!URL http://www.notetab.com/
```

You will find more examples in the Clipbook library "Utilities.clb" which is installed with NoteTab.

Program commands

You can execute any NoteTab command that is available in the toolbar (it does not matter if the button is displayed in the toolbar or not). You specify the command you want to execute by using the keyword **"TOOLBAR"** followed by a space and its tooltip text. The button tooltip text can be seen by placing the mouse cursor over the button or by looking at the content of the list box in the Toolbar properties dialog box.

The following example will activate the "Open Document" command:

```
^!TOOLBAR Open Document
```

You can of course create a sequence of instructions. The next example will create a new document, paste the content of the Clipboard into it, then activate the Save As dialog box:

```
^!TOOLBAR New Document  
^!TOOLBAR Paste  
^!TOOLBAR Save As
```

Clipbook Programming

You can significantly extend the functionality of NoteTab by using more sophisticated features of the Clip language. Starting with NoteTab v4.5, you can create quite sophisticated scripts to handle anything from underlining a title to creating complex Web pages. See the topic [Clip Programming](#) for more information on the subject.

Clip Programming

You can significantly extend the functionality of NoteTab by using more sophisticated features of the Clip language. Starting with NoteTab v4.5, you can create quite sophisticated scripts to handle anything from underlining a title to creating complex Web pages. See the following topics to learn about the Clip programming language:

[Clip Language - Introduction](#)

[Clip Language - Labels](#)

[Clip Language - Variables](#)

[Clip Language - Events](#)

[Clip Language - Custom Functions](#)

[Clip Language - Testing Conditions](#)

[Clip Language - Message Boxes / Prompts](#)

[Clip Language - Text Processing](#)

[Clip Language - HTML Processing](#)

[Clip Language - Select, Copy, and Insert Text](#)

[Clip Language - Clipboard Copy / Paste](#)

[Clip Language - Find / Replace Text](#)

[Clip Language - Document Processing](#)

[Clip Language - Document Management](#)

[Clip Language - Disk Files](#)

[Clip Language - Disk Directories](#)

[Clip Language - Executing Programs](#)

[Clip Language - Using Perl, Gawk, and Other Scripts](#)

[Clip Language - Using Other Clips](#)

[Clip Language - Clipbars](#)

[Clip Language - Toolbar Commands](#)

[Clip Language - Keyboard Macros](#)

[Clip Language - Windows Registry and Shortcuts](#)

[Clip Language - Miscellaneous Commands / Functions](#)

[Clip Language - Mathematics](#)

[Clip Language - Dates](#)

[Clip Language - Clip Wizard and Fields](#)

[Clip Language - Masked Input Fields](#)

Clipbook Programming

[Clip Language - Introduction](#)

[Clip Language - Labels](#)

[Clip Language - Variables](#)

[Clip Language - Events](#)

[Clip Language - Custom Functions](#)

[Clip Language - Testing Conditions](#)

[Clip Language - Message Boxes / Prompts](#)

[Clip Language - Text Processing](#)

[Clip Language - HTML Processing](#)

[Clip Language - Select, Copy, and Insert Text](#)

[Clip Language - Clipboard Copy / Paste](#)

[Clip Language - Find / Replace Text](#)

[Clip Language - Document Processing](#)

[Clip Language - Document Management](#)

[Clip Language - Disk Files](#)

[Clip Language - Disk Directories](#)

[Clip Language - Executing Programs](#)

[Clip Language - Using Perl, Gawk, and Other Scripts](#)

[Clip Language - Using Other Clips](#)

[Clip Language - Clipbars](#)

[Clip Language - Toolbar Commands](#)

[Clip Language - Keyboard Macros](#)

[Clip Language - Windows Registry and Shortcuts](#)

[Clip Language - Miscellaneous Commands / Functions](#)

[Clip Language - Mathematics](#)

[Clip Language - Dates](#)

[Clip Language - Clip Wizard and Fields](#)

[Clip Language - Masked Input Fields](#)

Clip Language - Introduction

The Clipbook tool is programmable and supports features such as variables, functions, commands, conditional statements, etc. It also integrates support for Perl and Gawk interpreters; scripts can be stored in Clips and applied directly to highlighted text or a whole document. A good way to learn how the Clip language is used is to look at some of the sample Clipbook libraries provided with NoteTab.

The Clip language has been designed so that it can be mixed with plain text and even other script languages. Because of this, it looks and behaves differently from other programming languages. Clip instructions use tags to clearly delimit them from other text. The language is very easy to learn once you become familiarized with the various syntax elements and their format. Names of the various syntax elements are not case sensitive.

The Clip language understands and uses string (a sequence of characters), integer (negative and positive whole numbers), and Boolean (True/False) data types. However, values are always stored in text format no matter what data type they represent. Conversion between the data types is handled automatically internally. This approach makes the language particularly easy to use even for those with no prior experience in programming.

The "SampleCode.clb" library that is included with NoteTab provides examples for many of the commands and functions explained here. The best way to learn how to use the new Clip language is to look at some of the sample Clips in detail. You can also learn about Clip programming by joining one of the NoteTab mailing lists or by subscribing to the "NoteTab Tips and Tricks" weekly newsletter (information about all this is available at the NoteTab Web site).

Useful tip when testing Clip code:

You can use the "Ctrl+Alt" key combination to stop the execution of a Clip (especially useful when stuck in a loop).

See below for a brief explanation of the various syntax elements.

Clip Language Elements:

Commands:

A command is an instruction that performs a specific task. All commands must start at the beginning of a line with the two-letter code "^!" followed either by the name of the Clip command or a program command line. Note that since version 4.8, you can now also have commands after conditional statements. Each command must occupy a single line in the Clip. Many commands accept or require one or more parameters – data that you pass to the instruction that can determine how the command operates. Parameters must be separated with a blank character; if a parameter contains a blank character it must be placed between double-quotes. Example of a command:

`^!Close "File Name" SAVE`

Functions:

Unlike commands, functions return a result. A function can be placed anywhere in a Clip text but it must not be split by a line break. Functions can also be embedded within each other. Each function is bound by a dollar character "\$" and preceded with the circumflex accent "^". Many functions require one or more parameters which must be placed between parenthesis immediately after the function name. If there are several parameters, they must be separated from each other with a semicolon ";". Note that if a parameter contains a semicolon character it must be placed between double-quotes. Example of a function: `^$StrIndex(String;Index)$`

Variables:

A variable is a named memory element used for storing values of any data type. It can contain anything from nothing to multiple megabytes of text. A variable can be placed anywhere in a Clip text but it must not be split by a line break. Each variable is bound by a percentage character "%" and preceded with the circumflex accent "^" when referring to the value it contains. A variable is defined as soon as it is used and it is always empty at first – unless you immediately assign a value to it. A variable name can have any character except those that delimit words (space, tab, punctuation, etc.). Example of a variable:

`^%MyVariable%`

Labels:

Labels are used to mark a place within a Clip to which an instruction can jump. They are useful for controlling the sequence of execution and for creating repeating instructions (loops). A label is defined by starting a line with a colon ":" followed by a name. The name can have any character except those that delimit words (space, tab, punctuation, etc.). You can jump to a label with the `^!Goto` command or after a conditional operation. Labels are skipped over when not used within a set of instructions. The scope of a label is limited to the Clip in which it is defined. A Clip should not contain duplicate label definitions (only the first occurrence would be used). Example of a label:

`:MyLabel`

Events:

Events are triggered when something particular happens in the program or Windows. The Clip language recognizes four distinct events: the opening and closing of a Clipbook library, sending text to the Clipboard, and timer intervals. You can execute Clip code when these events are triggered. Timer events are controlled with the `^!TimerPlay` command; the three other events are defined by creating Clips with the corresponding event name (OnOpen, OnClose, OnClipboardChange).

Comments:

Inserting comments in a Clip is very easy: just place a semicolon (;) at the start of each line containing a comment. If you come across a situation when a line in a Clip should start with a semicolon and it is not a comment, just add an extra semicolon in front of it. Note that comments are only valid *within* a Clip!

Errors:

When the Clip interpreter encounters a syntax error, it displays an error message and the line in which the fault was encountered. The line number indication, however, does not take into account eventual comments that may have been inserted between the instructions. The reason behind this is that comments are stripped from the code before it is executed.

Parsing and evaluation sequence

A. Whole Clip

1. Comments removed
2. Date fields evaluated `^[...^]` (old format)
3. Text selection inserted if `^&` found
4. `^!CONTINUE` command executed if on first line
5. Clip wizard based on `^?[...]` fields displayed
6. Mathematical fields evaluated `^$[...]` (old format)

B. Line by line

1. Variables resolved (from right to left)
2. Document name using `^*`, `^**`, `^#`, `^##` codes resolved
3. Functions resolved (from right to left)
4. Clip wizard based on `^?{...}` fields in line displayed
5. Clip Command executed

Clip Language Topics

[Clip Language - Labels](#)

[Clip Language - Variables](#)

[Clip Language - Events](#)

[Clip Language - Custom Functions](#)

[Clip Language - Testing Conditions](#)

[Clip Language - Message Boxes / Prompts](#)

[Clip Language - Text Processing](#)

[Clip Language - HTML Processing](#)

[Clip Language - Select, Copy, and Insert Text](#)

[Clip Language - Clipboard Copy / Paste](#)

[Clip Language - Find / Replace Text](#)

[Clip Language - Document Processing](#)

[Clip Language - Document Management](#)

[Clip Language - Disk Files](#)

[Clip Language - Disk Directories](#)

[Clip Language - Executing Programs](#)

[Clip Language - Using Perl, Gawk, and Other Scripts](#)

[Clip Language - Using Other Clips](#)

[Clip Language - Clipbars](#)

[Clip Language - Toolbar Commands](#)

[Clip Language - Keyboard Macros](#)

[Clip Language - Windows Registry and Shortcuts](#)

[Clip Language - Miscellaneous Commands / Functions](#)

[Clip Language - Mathematics](#)

[Clip Language - Dates](#)

[Clip Language - Clip Wizard and Fields](#)

[Clip Language - Masked Input Fields](#)

Clip Language - Labels

A label is defined by placing a colon at the beginning of a line followed by the label name. The label name must be at least one character up to any length; it can have any character except word delimiters (space, tab, punctuation, etc.). Character case is ignored. The scope of a label is limited to the Clip in which it has been defined; you cannot jump to a label in another Clip. Example of a label:

`:Label`

The Clip syntax supports predefined label names which have a special meaning. If you define labels with the same name, you will not be able to use the predefined label for the purpose it was defined for. The following labels are predefined:

NEXT

tells the interpreter to execute the next instruction.

SKIP

will skip the next instruction (text and blank lines are counted, but not commented lines).

SKIP_xx *(added in v4.52)*

will skip xx lines of Clip code (including blank lines, but not commented lines). xx represents a number (positive or negative). For example, Skip_3 will skip the next three lines in the Clip. "Skip" and "Skip_1" have the same effect.

END

ends the execution of the current clip but continues normally in parent Clip (the Clip from which it was executed, if any).

EXIT

ends the execution of the current clip, including any parent Clip.

Labels are useful to skip instructions or repeat them when a certain condition is met. You can use the "**^!Goto**" command or one of the conditional testing commands to jump to a specific label.

Useful tip when testing Clip code:

You can use the "Ctrl+Alt" key combination to stop the execution of a Clip (especially helpful when stuck in a loop).

Commands:

^!Goto Label

Continues Clip execution from the specified label defined in the Clip.

^!IfLabelExist LabelName GoToLabelTrue [ELSE GoToLabelFalse] *(added in v4.52)*

Jumps to the specified label if the indicated label name exists in the active Clip.

Clip Language - Variables

Variables hold data on which a program operates. In the Clip text, the variable's name represents the value of data stored somewhere in the computer memory when the Clip is executed. The data can be anything from nothing to multiple megabytes of text, and can represent a string (a sequence of characters), an integer (negative and positive whole numbers), or a boolean (True/False) data type.

A variable name can be made up of any character except those that delimit words (space, tab, punctuation, carriage return, etc.). The name must also be bound by a percentage character "%". The left side "%" must be preceded with a circumflex accent "^" when you are referring to the value it contains. Example of a variable: ^%MyVariable%. In NoteTab, a variable is defined as soon as its name is encountered in a Clip during execution. A variable can be placed anywhere in a Clip text and can even be used to define a command or function name. If you haven't initially assigned a value to it, the variable is always empty at first.

Variables keep their value as long as the Clipbook is kept open. NoteTab recognizes two types of variables: local variables that exist only within the context of the library that created them (they are cleared if you change libraries or close the Clipbook), and permanent variables that keep their value even when you change libraries (but not if you close the Clipbook). Add the prefix "p_" to define a permanent variable (e.g. ^%p_KeepValue%). If you want to preserve a variable's value even after closing NoteTab, you can store it in an INI file by using the "!SaveValue" command and then restore it later by using the "\$GetValue()" function.

Use the "!Set" command to assign data to a variable. For example:

```
^!Set %VariableName%=Data
```

If %VariableName% was never used before, it will be created by the command and then receive the value assigned to it after the equal sign. If %VariableName% already exists, the above command will replace the value it contains with the new data. Note that you must not precede the variable name with a circumflex accent "^" when you assign or change its value.

You can also assign values to multiple variables using a single "!Set" command. An assignment must be separated from the next one with a semicolon and a blank character "; ". For example:

```
^!Set %V1%=Value1; %V2%=Value2; %V3%=Value3
```

Note that the last assignment does not end with a semicolon.

Note that you must use the curly braces format ^?{ } rather than the square brackets ^?[] if you want to assign the result from a Clip wizard memo field. Example:

```
^!Set %NewVariable%=^?{(T=M)Message}
```

To use the data contained in a variable, just precede the variable name (bound by "%" characters) with a circumflex accent: ^%VariableName% anywhere in the Clip where you want to use it. Note that you cannot define a label with a variable. So, for example, the following will not work (the line is just ignored):
:^%VariableName%

Note that you can set the value of a variable with the data contained in another variable by using the following instruction:

```
^!Set %NewVariable%=^%OtherVariable%
```

The Clipbook also provides predefined variables. Unlike other variables, they already contain a value when you first use them. You may, however, reassign other values to them if you wish. These are the predefined variables and the data they contain:

^%NL% = new line (CR/LF pair)

^%PAGE% = Page break
^%TAB% = Tab character
^%SPACE% = Blank space character
^%EMPTY% = empty string
^%VBAR% = vertical bar (|) *(added in v4.6)*
^%CARET% = caret (^) *(added in v4.6)*
^%COLON% = colon (:)
^%SC% = semicolon (;)
^%AR_KEY% = Auto-replace activation key *(added in v4.6)*
^%RESULT% = Used to assign result to custom functions *(added in v4.8)*

Commands and Functions:

^!Set %VariableName%=Value

or

^!Set %VariableName%=^%OtherVariable%

or

^!Set %V1%=Value1; %V2%=Value2; %V3%=Value3

Assigns Value to VariableName. The variable name may not contain blanks or other word-delimiting characters. The value can be a string, an integer, or a boolean data type. It can also refer to the value of another variable.

^!SetArray %VariableName%=ArrayData *(added in v4.6)*

Similar to the **^!Set** command but creates an array from the data assigned to it. By default, the data delimiter is a plain semi-colon (;), or a semi-colon and double-quotes separating each item. You can change the default delimiter with the command **^!SetListDelimiter** (see below). During the assignment, the individual items from the ArrayData are each stored in a new variable with the same name as the base variable plus an index number reflecting their corresponding position. The variable at index 0 has a special purpose as it stores the actual number of items contained in the array. You can change the values of the individual indexed items by using the **^!Set**, **^!Inc**, **^!Dec** commands, and the changes are reflected in the base variable. If you assign an empty value to an array variable, or use the **^!Set** command to assign a new value to it, the array is automatically removed from memory.

Examples:

^!SetArray %Array%="Value 1";"Value 2";"Value 3"

The above instruction will create four new variables:

^%Array0% = 3

^%Array1% = Value 1

^%Array2% = Value 2

^%Array3% = Value 3

If we assign a new value to the second item, as follows:

^!Set %Array2%=New Value

then accessing the base variable **^%Array%** will return:

"Value 1";"New Value";"Value 3"

The following assignment will destroy the array, and only the base variable will be preserved with the newly assigned value:

^!Set %Array%=Another Value

^!SetListDelimiter Value *(added in v4.6)*

Defines a delimiter to be used by the **^!SetArray** command or Clip Wizard fields using the new Checkbox list (see below). Tokens are accepted, but the delimiter must not be more than 10 characters. The scope of this command is limited to the current Clip. Assigning an empty value results in using the default semi-colon (;) delimiter. Example: **^!SetListDelimiter ^p^p**

^!Append %Variable%=Value

Appends value to the content of %Variable%. This command also has the advantage of allowing assignments of function names and ^?{} prompts without causing these to be evaluated. This command is particularly useful for creating sophisticated Clip Wizards.

^!ClearVariable %VariableName%

Empties the specified variable.

^!ClearVariables

Clears all variables (including those defined as permanent).

^!SetIniFile Name *(added in v4.51)*

Defines where Clip variables are saved when using the SaveValue, ClearValue, and ClearValues commands and the GetValue function. By default, variables are saved in NoteTab's INI file (or registry if so configured). Use this command to define a different INI file. If you do not specify a path in the name, the specified INI file will be created in the Libraries directory. If you plan to distribute your library file, consider giving your INI file the same name as your library file (without the extension); you can use the ^\$GetLibraryName\$ for this purpose. The scope of this command is limited to the library in which it is invoked. It is reset when you change libraries or close the Clipbook. Invoke the command with no name to reset NoteTab's INI file as the default.

^\$GetIniFile\$

Returns the name of the INI file defined by the SetIniFile command. Returns a blank string if NoteTab's INI file is used.

^!SaveValue [Section:]Key=Value

Saves the key and its value in NoteTab's INI file, or in the INI file defined by the ^!SetIniFile command. If a section name is not defined, the key entry is written under the [ClipValues] section. Section names are only supported since version 4.51. The maximum amount of text that can be saved this way is 1024 characters. Example:

^!SaveValue MySection:UserName=Raymond

^\$GetValue("[Section:]Key")\$

Reads the specified key in NoteTab's INI file, or in the INI file defined by the ^!SetIniFile command. If a section name is not defined, the key entry is searched under the [ClipValues] section. Section names are only supported since version 4.51. Returns the value assigned to the key. Returns an empty string if the key does not exist. Example based on ^!SaveValue example: **^\$GetValue(MySection:UserName)\$** -> Raymond

^!ClearValue [Section:]Key

Deletes the specified key in NoteTab's INI file, or in the INI file defined by the ^!SetIniFile command. If a section name is not defined, the key entry is searched under the [ClipValues] section. Section names are only supported since version 4.51.

^!ClearValues [Section]

Deletes all the keys under the specified section in NoteTab's INI file, or in the INI file defined by the ^!SetIniFile command. If a section name is not defined, the [ClipValues] section will be cleared. Section names are only supported since version 4.51.

^!ProgIniSave [IniName] *(added in v4.6)*

Saves the current program settings into the NoteTab ini file or registry. Optionally, you can specify an alternative INI file (it is automatically created if it does not exist).

^!ProgIniLoad [IniName] *(added in v4.6)*

Loads options from the NoteTab ini file or registry. Optionally, you can specify an alternative INI file (it must exist or else the command sets the error condition). Note that the [AutoLoad] section is ignored by this command.

^!Dec VariableName [nn]

Subtracts 1 from specified variable (decrement). If you use the second optional parameter "nn", you can specify a different decrement value. For example, if %var% contains the value 30 and you apply the command "^!Dec %var% 7" to it, the new value will be 23.

^!Inc VariableName [nn]

Adds 1 to specified variable (increment). If you use the second optional parameter "nn", you can specify a different increment value. For example, if %var% contains the value 23 and you apply the command "^!Inc %var% 7" to it, the new value will be 30.

Clip Language - Events

An event is an action that takes place on a system level – showing or resizing a window for example, or the movement of a mouse. The Clip language recognizes four distinct events:

1. the opening of a Clipbook library;
2. the closing of a Clipbook library;
3. the Windows Clipboard receiving text;
4. and timer events that are triggered at defined time intervals.

You can execute Clip instructions when these events are triggered. Timer events are controlled with the "**^!TimerPlay**" command; the three other events are defined by creating Clips with the corresponding event name (OnOpen, OnClose, OnClipboardChange). When the event is triggered, the corresponding Clip is executed.

H="OnOpen"

Automatically executes the OnOpen Clip when library is opened. Holding the Shift key down cancels the execution of the Clip.

H="OnClose"

Automatically executes the OnClose Clip when library is closed (either by changing to another library or by closing the Clipbook). Holding the Shift key down cancels the execution of the Clip.

H="OnClipboardChange"

Automatically executes content of the OnClipboardChange Clip whenever text is copied to the Windows Clipboard. Note that you must close libraries containing this type of Clip if you no longer want to capture text changes to the Clipboard.

Timer Commands:

^!TimerPlay ClipName

Assigns the Clip instructions to the timer memory. The instructions remain available even when you change libraries but are cleared when you close the Clipbook.

^!TimerStart nnnn

Defines an event timer. nnnn represents the interval in milliseconds. For example, "**^!TimerStart 5000**" will execute the assigned Clip (set with "**^!TimerPlay**") every 5 seconds. Note that the Clip is not executed if the Clipbook is already in the process of executing another Clip.

^!TimerStop

Stops the timer and clears the assigned Clip instructions.

Clip Language - Custom Functions

Since NoteTab 4.8, you can create your own Clip functions, using an approach that is similar to the `^!Clip` command. Custom Clip functions must not have the same name as a predefined function. The function name corresponds to the name of the Clip that contains the code. You can create functions that take one parameter or none. You can access the parameter data in your function script by using the `^&` code. Your function script must assign the result to the new `^!Result` command, or by using the new predefined variable `%Result%`. Example:

```
H="Double"  
^!IfTrue ^$IsNumber("^&")$ Multiply  
^!Result ^&^&  
^!Goto End  
:Multiply  
^!Result ^$Calc(^& * 2)$
```

```
H="TestDouble"  
^!Prompt ^$Double(^?[Enter a number or some text])$
```

Note that Clips using custom functions cannot be run from the Clipbar.

Clip Language - Testing Conditions

The Clip language includes many commands and functions to test conditions within the NoteTab environment. You can use these tests to control which instructions are executed based on one condition or another. See below the commands and functions that are available for controlling the operational flow:

Note that NoteTab Light 4.6 and earlier (with trial mode not enabled) only supports the predefined label "Exit".

New since NoteTab 4.8: conditional commands now accept a command statement instead of a label. The second command statement may not be another conditional command, and it may not be followed by an ELSE-type label. Examples:

```
^!If ^%Variable% < 0 ^!Set %Variable%=0
```

```
^!IfFileExist "%Variable%" GoToLabelTrue ELSE ^!Prompt File does not exist
```

Commands:

```
^!If Integer1 <=> Integer2 GoToLabelTrue [ELSE GoToLabelFalse]
```

or

```
^!If "String1" <=> "String2" GoToLabelTrue [ELSE GoToLabelFalse]
```

If the comparison result is true, then jumps to the specified GoToLabelTrue label. If the result is false and the optional ELSE keyword follows the first label name, execution jumps to the specified GoToLabelFalse label; or to the next instruction if the "ELSE" label is not used. If the two values represent numbers then they are compared as numbers, otherwise they are compared as strings. String comparison is case sensitive.

You can use the following comparison operators:

- < returns True if left value is smaller than right value
- > returns True if left value is greater than right value
- <> returns True if left value is different from right value
- <= returns True if left value is smaller or equal to right value
- >= returns True if left value is greater or equal to right value
- = returns True if left value is equal to right value

Examples:

The following will cause the execution to jump to the label defined as IsTrue:

```
^!If "Apple" <> "Orange" IsTrue ELSE IsFalse
```

The following will cause the execution to jump to the label defined as LabelFalse:

```
^!If 16 < 7 LabelTrue ELSE LabelFalse
```

```
^!IfSame "Value1" "Value2" GoToLabelTrue [ELSE GoToLabelFalse]
```

Compares two string values. The comparison is not case sensitive.

Example:

The following will cause the execution to jump to the label defined as GoToLabelTrue:

```
^!IfSame "abc" "ABC" GoToLabelTrue
```

```
^!IfDiff "Value1" "Value2" GoToLabelTrue [ELSE GoToLabelFalse]
```

Compares two string values. The comparison is not case sensitive.

Example:

The following will cause the execution to skip the next instruction (Skip is a predefined label):

```
^!IfDiff "abc" "xyz" Skip
```

```
^!IfMatch "RegExp" "Value2" GoToLabelTrue [ELSE GoToLabelFalse] (added in v4.8)
```

Similar to ^!IfSame, but accepts a regular expression for the first value.

^!IfTrue Value GoToLabelTrue [ELSE GoToLabelFalse]

If Value is equal to 1, TRUE, YES, Y, or ON (case insensitive), then jumps to specified label name. Do not use comparison operators with this command; in that case, you should use the ^!If command to do a comparison between two values.

^!IfFalse Value GoToLabelTrue [ELSE GoToLabelFalse]

If Value is an empty string or equal to 0, FALSE, NO, N, or OFF (case insensitive), then jumps to specified label name. Do not use comparison operators with this command; in that case, you should use the ^!If command to do a comparison between two values.

^!IfInRange Value Range1..Range2 GoToLabelTrue [ELSE GoToLabelFalse]

Applies only to numerical values. If Value is in the range defined from Range1 to Range2, then jumps to GoToLabelTrue. For example: "**^!IfInRange 23 20..30 DoThis**" will jump to the label ":DoThis"

^!IfInDateRange yyyy/mm/dd-hh:nn yyyy/mm/dd-hh:nn..yyyy/mm/dd-hh:nn GoToLabelTrue [ELSE GoToLabelFalse]

Similar to IfInRange command but takes dates as arguments instead of integers. The date value can have the following formats (yyyy=year, mm=month, dd=day, hh=24 hour, nn=minutes):

yyyy/mm/dd-hh:nn

1998/07/13-16:30

yyyy/mm/dd

1998/07/13

hh:nn

16:30

^!IfError GoToLabelTrue [ELSE GoToLabelFalse]

If the previous command fails for one reason or another, it sets an error state that can be detected with "^!IfError". If the previous command failed, then execution will jump to specified label. Note that support for ELSE was added in NoteTab v4.51.

^!IfCancel GoToLabel

Jumps to specified label if a Clip Wizard built from "^{?}" fields has been canceled. If the Clip doesn't contain such a statement or if it does not appear after the call to the Clip Wizard and before the next one, then its execution is aborted.

^!Continue [Question]

Displays a message box with an OK and Cancel button. If the user clicks on the OK button, the Clip execution continues normally. If the Cancel button is pressed then Clip execution is halted at that point. NoteTab checks if this instruction is the first to execute in a Clip; if this is the case it will be displayed before the main Clip Wizard (if any).

^!Skip Question

Skips next instruction if the user answers Yes.

^!IfLabelExist LabelName GoToLabelTrue [ELSE GoToLabelFalse] *(added in v4.52)*

Jumps to the specified label if the indicated label name exists in the active Clip.

^!IfClipExist "ClipName" GoToLabelTrue [ELSE GoToLabelFalse]

Jumps to label if the specified Clip exists.

^!IfFileExist "FileName" GoToLabelTrue [ELSE GoToLabelFalse]

Jumps to specified label if FileName exists.

^!IfPathExist "PathName" GoToLabelTrue [ELSE GoToLabelFalse]

Jumps to specified label if PathName exists.

^!IfFileAttr "FileName" Attr GoToLabelTrue [ELSE GoToLabelFalse] *(added in v4.8)*

Compares FileName attributes with Attr. If Attr is equal or a subset of FileName's attributes, then GoToLabelTrue is activated. Attr can represent multiple values: A = Archive, D = Directory, H = Hidden, R = Read-only, S = System, V = VolumeID, * = All attributes. Example:

```
^!IfFileAttr "c:\Scandisk.log" AH LabelTrue ELSE LabelFalse
```

If Scandisk.log has at least the Archive and Hidden attributes set, then the instruction jumps to LabelTrue.

^!IfAppOpen AppTitle GoToLabelTrue [ELSE GoToLabelFalse] *(added in v4.52)*

Jumps to the specified label if a window with the indicated title caption is found. For example, if the Windows calculator utility is open, the following command will continue execution after the CalcOpen label:

```
^!IfAppOpen Calculator CalcOpen
```

Most applications usually show other information in the title bar. For these cases, you can use an asterisk "*" either at the beginning or the end of the AppTitle text to specify that the text should match the end or the beginning of the title respectively. For example, MS Wordpad displays the document name in front of the program name, so the following instruction will work if Wordpad is open:

```
^!IfAppOpen *WordPad WordPadOpen
```

Functions:

^\$IsEmpty(Value)\$

Returns 1 if the value/variable is empty and 0 if it contains data.

^\$IsHex(Value)\$ *(added in v4.52)*

Returns 1 if specified text is a hexadecimal number and 0 if it is not.

^\$IsLowercase("Str")\$ *(added in v4.8)*

Returns 1 if Str does not contain any uppercase characters, and 0 if it does.

^\$IsUppercase("Str")\$ *(added in v4.8)*

Returns 1 if Str does not contain any lowercase characters, and 0 if it does.

^\$IsCapitalized("Str")\$ *(added in v4.8)*

Returns 1 if Str does not start with a lowercase character and is not followed by any uppercase characters, and 0 if this condition is not met.

^\$IsMixedCase("Str")\$ *(added in v4.8)*

Returns 1 if Str contains both lowercase and uppercase characters, and 0 if it does not. Note that text that is capitalized is not considered mixed case.

^\$IsAlpha("Str")\$ *(added in v4.8)*

Returns 1 if Str contains only characters from the alphabet, and 0 if it does not.

^\$IsAlphaNumeric("Str")\$ *(added in v4.8)*

Returns 1 if Str contains only numbers and characters from the alphabet, and 0 if it does not.

^\$IsNumber(Value)\$

Returns 1 if specified text is a number and 0 if it is not.

^\$IsBlank("Str")\$ *(added in v4.8)*

Returns 1 if Str contains only blank characters (tabs, spacebar, line breaks, etc.), and 0 if it does not.

^\$IsUrl("AnyText")\$

Tests if parameter is a URL or not. It returns 1 to indicate True and 0 to indicate False.

^\$IsWildcard("FileName")\$ *(added in v4.82)*

Returns 1 if FileName contains one or more wildcard characters (* and ?), and 0 if it does not.

^\$IsOpen("DocumentName")\$

Returns 1 if specified document is open in NoteTab and 0 if it is not.

^\$IsReadOnly\$

Returns 1 if current document is Read-Only, or 0 if it is not.

^\$IsModified\$

Returns 1 if current document needs saving, or 0 if it is not modified.

^\$IsWordWrap\$

Returns 1 if word wrap is active in current document, or 0 if turned off.

^\$IsAutoIndent\$

Returns 1 if AutoIndent is enabled in current document and 0 if it is turned off.

^\$IsPasteIndent\$ *(added in v4.8)*

Returns 1 if Clips insert text with automatic indenting and 0 if not.

^\$IsTrimBlanks\$ *(added in v4.6)*

Returns 1 if the option "Trim White Spaces" is enabled in current document and 0 if it is turned off.

^\$IsHtmlDoc\$

Returns 1 if current document is HTML type document, or 0 if it is not.

^\$IsOutlineDoc\$

Returns 1 if current document is Outline type document, or 0 if it is not.

^\$IsShowHeadings\$ *(added in v4.52)*

Returns 1 if headings are displayed in the outline document window and 0 if they are not.

^\$IsPasteBoard\$

Returns 1 if current document is the Paste Board, or 0 if it is not.

^\$IsSecondWindow\$

Returns 1 if current document is in the second window, or 0 if it is not.

^\$IsClipDoc\$

Returns 1 if current document is the [Clip] editor, or 0 if it is not.

^\$IsFileType("FileName";"extensions")\$

Returns 1 if the extension of FileName is listed in "extensions", or 0 if turned off. Multiple extensions must be delimited with a comma. The following example returns 1:

^\$IsFileType("c:\temp\MyFile.txt";"html,txt,otl,ini")

^\$IsCtrlKeyDown\$ *(added in v4.8)*

Returns 1 if the Control key is pressed while the function is executed and 0 if it is not.

^\$IsShiftKeyDown\$ *(added in v4.8)*

Returns 1 if the Shift key is pressed while the function is executed and 0 if it is not.

^\$IsAltKeyDown\$ *(added in v4.8)*

Returns 1 if the Alt key is pressed while the function is executed and 0 if it is not.

^\$IsTopStyleEnabled\$ *(added in v4.8)*

Returns 1 if the TopStyle editor is available through NoteTab, and 0 if it is not.

^\$IsFullVersion\$

Returns 1 if NoteTab is the commercial version and 0 if it is the Freeware or trial version.

^\$IsLightVersion\$

Returns 1 if the NoteTab program currently running is the Freeware version (trial mode not enabled), or 0 if it is not.

Clip Language - Message Boxes / Prompts

Use message boxes to prompt users for an answer or to inform them of what is being done. See also [Wizard Dialog and Fields](#) for related information.

^!Info [Alignment:L|C|R]Multiline message

Displays a Web-enabled scrollable message window. The text can be up to about 40 KB in size. Users can double-click on links or e-mail addresses in the text to open them in their browser or e-mail program respectively. Authors of Clipbook libraries can use this feature to advertise their work. Use the token "^P" to indicate line breaks in your text. By default, lines displayed in the window are centered. You can use the alignment parameter (character between square brackets) to change the setting:

[L] = Left aligned

[C] = Centered

[R] = Right aligned

Users can change the default font used for the text by right clicking in the text area to open the shortcut menu, then choosing the Font command to open the font dialog box.

The following example will display a dialog box with two left-aligned lines:

^!Info [L]You can find more Clipbook libraries from^Pwww.notetab.com/libraries.htm

Other examples are provided in the libraries packaged with NoteTab.

^!Prompt [Message]

Displays a message box with the specified text (optional). The user must click on the OK button to continue Clip execution.

^!Continue [Question]

Displays a message box with an OK and a Cancel button. If the user clicks on the OK button, the Clip execution continues normally. If the Cancel button is pressed then Clip execution is halted at that point. NoteTab checks if this instruction is the first to execute in a Clip; if this is the case it will be displayed before the main Clip Wizard (if any).

^!Skip [Question]

Skips next instruction if the user answers Yes.

^!SetDebug [On|Off] (added in v4.8)

When enabled, a dialog box pops up just before each statement and paste operation is executed. The dialog box shows the parsed instruction or the text about to be pasted in the document. There are three buttons; "Yes" is for continuing execution until the next instruction, "No" disables Debug mode, and "Cancel" ends Clip execution.

^!SetHintInfo AnyText

Turns the status bar Clip messages on or off. When you specify text, it will be displayed in the status bar instead of the usual Clip messages. To re-enable the display of normal Clip messages, just invoke the command without any text. This command only has an effect during the time the Clip is running.

^!StatusShow Message

Shows a modeless status window. Useful for providing feedback to the user during long processes. If the Window is already open, the current message is replaced by the new one. The window is automatically closed when the Clip ends.

^!StatusClose

Closes the modeless status window opened with the ^!StatusShow command.

Clip Language - Text Processing

Use the following functions to analyze or modify text data.

^\$IsLowercase("Str")\$ *(added in v4.8)*

Returns 1 if Str does not contain any uppercase characters, and 0 if it does.

^\$IsUppercase("Str")\$ *(added in v4.8)*

Returns 1 if Str does not contain any lowercase characters, and 0 if it does.

^\$IsCapitalized("Str")\$ *(added in v4.8)*

Returns 1 if Str does not start with a lowercase character and is not followed by any uppercase characters, and 0 if this condition is not met.

^\$IsMixedCase("Str")\$ *(added in v4.8)*

Returns 1 if Str contains both lowercase and uppercase characters, and 0 if it does not.

^\$IsAlpha("Str")\$ *(added in v4.8)*

Returns 1 if Str contains only characters from the alphabet, and 0 if it does not.

^\$IsAlphaNumeric("Str")\$ *(added in v4.8)*

Returns 1 if Str contains only numbers and characters from the alphabet, and 0 if it does not.

^\$IsNumber(Value)\$

Returns 1 if specified text is a number and 0 if it is not.

^\$IsBlank("Str")\$ *(added in v4.8)*

Returns 1 if Str contains only blank characters (tabs, spacebar, line breaks, etc.), and 0 if it does not.

^\$IsHex(Value)\$ *(added in v4.52)*

Returns 1 if specified text is a hexadecimal number and 0 if it is not.

^\$StrSize("Str")\$

Returns the size of Str. Example:

^\$StrSize("abcdefghi")\$ -> 9

^\$StrLower("Str")\$

Returns Str with all characters in lower case. Example:

^\$StrLower("ABCdefGHI")\$ -> abcdefghi

^\$StrUpper("Str")\$

Returns Str with all characters in upper case. Example:

^\$StrUpper("abcdefghi")\$ -> ABCDEFGHI

^\$StrCapitalize("Str")\$ *(added in v4.8)*

Capitalizes all words in Str and returns the resulting string.

^\$StrInvertCase("Str")\$ *(added in v4.8)*

Converts all uppercase characters to lowercase, and lowercase characters to uppercase, and returns the resulting string.

^\$StrSentenceCase("Str")\$ *(added in v4.8)*

Capitalizes all words following a period, exclamation, and question mark and converts all other words to

lowercase.

^\$StrTrim("Str")\$

Removes blank spaces from the start and end of the specified value. Example:

`^$StrTrim(" some text ")$` returns "some text"

^\$StrCopy("Str";Index;Amount)\$

Returns a substring of Str starting at position Index and a size determined by Amount. Example:

`^$StrCopy("abcdefghi";4;3)$` -> def

^\$StrCopyRight("Str";Amount)\$ *(added in v4.52)*

Similar to `^$StrCopy()` but copies the substring from the end of the specified string.

^\$StrCopyLeft("Str";Amount)\$ *(added in v4.82)*

Similar to `^$StrCopy()` but copies the substring from the start of the specified string.

^\$StrInsert("SubStr";"Str";Index)\$

Returns a string composed of SubStr inserted at position Index in Str. Example:

`^$StrInsert("xyz";"abcdefghi";3)$` -> abxyzcdefghi

^\$StrDelete("Str";Index;Amount)\$

Returns the resulting string with a deleted portion starting at position Index and a size determined by Amount. Example:

`^$StrDelete("abcdefghi";4;3)$` -> abcghi

^\$StrDeleteRight("Str";Amount)\$ *(added in v4.82)*

Similar to `^$StrDelete()` but deletes the text from the end of the specified string.

^\$StrDeleteLeft("Str";Amount)\$ *(added in v4.82)*

Similar to `^$StrDelete()` but deletes the text from the start of the specified string.

^\$StrIndex("Str";Index)\$

Returns the character at position Index in Str. Returns an empty string if an invalid Index value is specified. Example:

`^$StrIndex("abcdefghi";4)$` -> d

^\$StrPos("SubStr";"Str";CaseSensitive)\$

Returns the index position of SubStr in Str. Use the CaseSensitive parameter to determine if the match should be case sensitive or not. Returns 0 if there is no match. Tab and line-break tokens are automatically converted to their real characters since version 4.6. Example:

`^$StrPos("def";"abcdefghi";True)$` -> 4

^\$StrPosRight("SubStr";"Str";CaseSensitive)\$ *(added in v4.52)*

Similar to the `^$StrPos()` function but searches from right to left.

^\$StrPosMid("SubStr";"Str";StartIndex;CaseSensitive)\$ *(added in v4.52)*

Similar to the `^$StrPos()` function but starts searching from the specified index position.

^\$StrFill("Str";Amount)\$

Returns a string entirely composed of the defined character or text repeated Amount times. Examples:

`^$StrFill(+;4)$` -> ++++

`^$StrFill("<>";4)$` -> <><><><>

^\$StrCount("SubStr";"Str";CaseSensitive;WholeWord)\$

Returns the number of occurrences of SubStr in Str. Use the CaseSensitive parameter to determine if the match should be case sensitive or not, and WholeWord to indicate that SubStr must only match whole

words. Example:

```
^$StrCount("t";"NoteTab";False;False)$ -> 2
```

^\$StrReplace("SubStr";"ReplaceStr";"Str";CaseSensitive;WholeWord)\$

Returns a new string with all occurrences of SubStr replaced with ReplaceStr according to the settings of the CaseSensitive and WholeWord parameters. Tab and line-break tokens are automatically converted to their real characters since version 4.6. Example:

```
^$StrReplace("o";"?";"NoteTab Pro";True;False)$ -> N?teTab Pr?
```

^\$StrSplit("Str";TextWidth;TrimLines)\$ *(added in v4.6)*

Returns the "Str" string with long lines split at specified width. TextWidth defines the maximum width of the text in each line; TrimLines is either True or False and determines if blank spaces at the end of each line are removed or not.

^\$StrFormat("Str";LeftMargin;TextWidth;Justify)\$

Returns the "Str" string in specified format. LeftMargin determines the number of blank spaces to insert between the left margin and the text; TextWidth defines the maximum width of the text in each line; Justify is either True or False and determines if the text is left and right justified. Prior to version 4.8, this command was only available in the commercial and trial versions of NoteTab.

^\$StrSort("Str";CaseSensitive;Ascending;RemoveDuplicates)\$ *(added in v4.52)*

Returns the specified text "Str" sorted according to the defined criteria. For example, the following instruction will sort the lines contained in the Clipboard, in ascending order, ignoring character case, and removing duplicates:

```
^$StrSort("^$GetClipboard$";False;True;True)$
```

^\$StrAnsiToAscii("Str")\$ *(added in v4.8)*

Returns Str with all extended ANSI characters converted to ASCII.

^\$StrAsciiToAnsi("Str")\$ *(added in v4.8)*

Returns Str with all extended ASCII characters converted to ANSI.

^\$StrFixLines("Str")\$ *(added in v4.51)*

Corrects carriage returns in text criteria (Unix and Mac-formatted text is converted to Dos/Windows format).

^\$StrStripHTML("Str";PreserveURLs)\$ *(added in v4.82)*

Returns a copy of Str with all HTML tags removed. PreserveURLs is either True or False, and determines if URLs are preserved or not.

^\$CharToDec(Character)\$ *(added in v4.8)*

Returns the decimal value of Character. Example:

```
^$CharToDec(S)$ -> 83
```

^\$DecToChar(Decimal)\$ *(added in v4.8)*

Returns the character corresponding to the decimal value. Example:

```
^$DecToChar(83)$ -> S
```

Note that you should avoid converting the following decimal values 1, 4, 5, and 6 to characters as they have a special meaning for the Clip interpreter.

^\$HexToInt(Value)\$ *(added in v4.52)*

Returns the specified hexadecimal value as an integer.

^\$IntToHex(Value)\$ *(added in v4.52)*

Returns the specified integer value as a hexadecimal number. In version 4.8, also returns the ANSI hex code of single character if Char does not represent a number.

^\$UseTokens("Str")\$

Converts control characters (tabs, carriage returns, etc.) to their corresponding tokens.

^\$RemoveTokens("Str")\$

Converts tokens to their corresponding control characters (tabs, carriage returns, etc.)

^!DirectPrint "Title" Data *(added in v4.8)*

Prints text directly to the printer, bypassing NoteTab's print procedure. Uses default printer settings. Title is the text shown in the Print manager and Data is the text to send directly to the printer. Data can contain printer control codes.

Clip Language - HTML Processing

Use the following commands and functions to process HTML code.

^!InsertHtml Any text with tokens (^p/^t) *(added in v4.8)*

Similar to ^!InsertText command, but converts HTML tags to Uppercase, Lowercase, or XHTML format (depending on settings in Options). This command is sensitive to the "Indent on Paste" setting in the Options dialog box on the Clipboard tab. You can precede it with the command

^!SetPasteIndent Off

to make it ignore the indent setting.

^!InsertWizardHtml Any text with tokens (^p/^t) *(added in v4.8)*

Similar to ^!InsertHtml command, but removes attributes that have no value. This command is useful when used together with the Clip wizard. It allows the user to leave certain fields blank in the Wizard, and the corresponding tags are then automatically removed before being inserted in the document. Example:

^!InsertWizardHtml <TD ALIGN="" WIDTH="">

will insert the following tag in your document: <TD>

^!RunTidy [i|d] *(added in v4.8)*

^!RunTidy [Switches]

HTML Tidy is a free utility that cleans HTML code and fixes errors. This new Clip command will send your document text to the HTML Tidy program. The corrected output returned by HTML Tidy is then inserted into the document. The StdError stream is saved in a file called "StdError.err" in NoteTab's application directory, unless another file is specified with the SetStdError Name command. You can use the optional parameter "Switches" to specify Tidy command-line arguments; example:

^!RunTidy -f errs.txt -qiu

If no argument is specified and you have a Tidy.cfg file in the NoteTab folder, then HTML Tidy will be configured according to the settings in that configuration file. If NoteTab does not find the configuration file, it will determine the appropriate switches based on the settings in NoteTab. If you have a Tidy.cfg file, but you want NoteTab to ignore it and use default settings, then use the "d" argument (without the quotes). If you want NoteTab to determine the best settings, but would like to have your tags indented, just use the "i" argument; example:

^!RunTidy i

You can learn more about HTML Tidy and download the latest version by visiting the following Web page:

<http://www.w3.org/People/Raggett/tidy/>

The easiest setup is to install the Tidy.exe program file in the NoteTab folder, but you may place it anywhere else if you prefer.

^\$GetTidyExe\$ *(added in v4.8)*

Returns full path and name of HTML Tidy application.

^\$GetHtmlTagName("TagStr";Format)\$ *(added in v4.8)*

Returns the name of the HTML Tag, either in uppercase or lowercase. If the Format parameter is not specified, the character case is determined by the settings in NoteTab's Options. Use the Format parameter to control the format of HTML tags. It can have one of the following values: UPPERCASE, LOWERCASE, or XHTML. Example:

^\$GetHtmlTagName("<TD ALIGN='left' WIDTH='250'>";LOWERCASE)\$ -> td

^\$GetHtmlTagAttr("TagStr";Attr;Format)\$ *(added in v4.8)*

Returns the value of the specified HTML attribute. If the Format parameter is not specified, the result is determined by the settings in NoteTab's Options. Use the Format parameter to control the case of minimized attributes (it has no effect on the value returned for normal attributes). It can have one of the following values: UPPERCASE, LOWERCASE, or XHTML. Example:

`^$GetHtmlTagAttr("<DL COMPACT>;Compact;XHTML")$ -> compact`
`^$GetHtmlTagAttr("<DL COMPACT>;Compact;UPPERCASE")$ -> COMPACT`
`^$GetHtmlTagAttr("<TD ALIGN='left' WIDTH='250'>;Width")$ -> 250`

`^$GetHtmlTagTidy("TagStr";Format)$` *(added in v4.8)*

Returns a properly formatted HTML tag converted to uppercase, lowercase, or XHTML format. If the Format parameter is not specified, the result is determined by the settings in NoteTab's Options. Use the Format parameter to control the format of HTML tags. It can have one of the following values:

UPPERCASE, LOWERCASE, or XHTML. Example:
`^$GetHtmlTagTidy("<HR SIZE=2 NOSHADE>;XHTML")$ -> <hr size="2" noshade="noshade" />`

`^$GetHtmlTextTidy("Str";Format)$` *(added in v4.8)*

Returns the string with HTML tags converted to Uppercase, Lowercase, or XHTML format. If the Format parameter is not specified, the result is determined by the settings in NoteTab's Options. Use the Format parameter to control the format of HTML tags. It can have one of the following values: UPPERCASE, LOWERCASE, or XHTML.

`^!SetHtmlFormat [UPPERCASE|LOWERCASE|XHTML]` *(added in v4.801)*

Sets the default format of HTML tags created by NoteTab.

`^$GetHtmlFormat$` *(added in v4.801)*

Returns the default format used for creating HTML tags. Possible values are: Uppercase, Lowercase, XHTML.

`^$StrToHtml("Str";ConvertLineBreaks;ConvertEntities)]$` *(added in v4.83)*

Converts plain text to HTML format. "Str" represents the text you want to convert. Use False for ConvertEntities if you don't want special characters (e.g. ©) to be converted to HTML entities (e.g. ©); the default value when omitted is True. Use False for ConvertLineBreaks if you don't want <p> and
 tags to be inserted at line breaks in the text; the default value when omitted is True.

TopStyle

NoteTab now supports Bradsoft's [TopStyle CSS editor](http://www.bradsoft.com/redir.asp?id=130), version 1.51 and above -- a program for editing Cascading Style Sheets. You can learn more about this useful program and download the latest version by visiting the following Web page:

<http://www.bradsoft.com/redir.asp?id=130>

Note that the TopStyle commands described below are used in the TopStyle library packaged with NoteTab. There is also a convenient TopStyle Clipbar to access the Clip commands.

`^$IsTopStyleEnabled$` *(added in v4.8)*

Returns 1 if the TopStyle editor is available through NoteTab, and 0 if it is not.

`^$EditStyleBlock("Str")$` *(added in v4.8)*

Opens TopStyle in <style>..</style> block editing mode. Returns the updated style block. Note that the "Str" argument should not include the surrounding <style>..</style> tags.

`^$EditStyleAttr("Str")$` *(added in v4.8)*

Opens TopStyle in inline (STYLE attribute) editing mode. Returns the updated style. Note that the contents of the style attribute should only include the value assigned to the style attribute. For example, if you wish to edit the style attribute in the tag , then you should only pass color:blue to this function.

`^!EditStyleSheet [FileName]` *(added in v4.8)*

Opens the specified style sheet file in TopStyle. If no file name is specified, this command will launch

TopStyle without opening a specific style sheet. Note that unlike the EditStyleBlock and EditStyleAttr functions, which block the calling application until the user closes TopStyle, EditStyle Sheet simply executes TopStyle and returns immediately.

Clip Language - Select, Copy, and Insert Text

Use the following commands and functions to select text in the editor, move or retrieve the cursor position, copy text to a variable, and insert or paste text into NoteTab or another application. See also [Clipboard Copy / Paste](#), [Find / Replace](#), and [Keyboard Macros](#) for related topics.

Commands:

^!SetCursor Row:Col

Places cursor at specified row and column.

^!SetCursorPara Row:Col

Similar to ^!SetCursor command but treats long word-wrapped lines as single lines. Both functions have the same effect if word wrap is turned off.

^!MoveCursor +/-nnn

Moves relative position of cursor in document by nnn steps. A positive value moves the cursor towards the end of the document and a negative value towards the start. Note that a line break represents two characters.

^!Jump nnn

or

^!Jump +/-nnn

or

^!Jump [LINE_START|LINE_END][SELECT_START|SELECT_END][TEXT_START|TEXT_END]

Moves cursor to the indicated position. If the parameter is a number without a sign (+/-), the command will treat it as the line position from the start of the document. When a sign is used, the command will move the cursor a relative number of lines from the current position; a positive number moving it towards the end of the text and a negative number towards the start. If you don't specify a line number, you can use one of the following settings:

LINE_START or **LINE_END**: places cursor at the beginning or end of the current line. This feature treats a line as if word wrap is turned off.

SELECT_START or **SELECT_END**: places cursor at the beginning or end of the selection and unselects the text.

TEXT_START or **TEXT_END**: moves to start or end of a plain document or the content of an Outline heading.

DOC_START or **DOC_END**: moves to start or end of any kind of document.

^!Select nnn

or

^!Select +/-nnn

or

^!Select [WORD(s)][LINE(s)][PARAGRAPH(s)][URL][FILENAME][HYPERLINK][ALL][BOL][EOL]

^!Select [EMAIL][HTMLTAG] *(added in v4.8)*

Selects text in the document. Since version 4.8, setting the size to 0 removes the selection. If the parameter is a number without a sign (+/-), the command will treat it as the number of lines to select from the cursor position. When a sign is used, the command will increase or decrease the size of the selection (characters) by the amount specified. If you don't specify a size value, you can use one of the following settings:

WORD(s): selects the word at the cursor position. If several words are partially selected, WORDS extends the selection to include a whole word at each end.

LINE(s): selects the line at the cursor position. If several lines are partially selected, LINES extends the selection to include a whole line at each end.
PARAGRAPH(s): selects the paragraph at the cursor position. If several paragraphs are partially selected, PARAGRAPHS extends the selection to include a whole paragraph at each end.
URL: selects a URL at the cursor position, if any.
FILENAME: selects a file name at the cursor position.
HYPERLINK: selects a NoteTab hyperlink at the cursor position.
ALL: selects all the text in the document window.
BOL: extends the selection to the beginning of the line.
EOL: extends the selection to the end of the line.
EMAIL: extends the selection of an e-mail address under the cursor (*added in v4.8*)
HTMLTAG: selects a complete HTML tag (*added in v4.8*)

^!SelectTo row:col

Extends the selection from the current cursor position to the coordinates defined by row and col.

^!SetFocus [Clipbook|ClipEdit|Document|Frame1|Frame2|List_Clip|List_Disk|List_Docs|List_Favs|Outline|PasteBoard]

Moves the focus to the specified screen object. This command will fail if the NoteTab window is minimized. Example to move the focus to the Outline-headings window:

^!SetFocus Outline

^!SetBookmark n

Only NTP: sets a bookmark at the current cursor position. "n" should be a value between 0 and 9.

^!GoToBookmark n

Only NTP: moves the cursor to the specified bookmark. "n" should be a value between 0 and 9.

^!ClearBookmarks

Only NTP: clears all bookmarks.

^!SetPasteIndent [On|Off] (*added in v4.8*)

Enables or disables automatic indenting when a Clip inserts text into a document. The scope of this command is limited to the Clip (and eventual sub-Clips) where it was invoked. When a Clip ends, the original user setting is automatically re-enabled.

^!InsertText Any text with tokens (^p/^t)

Inserts the text following the command into the document. Since the text must hold on the same line as the command, you will have to use the ^p token to indicate line breaks. Since version 4.8, this command is sensitive to the "Indent on Paste" setting in the Options dialog box on the Clipbook tab. Inserted text remains selected when the ^& token is used to add text to the criteria. You can precede it with the command

^!SetPasteIndent Off

to make it ignore the indent setting.

^!InsertFile FileName

Inserts the content of the specified file "FileName" at the cursor position.

Functions:

^\$GetCol\$

Returns column index of cursor in current document. Since v4.6: If some text is selected, then the beginning of the selection determines the column index.

^\$GetColScreen\$ (*added in v4.6*)

Returns the actual column position on the screen. Unlike **^\$GetCol\$**, this function takes into account expanded tabs. This function is only meaningful in NoteTab Pro; NoteTab Light/Std return the same result as the **^\$GetCol\$** function.

^\$GetRow\$

Returns line index of cursor in current document. Since v4.6: If some text is selected, then the beginning of the selection determines the line index.

^\$GetColStart\$ *(added in v4.8)*

Returns the cursor column index in the current document. If some text is selected, then the beginning of the selection determines the column position.

^\$GetRowStart\$ *(added in v4.8)*

Returns the cursor line index in the current document. If some text is selected, then the beginning of the selection determines the line index.

^\$GetColEnd\$ *(added in v4.8)*

Returns the cursor column index in the current document. If some text is selected, then the end of the selection determines the column position.

^\$GetRowEnd\$ *(added in v4.8)*

Returns the cursor line index the in current document. If some text is selected, then the end of the selection determines the line index.

^\$GetColLeft\$ *(added in v4.8)*

Returns the leftmost column index visible in current document window.

^\$GetRowTop\$ *(added in v4.8)*

Returns the index of the top line visible in current document window.

^!SetView RowTop:ColLeft *(added in v4.8)*

Moves the top line and leftmost column to the specified coordinates in current document window.

^\$GetSelSize\$

Returns size of selection in current document.

^\$GetSelection\$

Returns the text contained in the document selection.

^\$GetChar\$

or

^\$GetChar(Index)\$

or

^\$GetChar(+/-Index)\$

Returns the character at the cursor position if no parameter is specified. If the parameter is used, it returns the character at the position defined by Index. Use a + or - sign to indicate a relative position with respect to the cursor position.

^\$GetWord\$

^\$GetWord[(SelectText)]\$ *(added in v4.8)*

Returns the word at cursor position. Version 4.8: added the optional Boolean argument "SelectText" to make NoteTab select the corresponding block of text.

^\$GetBlock\$

^\$GetBlock[(SelectText)]\$ *(added in v4.8)*

Returns a block of text at cursor position. Similar to **^\$GetWord\$** but only uses tabs and blank spaces as

a delimiter. This function can, for example, retrieve file names, URLs, etc. Version 4.8: added the optional Boolean argument "SelectText" to make NoteTab select the corresponding block of text.

^\$GetHtmlTag[(SelectText)]\$ *(added in v4.8)*

Returns the HTML tag at cursor position. Use the optional Boolean argument "SelectText" to make NoteTab select the corresponding block of text.

^\$GetQuoteBlock[(SelectText)]\$ *(added in v4.8)*

Returns the block of text between double quotes (") at the cursor position. Use the optional Boolean argument "SelectText" to make NoteTab select the corresponding block of text. This function is especially useful for retrieving attribute values within HTML tags.

^\$IsBookmark(Index)\$

Determines if specified bookmark exists in the active document. Returns 1 if the bookmark number is defined and 0 if it is not used.

Clip Language - Clipboard Copy / Paste

Use the following commands and functions to use the Windows Clipboard and exchange text with other applications.

^\$GetClipboard\$

Returns the text contained in the Clipboard.

^\$GetClipboardSize\$

Returns the size of text contained in the Clipboard.

^!SetClipboard Any text with tokens (^p/^t)

Copies the text following the command into the Clipboard. Since the text must hold on the same line as the command, you will have to use the ^p token to indicate line breaks.

^!ClipboardSave *(added in v4.6)*

Saves the Clipboard contents (any type of data) in NoteTab's internal memory. This command is to be used in conjunction with **^!ClipboardRestore**. It is particularly useful when you want to use the Clipboard for temporary storage without permanently removing what the user may have initially stored in it.

^!ClipboardRestore [+] *(added in v4.6)*

Restore the contents of clipboard saved through the **^!ClipboardSave** command. Use the optional "+" parameter if you want NoteTab to keep the saved Clipboard data in its internal memory (lets you restore it again at a later stage).

^!CopyClip ClipName

Copies the content of a Clip directly into the Clipboard. Clips are not executed through this procedure, so even if the Clip is executable or contains variables, these are not processed and the text is used in its original state.

^!Paste

Inserts the Clipboard contents into the active document.

^!PasteSelect

Pastes the content of Clipboard into active document and selects the pasted text.

^!PasteBack

Unlike the "**^!Paste**" command, which pastes the Clipboard content into the current NoteTab document, "**^!PasteBack**" attempts to paste into the last control that copied text to the Clipboard (the control is not reset when text is copied through Clip commands). Thanks to this command, you can paste text processed by the Clipbook into practically any active application. Note that this feature is only available in the commercial and trial versions of NoteTab.

^!TypeBack

Similar to **PasteBack** but inserts text by typing it instead of pasting it. Slower than PasteBack but may work better with some applications.

Clip Language - Find / Replace Text

Use the following commands and functions to search or replace text in the active document. Special instructions are also available to search and process fields in delimited (tab, colon, or other) text files.

^!Find "SearchText" [Options TCIBGWHRS]

Searches for the specified text. Tab and line break characters should be replaced by their tokens (^T and ^P respectively). You can use the following optional settings:

T: Does not search for whole words only (search string can be found within longer words).

C: Searches for whole words only.

If the options **T** or **C** are not used, NoteTab will search for whole words if the search criteria does not contain spaces or other word delimiting characters. If the search criteria contains such delimiters, then NoteTab will accept matches that are not whole words.

I: Ignores character case.

B: Searches backwards.

G: Global scope; searches through all the open documents.

W: Searches through the whole document (not just from the cursor position).

H: Only searches within selection if text is highlighted (setting ignored if no text selected).

R: Specifies that the search criteria represents a regular expression.

S: Silent search. NoteTab will not display any message box.

The following example will find the next occurrence of the word NoteTab starting from the cursor position. If a match is not found in the current document, the other open documents will also be searched:

```
^!FIND "NoteTab" G
```

The search criteria must always be placed between double quotes if you don't specify search options.

^!Replace "SearchText" >> "ReplaceText" [Options TCIBGWHRSA]

or

^!Replace SearchText ReplaceText [Options TCIBGWHRSA]

Similar to the "^!Find" command but replaces matches with the replace criteria. The first format uses spaces to separate parameters and the second uses a tab character. Use one or the other but do not mix the two formats within an instruction. You can use the following optional setting, which is specific to the Replace command:

A: Use to replace all matched occurrences. When this option is not used, only the first match is replaced.

The following Clip example will search all open documents and replace the listed upper case HTML tags to their lower case equivalent:

```
^!Replace <B> <b> GA
```

```
^!Replace </B> </b> GA
```

```
^!Replace <I> <i> GA
```

```
^!Replace </I> </i> GA
```

^!SetDelimiter "Delimiter"

Defines field delimiter used by "GetField" and "GetPosInField" functions.

^\$GetField(Row;Col)\$

Returns the field text found at the specified row and data column in a delimited file. Set Row to -1 to indicate the current line.

^\$GetPosInField(Row;Col;Options;"SearchCriteria")\$

Returns the position of a search criteria in the specified field in a delimited text. No match returns 0. You can use the following optional settings:

T: Does not search for words only (search string can be found within longer words)

I: Ignore character case

W: Criteria must match whole field

S: Criteria must be at start of field

Clip Language - Document Processing

Various commands and functions to process the text in a document.

^!InsertFile FileName

Inserts the content of the specified file "FileName" at the cursor position.

^\$GetText\$

Returns the whole text contained in the active document.

^\$GetTextSize\$

Returns the text size of the current document.

^\$GetLineCount\$

Returns total number of lines in the current document. This count is affected by the word-wrap status. The last line is considered to follow the last new line ("Enter") character, even if this line contains no characters. Or think of it as the total number of rows in the editor window.

^\$GetParaCount\$

Returns total number of paragraphs in the current document. In this context, a paragraph is defined as a sequence of characters terminated by a new line character. (New line characters are what is inserted when you press the "Enter" key in a document.) NoteTab's "paragraphs" are what many other programs call "lines". The GetParaCount command is not affected by the word-wrap status. When word-wrap is turned off, GetParaCount works the same as GetLineCount.

^\$GetTextLineCount\$

Similar to **^\$GetParaCount\$**, but if the last character in the file is a new line it will not count the empty space after it as a line, the way that GetParaCount does. (A new line is what you get when you press "Enter" in a document. NoteTab's ^P token represents this.) That is, new line characters are considered by GetTextLineCount to merely terminate the current line. This gives a result that agrees with many other utilities, especially those used on UNIX systems.

^\$GetLineSize\$

or

^\$GetLineSize(Index)\$

or

^\$GetLineSize(+/-Index)\$

Returns the size of the text contained in the specified line. The "Index" parameter is optional. When it is not used, this function refers to the line where the cursor is placed; otherwise it refers to the line position from the start of the document. By preceding the number with a + or - sign, the Index value refers to the relative position with respect to the current line (e.g. **^\$GetLineSize(+1)\$** refers to the size of the next line).

^\$GetLine\$

or

^\$GetLine(Index)\$

or

^\$GetLine(+/-Index)\$

Returns the text contained in the specified line. The "Index" parameter is optional. When it is not used, this function refers to the line where the cursor is placed; otherwise it refers to the line position from the start of the document. By preceding the number with a + or - sign, the Index value refers to the relative line position with respect to the current line (e.g. **^\$GetLine(-1)\$** refers to the previous line).

^\$GetParagraph\$ *(added in v4.6)*

or

^\$GetParagraph(Index)\$ *(added in v4.6)*

or

^\$GetParagraph(+/-Index)\$ *(added in v4.6)*

Returns the text contained in the specified paragraph (=single word-wrapped line). The "Index" parameter is optional. When it is not used, this function refers to the paragraph where the cursor is placed, otherwise it refers to the paragraph position from the start of the document. By preceding the number with a + or - sign, the Index value refers to the relative position with respect to the current paragraph (e.g.

^\$GetParagraph(-1)\$ refers to the previous paragraph).

^!DeleteLine

Deletes the current line.

^!SetWordWrap ON/OFF *(added in v4.51)*

Changes word wrap mode in active document.

^!SetColumnWrap nn

Defines the column position for word wrap. nn is the value in number of characters.

^!UpdateColumnWrap

NTS only. Reformats the text to fit within the defined column wrap. This command has no effect in NoteTab Pro.

^!AddHeading "HeadingName"

Adds a new heading to the active Outline document.

Clip Language - Document Management

Use the following commands and functions to manage the documents in NoteTab.

^!Open "FileName" [/R] [/J=nnn]

or

^!Open "File1";"File2";"File3" *(added in v4.6)*

Opens or selects the specified document "FileName", or opens multiple files (names separated by semicolon). Use fully qualified file names (with path name) to ensure the command finds the correct file. You can use wild cards with this command. The optional switch **/R** can be used to open the file(s) as Read-Only, and the **/J=** switch can be used to place the cursor at a specific line number when it is opened. If **/J=-1**, the cursor will be placed at the beginning of the last line.

^!OpenReadOnly FileName *(added in v4.6)*

Opens or selects the specified document "FileName". You can use wild cards with this command. Newly opened files are set as Read-Only.

^!OpenAscii FileName *(added in v4.6)*

or

^!OpenAscii FileName1;FileName2;FileName3 *(added in v4.6)*

Opens and translates the text from ASCII format to ANSI, or selects the specified document "FileName". You can use wild cards with this command.

^!OpenEbcDic FileName [/W=nnn] *(added in v4.6)*

or

^!OpenEbcDic FileName1;FileName2;FileName3 *(added in v4.6)*

Opens and translates the text from EBCDIC format to ANSI, or selects the specified document "FileName". You can use wild cards with this command. The optional parameter **/W=nnn** can be used to define a line length for fixed-width EBCDIC files. "nnn" represents the width in characters; a value of 0 indicates variable line lengths delimited with carriage returns. Example: **^!OpenEbcDic /W=132**

^!LoadEbcDicCharTable FileName *(added in v4.6)*

Loads the specified file containing the EBCDIC character tables for conversion to and from Windows ANSI. The NoteTab program path is added if you do not specify a fully qualified path name. The extension .ebc is added if you do not specify a filename extension. You can reset the default character table by specifying the Default.ebc file.

^!Save [AS|ALL]

or

^!Save AS FileName

Saves the active document. If the optional parameter **"AS"** is used, the "Save As" dialog box is displayed. If the optional parameter **"ALL"** is used, all the modified documents are saved. You cannot use both parameters together. If you specify a file name after the **"Save AS"** element, the Save As dialog box is not displayed and the specified file name is used.

The following Clip example will save the changes made to the active document:

^!Save

The next example will save all modified documents:

^!Save ALL

The following Clip example will save the active document as a new file called Temp.txt:

^!Save AS c:\My Documents\Temp.txt

^!SaveAscii FileName *(added in v4.6)*

Similar to the **^!Save** command (and accepts the same parameters) but saves the document using the

DOS extended ASCII character set instead of Windows ANSI.

^!SaveEbcdic FileName [/W=nnn] *(added in v4.6)*

Similar to the **^!Save** command (and accepts the same parameters) but saves the document in EBCDIC format. The optional parameter "/W=nnn" can be used to define a line length for fixed-width EBCDIC files. "nnn" represents the width in characters; a value of 0 indicates variable line lengths delimited with carriage returns. Example: **^!SaveEbcdic /W=132**

^!RenameDoc NewName *(added in v4.6)*

Renames the active document. The disk file is also renamed if it exists.

^!Close [ALL,SAVE|DISCARD]

or

^!Close "DocumentName" [SAVE|DISCARD]

Closes the specified document(s). If **SAVE** is specified, a modified document will be saved without prompting (unless it has never been saved before). If **DISCARD** is specified, modified files are closed without prompting or saving. The **ALL** parameter can be used together with either **SAVE** or **DISCARD**. Options should be separated by a comma or a semicolon, and no spaces; For example: **^!Close ALL,SAVE**

^!DestroyDoc [DocumentName]

Closes the current document or the specified document without prompting and deletes the corresponding file from disk if it exists (the file is actually sent to the recycle bin).

^!Document [FIRST|LAST|NEXT|PREV]

Selects the specified document tab.

^!FocusDoc

Moves the input focus to the active document.

^!Print [ALL]

Prints the active document. If the optional parameter "ALL" is used, all the open documents are printed.

^!DirectPrint "Title" Data *(added in v4.8)*

Prints text directly to the printer, bypassing NoteTab's print procedure. Uses default printer settings. Title is the text shown in the Print manager and Data is the text to send directly to the printer. Data can contain printer control codes.

^!OpenFavorites CategoryName *(added in v4.52)*

Opens all the documents listed in the specified Favorites list. CategoryName should be specified without a path name or extension.

^!SaveFavorites CategoryName *(added in v4.52)*

Adds all the currently open document names to the specified Favorites list. The Favorites category is created if it doesn't exist. CategoryName should be specified without a path name or extension.

^!SetNameFormat Mask *(added in v4.6)*

The name format is used when new documents are created. The code "%d" indicates where the incremental numbering is inserted in the name. In the following example, the "0.2" between the % and d tells NoteTab to format numbers smaller than 2 digits with leading zeroes: NoName%0.2d.txt

^\$GetNameFormat\$ *(added in v4.6)*

This function returns the current new document name format.

^\$GetDocCount\$

Returns the number of open documents in editor.

^\$GetDocName\$

^\$GetDocName(Index)\$ *(added in v4.52)*

Returns the active document name, if no index is specified, or the document name at specified index position. An invalid index number returns an empty string. The first document is at index 1.

^\$GetDocIndex\$

^\$GetDocIndex(Name)\$

Returns the index position of the active document, or the document referred by the Name parameter.

^!SetDocIndex IndexValue

Activates the document at the specified position in the tab bar. First document is 1.

^\$GetTopicCount\$

Returns number of topics in the current outline document.

^\$GetTopicName\$

^\$GetTopicName(Index)\$ *(added in v4.52)*

Returns the current outline heading name, if no index is specified, or the heading name at specified index position. An invalid index number returns an empty string. The first heading is index 1.

^\$GetTopicIndex\$

^\$GetTopicIndex(Name)\$ *(added in v4.51)*

Returns the index position of current outline topic. Use the optional Name parameter to find the index position of another topic.

^!SetTopicIndex IndexValue

Activates the topic at the specified position in the current outline document. First topic is 1.

^!ShowHeadings [True|False] *(added in v4.52)*

Shows or hides Outline heading names in the editor window.

^!IsShowHeadings\$ *(added in v4.52)*

Returns 1 if headings are displayed in the outline document window and 0 if they are not.

^\$GetPasteBoardIndex\$

Returns the index position of the PasteBoard document. Returns 0 if there is no PasteBoard document.

^\$GetSecondWindowIndex\$

Returns the index position of document in second window. Returns 0 if there is no second window.

Clip Language - Disk Files

Use the following commands and functions to manage disk files from NoteTab. Use fully qualified file names (with path name) to ensure the commands and functions find the correct file. See also [Disk Directories](#) for related commands.

^!CopyFile "FileSpecs" "Destination"

Copies the specified file(s) (wildcards can be used) to another directory. Use double quotes around the file names if they contain spaces. If no wildcards are used and the destination file exists, it is sent to the recycle bin before being replaced by the new file.

^!MoveFile "FileSpecs" "Destination"

Moves the specified file(s) (wildcards can be used) to another directory. Use double quotes around the file names if they contain spaces.

^!RenameFile "OldName" "NewName"

Renames the specified file (no wildcards allowed). Use double quotes around the file names if they contain spaces.

^!DeleteFile FileSpecs

Deletes the specified file(s) (wildcards can be used) from the disk.

^!RecycleFile FileSpecs

Moves the specified file(s) (wildcards can be used) to the recycle bin.

^!Refresh

Use this command after file management actions to update the list of available Clipbook libraries and the contents of the Quick List tool. It also checks if any of the open documents were changed on disk.

^!TextToFile "FileName" AnyText

Creates a disk file containing the specified text. If a file with such a name already exists on disk, it will be replaced by the new file.

^!AppendToFile "FileName" AnyText

Appends the specified text to the end of a disk file. A new file is created if FileName does not exist.

^!TextToUnicodeFile "FileName" AnyText *(added in v4.82)*

Saves the specified text to FileName in the Unicode format using the current Windows code page.

^!AppendTextToUnicodeFile "FileName" AnyText *(added in v4.82)*

Appends the specified text to the end of a disk file in Unicode format using the current Windows code page. A new file is created if FileName does not exist.

^!Export "FileName" [ASCII|ANSI][MAC|UNIX][SELECTION|ALL][HARDBREAKS]

or

^!Export "FileName" [EBCDIC][SELECTION|ALL][LRECL=nnn] *(added in v4.6)*

Exports current document under specified file name. Optional settings are used to define output format. Use a comma or colon to separate multiple settings and use double quotes around the file name if it contain spaces. When creating EBCDIC files, you can optionally define a record length (for fixed-width lines) by using the LRECL parameter followed by an equal sign and the width value.

^\$GetFileDate(FileName)\$

Returns the file date as yyyyymmdd. Returns 0 if file does not exist.

^\$GetExt(FileName)\$

Returns the extension of FileName (with extension dot).

^\$GetName(FileName)\$

Returns the name of FileName (without path or file extension). Version 4.8: this function works both on Windows file names and URLs.

^\$GetFileName(FileName)\$

Returns the name and extension of FileName (without path). Version 4.8: this function works both on Windows file names and URLs.

^\$GetPath(FileName)\$

Returns the path segment of FileName (always ends with backslash). Version 4.8: this function works both on Windows file names and URLs.

^\$GetFileSize(FileName)\$

Returns the size of FileName.

^\$GetFileText(FileName)\$

Returns the content of the specified file. The value is an empty string if the file does not exist.

^\$GetUnicodeFileText("FileName")\$ *(added in v4.82)*

Reads Unicode text from FileName and returns it as normal text based on the current Windows code page.

^\$GetFileTime(FileName)\$

Returns the file time as hhnns. The value is 0 if file does not exist.

^\$GetImgHeight(FileName)\$

Returns the height in pixels of the specified image file (.gif, .png, .jpg, .bmp). The value is 0 if file does not exist or is not an image.

^\$GetImgWidth(FileName)\$

Returns the width in pixels of the specified image file (.gif, .png, .jpg, .bmp). The value is 0 if file does not exist or is not an image.

^!SetFileAttr "FileSpecs" Attr *(added in v4.8)*

Changes the file attributes. FileSpecs is the file name, which may have wild cards. Attr can represent multiple values: A = Archive, H = Hidden, R = Read-only, S = System, * = All attributes. Use a minus sign in front of the letter to remove an attribute. The following example sets the Archive attribute and removes the Read-only attribute, if set:

```
^!SetFileAttr "c:\Scandisk.log" A-R
```

^!IfFileAttr "FileName" Attr GoToLabelTrue [ELSE GoToLabelFalse] *(added in v4.8)*

Compares FileName attributes with Attr. If Attr is equal or a subset of FileName's attributes, then GoToLabelTrue is activated. Attr can represent multiple values: A = Archive, D = Directory, H = Hidden, R = Read-only, S = System, V = VolumeID, * = All attributes. Example:

```
^!IfFileAttr "c:\Scandisk.log" AH LabelTrue ELSE LabelFalse
```

If Scandisk.log has at least the Archive and Hidden attributes set, then the instruction jumps to LabelTrue.

^\$GetFileAttr(FileName)\$ *(added in v4.8)*

Returns the file attributes as a string of characters. A file can have zero, one, or several attributes. The string may contain the following characters: A = Archive, D = Directory, H = Hidden, R = Read-only, S = System, V = VolumeID, * = All attributes. Example:

```
^$GetFileAttr(c:\Scandisk.log)$ -> AH
```

^\$GetRelativeName(FileName;BaseDir)\$ *(added in v4.8)*

Returns the relative name of FileName against BaseDir. FileName must include a fully qualified path. Enclose the parameters in quotes if they contain a semicolon in the name. If BaseDir is not defined, the active document path is used as the reference directory. Example:

```
^$GetRelativeName(c:\temp\test\MyFile.txt;c:\temp\)$ -> test\MyFile.txt
```

^\$GetExpandedName(FileName;BaseDir)\$ *(added in v4.8)*

Returns the fully qualified name of FileName against BaseDir. If BaseDir is not defined, the active document path is used as the reference directory. Example:

```
^$GetExpandedName(..MyFile.txt;c:\temp\test\)$ -> c:\temp\MyFile.txt
```

^\$FileToUrl("FileName")\$ *(added in v4.8)*

Converts a Windows file name to a URL file name. Example:

```
^$FileToUrl("c:\program files\index.htm")$ -> file:///c:/program%20files/index.htm
```

^\$UriToFile("URL")\$ *(added in v4.8)*

Converts a URL file name to a Windows file name. Example:

```
^$UriToFile("file:///c:/program%20files/index.htm")$ -> c:\program files\index.htm
```

^\$GetFileFirst(FilePath;FileSpecs;[SortingOrder])\$

^\$GetFileFirst(FilePath;FileSpecs;[Attr];[SortingOrder])\$ *(added in v4.8)*

^\$GetFileFirst([+]FilePath;FileSpecs;[Attr];[SortingOrder])\$ *(added in v4.83)*

Returns the first file name in the list generated by the function. FilePath indicates the directory you want to search and FileSpecs the specs to match the files. FileSpecs can contain the standard wildcards * and ?, and you can specify several criteria separated by a semicolon (;). You must use double-quotes around the FileSpecs parameter if you specify several criteria. The third parameter, SortingOrder, is optional and can be used to specify a sorting order that is different from the default which is file names sorted in alphabetical order. You can use the following values for SortingOrder:

UNSORTED: No sorting is used.

NAME: Files are sorted by name.

TYPE: Files are sorted by type (based on the extension).

DATE: Files are sorted by date.

SIZE: Files are sorted by size.

Use the prefix "REV" to reverse the sorting order (e.g. REVSIZE starts with the largest file).

Version 4.8: added optional Attr parameter. Attr can represent multiple values: A = Archive, H = Hidden, R = Read-only, S = System, * = All attributes. If the Attr parameter is not used, then file attributes are ignored.

Version 4.83: When the Attr field is used, only files that use the specified attributes are selected. To select files that have no attribute settings, use "-" as a value for Attr. You can also span through subdirectories by adding a + sign in front of the FilePath criteria.

^\$GetFileNext\$

Returns the next file name in the list. If there are no more files, then the result is an empty string.

^!CloseFileFind

Optional command to close the list of files created with the GetFileFirst function. It is recommended to use this command when you have finished using the list of files as it releases system resources.

^\$GetFiles(FilePath;FileSpecs;[Attr];[SortingOrder])\$ *(added in v4.8)*

^\$GetFiles([+]FilePath;FileSpecs;[Attr];[SortingOrder])\$ *(added in v4.83)*

The parameters are identical to the ^\$GetFileFirst\$ function. Returns all matching files as a single string using a blank space separator, or the one defined by ^!SetListDelimiter. Note that ^\$GetFileNext\$ and ^!CloseFileFind do not work with this function. This new function is useful to assign the result to an array variable. Example:

```
^!SetArray %Files%=^$GetFiles("c:\Program Files\NoteTab Pro\Documents";*.txt;AR;DATE)$
```

Version 4.83: When the Attr field is used, only files that use the specified attributes are selected. To select files that have no attribute settings, use "-" as a value for Attr. You can also span through subdirectories by adding a + sign in front of the FilePath criteria. The following example will select all files under Windows and its subdirectories, that have the System attribute set:

```
^$GetFiles("+C:\Windows";*.*;S;Name)$
```

^\$GetShort(FileName)\$

Returns the DOS short name of FileName.

^\$GetTempFile\$

Returns a Windows temporary filename which can then be used to store data. This file is automatically deleted when you exit Windows.

Clip Language - Disk Directories

Use the following commands and functions to manage disk directories from NoteTab. See also [Disk Files](#) for related commands.

^!Mkdir DirName

Create a new directory with the name specified by DirName.

^!Rmdir DirName

Deletes the specified directory from the disk. This function will fail if the directory is not empty.

^!Chdir DirName

This command only affects shell commands and is used to define the default directory used by the launched application.

^\$GetPath(FileName)\$

Returns the path segment of FileName (always ends with backslash). Version 4.8: this function works both on Windows file names and URLs.

^\$GetAppPath\$

Returns the NoteTab application path (always ends with backslash).

^\$GetLibraryPath\$

Returns the path for NoteTab Clipbook library files (always ends with backslash).

^\$GetDocumentPath\$

Returns the path for NoteTab document files (always ends with backslash).

^\$GetSamplesPath\$

Returns the path for NoteTab sample files (always ends with backslash).

^\$GetScriptPath\$

Returns the path for NoteTab script files (e.g. AWK, Perl, etc.; always ends with backslash).

^\$GetSoundPath\$

Returns the path for NoteTab sound files (always ends with backslash).

^\$GetTemplatePath\$

Returns the path for NoteTab template files (always ends with backslash).

^\$GetFavoritePath\$

Returns the path for NoteTab "favorites" files (always ends with backslash).

^\$GetTmpPath\$

Returns the Windows temporary path (always ends with backslash).

^\$GetWinPath\$

Returns the Windows path (always ends with backslash).

^\$GetSysPath\$

Returns the Windows System path (always ends with backslash).

^\$GetSysProgPath\$ *(added in v4.82)*

Returns the path containing system programs like MS-Notepad and FTP.exe. By default, the path is "c:\WINDOWS\" under Windows 95 and 98, and "c:\WINNT\System32\" under Windows NT.

^\$GetShort(FileName)\$

Returns the DOS short name of FileName.

Clip Language - Executing Programs

Use the following commands and functions to execute external programs and minimize/restore the NoteTab window. See also [Disk Directories](#) for related commands.

^!CmdShow [Normal|Minimize|Maximize]

Use this command to display the next program to be launched through the Clipboard minimized/maximized. The following values can be used:

Normal: Program is launched using the default window style.

Minimize: Program is launched minimized.

Maximize: Program is launched maximized.

^!ChDir DirName

This command only affects shell commands and is used to define the default directory used by the launched application.

^!CommandLine

Executes the specified program command unless it represents a Clip command. See also the new **^!Shell** command below.

^!Shell CommandLine *(added in v4.82)*

Launches the specified file, program, or URL through the Windows shell. Proposed as a replacement for the default "**^!**" launch method. Example: instead of using "**^!WordPad c:\autoexec.bat**", use "**^!Shell WordPad c:\autoexec.bat**"

^!ShellWait CommandLine *(added in v4.82)*

Similar to the **^!Shell** command, but waits until the launched program is terminated before moving to the next Clip instruction. Unlike the **^!Wait** command, you can continue editing files in NoteTab while the application is running. Example:

^!ShellWait WordPad c:\autoexec.bat

^!Run CommandLine

This command is provided for backward compatibility. It can be used to launch other programs. The **^!Wait** command does not work with programs launched this way.

^!Dos DosCommand

Executes the specified Dos-shell command or program. The Dos window is closed when the command or program has finished executing. Example: **^!Dos dir *.***

^!Wait

This instruction will interrupt Clip execution until the previously launched application has closed. You cannot use the editor when it is in Wait mode (the editor's window caption indicates it is waiting for the application to close). Note that the **^!Wait** command may not work with all applications (during testing, it did not work with the MS-Write program).

^!Url www.notetab.com

Opens the specified URL in the default browser.

^!Email Address;Subject[;+]

or

^!Email Address;Subject;MessageText *(added in v4.6)*

Creates a new e-mail message. The e-mail address and subject should be separated by a semicolon (;). The third parameter with the "+" is optional. When used, it tells the Clipboard that the content of the current

document should be used for the body of the message. Since version 4.6, you can include a whole message text after the Subject parameter.

The following Clip example will create a new e-mail message with the address "sales@notetab.com" and the subject "Order for NoteTab Pro":

```
^!Email sales@notetab.com;Order for NoteTab Pro
```

The next example will create a new e-mail message with no preset address, the subject "Please check!", and the content of the current document:

```
^!Email ;Please check!;+
```

^!FocusApp AppTitle *(added in v4.52)*

Activates the application window with a title caption that matches the specified text. The error flag is set if no matching window is found. For example, if the Windows calculator utility is open, the following command will bring it to the front:

```
^!FocusApp Calculator
```

Most applications usually show other information in the title bar. For these cases, you can use an asterisk "*" either at the beginning or the end of the AppTitle text to specify that the text should match the end or the beginning of the title respectively. For example, MS Wordpad displays the document name in front of the program name, so the following instruction will work if Wordpad is open:

```
^!FocusApp *WordPad
```

^!SetStdErrorName FileName

Defines the default filename for the StdError output generated by console applications. If you don't specify a path, the NoteTab application directory is used.

^\$GetInputOutput("Command")\$

Launches the console program specified by "Command" and sends the current document text to the StdInput data stream. The function returns the console's StdOutput data stream. The stderr stream is saved in a file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdErrorName command. Prior to version 4.8, this command was only available in the commercial and trial versions of NoteTab.

^\$GetOutput("Command")\$

Returns the console's output. "Command" represents the command to execute. The stderr stream is saved in a file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdErrorName command. Prior to version 4.8, this command was only available in the commercial and trial versions of NoteTab.

Example: `^$GetOutput(command.com /c dir c:*.*)$`

^\$GetDosOutput("Command")\$

Returns output from DOS command. "Command" represents command to execute. The stderr stream is saved in a file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdErrorName command. Prior to version 4.8, this command was only available in the commercial and trial versions of NoteTab.

Example: `^$GetDosOutput(dir c:*.*)$`

^!MinimizeApp

Minimizes the NoteTab application Window. This can often make clips run faster. But note that some commands, (such as `^!Keyboard`), do not work properly when the editor window is minimized.

^!ActivateApp

Activates the NoteTab application Window. If NoteTab is minimized or behind other windows, the window is restored and brought to the front.

Clip Language - Using Perl, Gawk, and Other Scripts

Use the following commands and functions to execute Perl, Gawk, or other scripts. See also [Executing Programs](#) for related commands.

If you don't have a Win32 Perl or Gawk interpreter installed on your computer, there are a number of good ones you can download from the Internet. The following Freeware 32-bit interpreters have been tested with NoteTab and work very well:

For Perl:

<http://www.ActiveState.com/pw32/>

For Gawk:

<ftp://oak.oakland.edu/pub/simtelnet/gnu/gnuish/gawk304x.zip>

All you need to do is install them wherever you want on your hard disk. Then, the first time you try to run a script, NoteTab will prompt you for the file name of the interpreter. NoteTab will only prompt you again if you delete or move the interpreter to another location, or delete the NoteTab INI file.

^!RunPerl ScriptName

Applies specified Perl script to selection in current document or whole document. ScriptName can either represent the name of a Clip or a script file. The stderr stream is saved in file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdError Name command.

^!RunGawk ScriptName

Applies specified GAWK script to selection in current document or whole document. ScriptName can either represent the name of a Clip or a script file. The stderr stream is saved in file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdError Name command.

^!RunScript "Command" "ScripName"

Similar to ^!RunGawk except you must specify the command to execute the console application. The stderr stream is saved in file called "StdError.err" in NoteTab's application directory unless another file is specified with the SetStdErrorName command. Use the code "^1" to indicate where the Script file name should be inserted.

^!SetStdErrorName FileName

Defines the default filename for the StdError output generated by console applications. If you don't specify a path, the NoteTab application directory is used.

^\$GetStdErrorName\$

Returns the current filename for the StdError output generated by console applications.

^\$GetPerlExe\$

Returns full path and name of Perl application.

^\$GetGawkExe\$

Returns full path and name of GAWK application.

Clip Language - Using Other Clips

Use the following commands and functions to execute other Clips, retrieve their content, change libraries, etc.

^!Clip "ClipName" [Text]

Executes the specified Clip in the current library. Do not attempt to call clip items recursively or NoteTab will *freeze* in a never-ending loop (if this happens, you can press the Ctrl+Alt keys together to stop the execution). You can insert text in a Clip item by using an optional second parameter – this will make the Clipbook behave in the same way as if that text had been selected in a document. You must place double quotes around the Clip name if you want to specify text criteria or if the Clip name contains one or more blank spaces.

The following example will activate the Clip item called "Explorer":

```
^!Clip Explorer
```

If the Clip called Explorer contains the ^& code, the next example will substitute it with the text "C:\Windows":

```
^!Clip "Explorer" C:\Windows
```

^!FarClip LibraryName:ClipName

or

^!FarClip "LibraryName:ClipName" [Text]

Similar to the ^!Clip command but invokes a Clip from a library on disk instead of the active library. Clips invoked through this method are stored in an internal cache memory to improve performance (the cache will reload the Clip to memory if it detects that the Library file has been modified). LibraryName name must be stored in the Libraries folder if you do not specify a path, but does not need to have the .clb extension (you can use a different extension if you do not want the library to be listed in the Clipbook tool or the button bar). This command is particularly useful within specialized Template files. You can also use it to invoke frequently used procedures. Example: ^!FarClip HTML:Center

^\$GetClipText(ClipName)\$

Returns the content of a Clip as text. The specified Clip is not executed when invoked. Note that Clips using this function cannot be run from the Clipbar or the ^!FarClip command.

^\$GetClipVersion\$

Returns the version of the Clipbook parser. Useful to check for compatibility with Clip instructions. Since version 4.801, this function returns the actual NoteTab release version as an integer. For example, in version 4.801, ^\$GetClipVersion\$ returns 4801.

^\$GetClipName\$ *(added in v4.6)*

Returns the name of the currently interpreted Clip. This function returns an empty string within Clips run by the timer event and executable templates.

^\$GetLibraryName\$

Returns the name of the current Clipbook library.

^!LoadLibrary LibraryName

Loads the specified library into Clipbook.

^!ReloadLibrary *(added in v4.52)*

Reloads the current Clipbook library. Useful if another application has modified it.

^!DestroyLibrary [LibraryName] *(added in v4.8)*

Deletes the specified Clipbook library (path name and extension are not needed). If the optional

LibraryName argument is not used, then the library currently running the Clip is closed and destroyed.

^!ShowClipbook [TRUE|FALSE]

Shows or hides the Clipbook window.

^!TimerPlay ClipName

Assigns the Clip instructions to the timer memory. The instructions remain available even when you change libraries but are cleared when you close the Clipbook.

^!TimerStart nnnn

Defines an event timer. nnnn represents the interval in milliseconds. For example, "**^!TimerStart 5000**" will execute the assigned Clip (set with "**^!TimerPlay**") every 5 seconds. Note that the Clip is not executed if the Clipbook is already in the process of executing another Clip.

^!TimerStop

Stops the timer and clears the assigned Clip instructions.

Clip Language - Clipbars

Use the following commands and functions to control the new Clipbar feature.

^!OpenClipbar Name *(added in v4.8)*

Opens the specified Clipbar file.

^!CloseClipbar *(added in v4.8)*

Closes the current Clipbar.

^!ReloadClipbar *(added in v4.8)*

Reloads the current Clipbar.

^\$GetClipbarName\$ *(added in v4.8)*

Returns the name of the current Clipbar file.

Clip Language - Toolbar Commands

Execute any NoteTab toolbar command from a Clip. See also [Keyboard Macros](#) for related commands.

^!Toolbar ToolTipText

You can execute any NoteTab command that is available in the toolbar (it doesn't matter if the button is displayed in the toolbar or not). You specify the command you want to execute by using the keyword "Toolbar" followed by a space and its tooltip text. The button tooltip text can be seen by placing the mouse cursor over the button. The available commands are also listed in the Clip Assistant. You can check if the command was successfully executed with the **^!IfError** command. An error condition is only triggered if the command failed because the button was disabled.

The following example will activate the "Open Document" command:

```
^!TOOLBAR Open Document
```

You can of course create a sequence of instructions within a Clip. The next example will create a new document, paste the content of the Clipboard into it, then activate the Save As dialog box:

```
^!TOOLBAR New Document
```

```
^!TOOLBAR Paste
```

```
^!TOOLBAR Save As
```

Clip Language - Keyboard Macros

Use keyboard macros to simulate keys being pressed (either in NoteTab or other applications). See also [Toolbar Commands](#) for other types of macro commands.

^!Keyboard Keys

Plays the specified key sequence within NoteTab. Use blank characters as delimiters between the keys to play. To specify plain text, wrap it with the # character. Characters can also be represented as hexadecimal values by using a \$ symbol in front of the number. For example, \$20 represents the spacebar character and \$09 the tab key.

Since version 4.6, the keyboard command supports delay settings between key instruction. The value represents milliseconds and should be preceded with the "&" symbol. Example:

```
^!Keyboard Alt+V O &1000 Ctrl+Tab &400 Ctrl+Tab
```

Valid control keys are:

CTRL, SHIFT, ALT (Add UP/DOWN suffix to hold button up or down respectively)

Function keys:

F1, F2, ..., F12

Other keys:

BACKSPACE, TAB, ENTER, ESC, SPACEBAR,
PAGEUP, PAGEDOWN, END, HOME, LEFT, UP, RIGHT, DOWN,
INSERT, DELETE, HELP

Examples:

```
^!Keyboard CTRL+O #Hello World#
```

and

```
^!Keyboard SHIFTDOWN RIGHT RIGHT RIGHT SHIFTOP
```

and

```
^!Keyboard ALT+T S M
```

and

```
^!Keyboard SHIFT+F2
```

and

```
^!Keyboard $4E $6F $74 $65 $54 $61 $62
```

^!KeyboardDelay nnn *(added in v4.6)*

Controls the speed of typing with the ^!Keyboard command. nnn specifies the approximate delay in milliseconds between each keystroke. Set nnn to 0 for maximum speed. The scope of this command is limited to the current Clip execution.

^!Playback Keys

Similar to the TypeBack command but plays back the specified key sequences (same used by ^!Keyboard command). Works with most programs.

Clip Language - Windows Registry and Shortcuts

Use the following commands and functions to access the Windows registry and create Windows shortcuts.

^\$GetRegValue("[BaseRoot]Key[:Name]")\$ *(added in v4.8)*

Returns the value stored under the specified key in the Windows Registry. If BaseRoot is not defined, then HKEY_CURRENT_USER is used. If no value is specified for Name, the "(Default)" name is used. Returns an empty string if the key does not exist. Example:

```
^$GetRegValue(HKEY_CURRENT_USER\Software\Beta:UserName)$ -> Raymond
```

```
^$GetRegValue(Control Panel\International:sDate)$ -> /
```

```
^$GetRegValue(HKEY_CLASSES_ROOT\386)$ -> vxdfile
```

Version 4.83: if the Key value contains a colon, the Clip interpreter will incorrectly assume it marks the limit between the Key and Name elements. To remedy this situation, use a double colon after the key name. Example:

```
^$GetRegValue(HKEY_CURRENT_USER\Software\c:Temp::Folder)$
```

^!SaveRegValue [BaseRoot]Key[:Name]=Value *(added in v4.8)*

Saves the specified value under the key entry in the Windows Registry. If BaseRoot is not defined, then HKEY_CURRENT_USER is used. If no value is specified for Name, the "(Default)" name is used. The maximum amount of text that can be saved this way is 1024 characters. Use with care! Example based on ^\$GetRegValue\$ example:

```
^!SaveRegValue HKEY_CURRENT_USER\Software\Beta:UserName=Raymond
```

```
^!SaveRegValue Control Panel\International:sDate=/
```

```
^!SaveRegValue HKEY_CLASSES_ROOT\386=vxdfile
```

Version 4.83: if the Key value contains a colon, the Clip interpreter will incorrectly assume it marks the limit between the Key and Name elements. To remedy this situation, use a double colon after the key name. Example:

```
^!SaveRegValue HKEY_CURRENT_USER\Software\c:Beta::UserName=Raymond
```

^!ClearRegValue [BaseRoot]Key[:Name] *(added in v4.8)*

Deletes the specified name or key in the Windows Registry. If BaseRoot is not defined, then HKEY_CURRENT_USER is used. If a Name is specified, then only that name will be deleted from the Registry. If no Name is specified (no colon), then the key and all Names under it will be deleted. Use with care!

Version 4.83: if the Key value contains a colon, the Clip interpreter will incorrectly assume it marks the limit between the Key and Name elements. To remedy this situation, use a double colon after the key name. Example:

```
^!ClearRegValue HKEY_CURRENT_USER\Software\c:Beta::UserName=Raymond
```

BaseRoot can be any of the following values:

```
HKEY_CLASSES_ROOT
```

```
HKEY_CURRENT_USER
```

```
HKEY_LOCAL_MACHINE
```

```
HKEY_USERS
```

^\$GetSpecialPath(TypeFolder)\$ *(added in v4.8)*

Returns the path name of the specified Windows folder. TypeFolder can have one of the following values:

```
AppFolder
```

```
Desktop
```

```
DesktopDir
```

StartMenu
Programs
Startup
SendTo
Favorites
Recent
CommonDesktopDir
CommonStartMenu
CommonPrograms
CommonStartup
CommonFavorites

^!MakeShortcut ShortcutLocation "Target" ["Arguments"] ["Description"] [CmdShow] *(added in v4.8)*

Creates Windows shortcuts. Syntax:

- **ShortcutLocation** is the place where the shortcut is created. It accepts the same values as the TypeFolder argument used in the `^$GetSpecialPath(TypeFolder)$` function. - "Target" is the program to execute, or the document to launch.
- **"Arguments"** are eventual command-line switches you might want to use with the target program.
- **"Description"** is an optional parameter you can use to identify the shortcut. If it is left blank, the shortcut name will be the same as the target program name.
- **CmdShow** defines how the program is opened. It can have one of the following values:
 - Normal (default if no value specified)
 - Maximized
 - Minimized

Example:

`^!MakeShortcut SendTo "^$GetAppFileName$" "" "Open in NoteTab" MAXIMIZED`

Clip Language - Miscellaneous Commands / Functions

Miscellaneous but useful Clip commands and functions.

^!Delay nnn

Adds a delay in 10th seconds before next instruction is executed. nnn is the time value.

^!Sound FileName (or Windows sound)

Plays Wave file or Windows sound (SystemAsterisk, SystemExclamation, SystemHand, SystemQuestion, SystemDefault).

^!Goto Label

Continues Clip execution from the specified label defined in the Clip.

^!SetErrorLabel LabelName *(added in v4.52)*

Defines a label to which execution should jump whenever the error flag is set. The scope of this command is limited to the Clip where it was invoked. Note that when an error label is defined, the **^!IfError** command is ignored.

^!SetScreenUpdate [On|Off] *(added in v4.52)*

Enables or disables updating of the NoteTab application window. Disabling updating when a repetitive sequence of Clip commands affects the text layout can significantly improve the processing speed. The scope of this command is limited to the Clip (and eventual sub-Clips) where it was invoked. When a Clip ends, screen updating is automatically re-enabled. Since version 4.6, the following commands will now force a screen refresh while screen updating is disabled: **^!Prompt**, **^!Continue**, **^!Skip**, **^!Info**, **^!StatusShow**, and Clip Wizards based on **^?{...}** type fields.

^!DirectPrint "Title" Data *(added in v4.8)*

Prints text directly to the printer, bypassing NoteTab's print procedure. Uses default printer settings. Title is the text shown in the Print manager and Data is the text to send directly to the printer. Data can contain printer control codes.

^!SetPasteboardDivider DividerText

Use this command to change the Paste Board divider. Accepts tokens to specify new lines (^p) and tabs (^t).

^\$GetPasteboardDivider\$

Returns the value of the Paste Board divider.

^!ShowClipbook [TRUE|FALSE]

Shows or hides the Clipbook window.

^!ReloadLibrary *(added in v4.52)*

Reloads the current Clipbook library. Useful if another application has modified it.

^\$GetLibraryName\$

Returns the name of the library running the Clip.

^\$GetEditorType\$

Returns NTS if the editor is NoteTab Std or Light, and NTP if it is NoteTab Pro.

^\$GetProgName\$ (added in v4.6)

Returns the full name of the NoteTab variant.

^\$GetProgVersion\$ (added in v4.6)

Returns the current version number.

^\$GetAppFileName\$ (added in v4.8)

Returns the file and path name of the NoteTab program file. Example, with default folder for NoteTab Pro:

^\$GetAppFileName\$ -> C:\Program Files\NoteTab Pro\NotePro.exe

^\$GetAppTitle\$ (added in v4.52)

or

^\$GetAppTitle(TitleText)\$ (added in v4.52)

Returns the full title of the active application window when no title is specified, or the window matching the specified text.

^!ProgIniSave [IniName] (added in v4.6)

Saves the current program settings into the NoteTab ini file or registry. Optionally, you can specify an alternative INI file (it is automatically created if it does not exist).

^!ProgIniLoad [IniName] (added in v4.6)

Loads options from the NoteTab ini file or registry. Optionally, you can specify an alternative INI file (it must exist or else the command sets the error condition). Note that the [AutoLoad] section is ignored by this command.

^!Help (added in v4.6)

or

^!Help "FileName" (added in v4.6)

or

^!Help "[FileName]" Keyword (added in v4.6)

Displays the indicated Help file or the NoteTab Help file if no parameter is specified. Displays the topic in the keyword table that matches the specified keyword, if there is an exact match. If there is more than one match, displays the Index tab. The following example will display the topics discussing Perl in the NoteTab Help file: **^!Help "" Perl**

Clip Language - Mathematics

The Clip syntax is also capable of calculating mathematical expressions. The advantage of the Clipbook is that you can create Clip Wizards to define the values of variables. You can create Clips that will either paste the result in your document or display it in an information window.

^\$Calc(Expression[;Decimals])\$

Calculates mathematical expressions. Use the optional Decimals parameter to define how many decimals should be displayed; if this parameter is not used, integers are returned as whole numbers, and real numbers are configured with two decimal figures.

"Expression" can be any mathematical expression using the operators and functions listed under the Calculate in NoteTab topic. You can optionally define how many decimal values should be displayed in the result by placing a figure after the expression separated by a semicolon (;).

For example: **^\$Calc(7/11;3)\$** will return the result 0.636

You can also add prompts to supply values to variables. The following example will prompt the user to define a value for x and the number of decimals that the answer should display:

^\$Calc(Sin(^?[Sin(x): Enter a value for x]);^?[Decimals])\$

See the Utilities library for some examples.

Old syntax prior to NoteTab 4.51:

Note that since NoteTab 4.8, the old format will not work if the expression is built from the values of variables and functions. You should use the **^\$Calc(Expression[;Decimals])\$** function instead.

The syntax element supports several formats:

^\$[Expression]

or

^\$[Expression;Decimals]

or

^\$[Expression;Decimals]Units=Text on left side of answer

"Expression" can be any mathematical expression using the operators and functions listed under the Calculate in NoteTab topic. You can optionally define how many decimal values should be displayed in the result by placing a figure after the expression separated by a semicolon (;). If you want to place some meaningful text on the left side of the answer, add an equal sign (=) at the end of the tag followed by the text to display. If you want to display units or some other text after the result, just place that information before the equal sign.

For example: **^\$[7/11;3]** will display the result 0.636

You can also add prompts to supply values to variables. The following example will prompt the user to define a value for x and the number of decimals that the answer should display:

^\$[Sin(^?[Sin(x): Enter a value for x]);^?[Decimals]]

The next example:

$^{\wedge}\$[(^?[Fahrenheit]-32)*(5/9)]^{\circ}\text{C}=\wedge?[Fahrenheit]^{\circ}\text{F}$

will produce the following output if you enter the value 64 at the prompt (note the use of the repeated Fahrenheit field, which acts as a reusable variable):

17.78°C=64°F

Replace $^{\wedge}\$$ with the new tag " **$^{\wedge}\!CALC$** " if you want to display the output in a window rather than paste it in the active document.

For example: **$^{\wedge}\!CALC [7/11;3]$**

Clip Language - Dates

`^$GetDate(Date Mask)$`

Returns current date/time formatted according to the date mask.

For example:

This message was written `^$GetDate(mm/dd/yyyy)$`

Pasting it in a document will produce the following text (assuming the date is August 31, 1999):

This message was written 08/31/1999

Old syntax prior to NoteTab 4.51:

Note that since NoteTab 4.8, the old format will not work if the expression is built from the values of variables and functions. You should use the `^$GetDate(mm/dd/yyyy)$` function instead.

If you want to use date/time filters in a Clip, use `^[` and `^]` to delimit the filter element of the Clip text. For example, if you add the following text to the Clip:

This message was written `^[mm/dd/yyyy^]`

Pasting it in a document will produce the following text (assuming the date is August 31, 1999):

This message was written 08/31/1999

Clip Language - Clip Wizard and Fields

Sometimes you may want to complete a Clip text or script before it is actually executed. NoteTab can show Clip Wizards at two different stages during script execution. That is done through the use of square brackets "**^?[...]**" and curly braces "**^?{...}**".

When input fields are defined using the square bracket format, the Wizard is displayed before the first Clip instruction is executed/evaluated. The Wizard is built from all such fields encountered in the script. That is how and why you may see multiple input fields in a single Wizard as soon as you click on the Clip while others using curly braces may take a little while to pop up.

After processing the square brackets, NoteTab processes the script a line at a time, (which does not necessarily mean in sequential order). Before a line is executed, the Clip interpreter checks if it contains any Wizard fields with curly braces . If they are present, a Wizard is then built based on the fields contained in the line.

Additionally, Clip functions and variables are processed as NoteTab encounters them in the script. If a function is used in square brackets **^?[Today's date is=^\$GetDate\$]** the actual function would be seen in the prompt because the square brackets are processed first, but the function will not be processed until after it is reached in the script. On the other hand, since curly braces and functions are processed when the line is processed, the date will be shown in the prompt using **^?{Today's date is=^\$GetDate\$}**. The same is true when using variables.

Wizard controls can be simple text fields, masked fields, multiline fields, combobox fields, listbox fields, and checkbox fields. They are defined by parameters placed between the "**^?[parameters]**" or "**^?{parameters}**" field delimiters. Parameters can have the following structure:

```
FieldCaption
(Options)FieldCaption
FieldCaption=DefaultValue
FieldCaption=ListValues
FieldCaption==ListValues
```

The FieldCaption represents the text used for the field's caption. DefaultValue represents the text displayed in the field, which the user can edit. ListValues represent the text that is used to create the list of values displayed by comboboxes, listboxes, and checkboxes; each value must be separated by a | character. You can optionally precede a list item with an underscore character to make it the default value. Options can define the order in which fields should be displayed, the nature of the control, its height as well as other properties. When multiple options are specified, you must use a semicolon to separate them. The following field options are available:

T=M	Type is multiline control without word wrap. Only use with the " ^?{...} " field format
T=W	Type is multiline control with word wrap. Only use with the " ^?{...} " field format
T=C	Type is combobox (requires ListValues parameter)
T=L	Type is listbox (requires ListValues parameter)
T=A	Type is array of checkboxes (requires ListValues parameter)
T=O	Type is file open dialog box
T=S	Type is file save dialog box
T=D	Type is folder dialog box
T=T	Type is HTML-tag field, for anchor or image elements
C=L	Tag is returned in lowercase format; only used by HTML-tag fields.
C=U	Tag is returned in uppercase format; only used by HTML-tag fields.
F=Filter	Defines a file dialog filter. Use double quotes around the filter
H=Number	Defines a field height measured in lines

S=Number Defines a text-size limit for the field content (measured in characters)
M=Mask Defines a field with an input mask. Use double quotes around the mask
Number Defines field sequence position in Wizard

See also [Clip Basics](#) for step-by-step examples explaining Clip Wizards.

Field Examples:

^[Type a name]

Simple blank text field (one line of text)

^{(M="(000)_000-0000;0;*)Enter telephone number]

Simple field with input mask

^{(T=W)Type memo text}

Multiline field with word wrap.

^{(T=M;H=5)Type memo text}

Multiline field with a height of 5 lines and no word wrap.

^{(T=M;S=20)Type memo text}

Multiline field which accepts text of up to 20 characters long.

^[Choose a value=Value1|Value2|Value3|_DefaultValue4|Value5]

Combobox field with 5 items in the list. DefaultValue4 is selected by default because it is preceded by an underscore character. The single equal sign means the user can also enter values that are not contained in the list.

^{(T=C)Choose a value==Value1|Value2|_DefaultValue3|Value4|Value5]

Combobox field with 5 items in the list. DefaultValue3 is selected by default because it is preceded by an underscore character. The double equal sign means the user can only choose one of the values contained in the list.

^[Prompt==Value1|Value2|Value3|_DefaultValue4|Value5]

Listbox field with 5 items in the list. DefaultValue4 is selected by default because it is preceded by an underscore character.

^{(T=A)Prompt=Value1|_DefaultValue2|Value3|_DefaultValue4|Value5]

Checkbox field with 5 items in the list. Items that are checked by default are preceded by an underscore character. The result returned by this type of field contains all the checked items delimited by a semi-colon and double-quotes around each item.

^{(T=O;F="Delphi Files (*.pas;*.dpr)|*.pas;*.dpr")Open file]

Open File field.

^{(T=O;S=M;F="Delphi Files (*.pas;*.dpr)|*.pas;*.dpr")Open one or more files]

Open multiple files field.

^{(T=S;F="Text Files (*.txt)|*.txt")Save as]

Save file field.

^{(T=T;C=L/U;F="HTML Files|*.htm*;*.asp;*.css")Prompt]

or

^{(T=T;F="Image Files|.gif;*.png;*.jpg;*.jpeg")Prompt]

HTML-tag field

^?[(3)Field1] ^?[(1)Field2] ^?[(2)Field3]

Three fields with position in Wizard defined in parentheses.

Related Commands and Functions:

^!SetWizardLabel "LabelText"

You can change the default main label used in Clip Wizards generated from ^?{} prompts. The scope of this setting is limited to the Clip where it is used. This command must be placed before the ^?{} prompts they should apply to. Assign a blank string to restore the default label text.

^!SetWizardTitle "TitleText"

You can change the default title of Clip Wizards generated from ^?{} prompts, and the title of the messages boxes created by the ^!Info command. The scope of this setting is limited to the Clip where it is used. This command must be placed before the ^?{} prompts they should apply to. Assign a blank string to restore the default title.

^!IfCancel GoToLabel

Jumps to specified label if a Clip Wizard built from " ^?{}" fields has been canceled. If the Clip doesn't contain such a statement or if it does not appear after the call to the Clip Wizard and before the next one, then its execution is aborted.

^\$SetDefaultValue(ValueList;DefaultValue)\$ *(added in v4.6)*

This function can be used to set the default value displayed by the Clip Wizard in combobox or listbox fields. It returns the ValueList string with DefaultValue as the default value. If DefaultValue does not exist in the list, it is added to the ValueList text. If DefaultValue is an empty string, then the function returns a list with no default value set. Clip example (each time you activate this Clip, the previous default value is selected):

```
^!IfFalse ^$IsEmpty(^%ValueList%)$ Skip
^!Set %ValueList%= _1|2|3|4|5|6
^!Set %DefaultValue%=^?{Choose or enter a new default value=^%ValueList%}
^!InsertText Default value = ^%DefaultValue%^p
^!Set %ValueList%=^$SetDefaultValue(^%ValueList%;^%DefaultValue%)$
```

^\$GetDefaultValue(ValueList)\$ *(added in v4.6)*

Returns the default value selected in ValueList. Example: ^\$GetDefaultValue(1|2|_3|4|5|6)\$ -> 3

Clip Language - Masked Input Fields

Use the mask to restrict the characters a user can enter into the masked edit control to valid characters and formats. If the user attempts to enter an invalid character, the edit control does not accept the character. Validation is performed on a character-by-character basis.

A mask consists of three fields with semicolons separating the fields. The first part of the mask is the mask itself. The second part is the character that determines whether the literal characters of a mask are saved as part of the data. The third part of the mask is the character used to represent unentered characters in the mask.

These are the special characters used in the first field of the mask:

Character	Meaning in mask
-----------	-----------------

!	If a ! character appears in the mask, optional characters are represented in the input control as leading blanks. If a ! character is not present, optional characters are represented in the input control as trailing blanks.
>	If a > character appears in the mask, all characters that follow are in uppercase until the end of the mask or until a < character is encountered.
<	If a < character appears in the mask, all characters that follow are in lowercase until the end of the mask or until a > character is encountered.
<>	If these two characters appear together in a mask, no case checking is done and the data is formatted with the case the user uses to enter the data. The character that follows a character is a literal character. Use this character to use any of the mask special characters as a literal in the data.
L	The L character requires an alphabetic character only in this position. For the US, this is A-Z, a-z.
I	The I character permits only an alphabetic character in this position, but doesn't require it.
A	The A character requires an alphanumeric character only in this position. For the US, this is A-Z, a-z, 0-9.
a	The a character permits an alphanumeric character in this position, but doesn't require it.
C	The C character requires an arbitrary character in this position.
c	The c character permits an arbitrary character in this position, but doesn't require it.
0	The 0 character requires a numeric character only in this position.
9	The 9 character permits a numeric character in this position, but doesn't require it.
#	The # character permits a numeric character or a plus or minus sign in this position, but doesn't require it.
:	The : character is used to separate hours, minutes, and seconds in times. If the character that separates hours, minutes, and seconds is different in the regional settings of the Control Panel utility on your computer system, that character is used instead.
/	The / character is used to separate months, days, and years in dates. If the character that separates months, days, and years is different in the regional settings of the Control Panel utility on your computer system, that character is used instead.
;	The ; character is used to separate the three fields of the mask.
_	The _ character automatically inserts spaces into the text. When the user enters characters in the field, the cursor skips the _ character.

Any character that does not appear in the preceding table can appear in the first part of the mask as a literal character. Literal characters must be matched exactly in the edit control. They are inserted automatically, and the cursor skips over them during editing. The special mask characters can also

appear as literal characters if preceded by a backslash character (\).

The second field of the mask is a single character that indicates whether literal characters from the mask should be included as part of the text for the edit control. For example, the mask for a telephone number with area code could be the following string:

```
(000)_000-0000;0;*
```

The 0 in the second field indicates that the Text property for the edit control would consist of the 10 digits that were entered, rather than the 14 characters that make up the telephone number as it appears in the edit control.

A 0 in the second field indicates that literals should not be included, any other character indicates that they should be included.

The third field of the mask is the character that appears in the edit control for blanks (characters that have not been entered). By default, this is the same as the character that stands for literal spaces. The two characters appear the same in an edit window. However, when a user edits the text in a masked edit control, the cursor selects each blank character in turn, and skips over the space character.

NOTE:

When working with multibyte character sets, such as Japanese shift-JIS, each special mask character represents a single byte. To specify double-byte characters using the L, l, A, a, C, or c specifiers, the mask characters must be doubled as well. For example, LL would represent two single-byte alphabetic characters or a one double-byte character. Only single-byte literal characters are supported.

Editor Shortcut Menus

Keyboard Shortcuts

Command-line Parameters

Date and Time filters

Editor Shortcut Menus

Click with the right mouse button on the document and you will have access to a customizable shortcut menu (use the Options dialog box to choose the commands you would like to display).

Use the same mouse button on the tab bar and you will get a choice of commands for managing the open documents.

Most of the editor's other components also have dedicated shortcut menus.

Look at the hint line for more information about a highlighted menu item.

Keyboard Shortcuts

The letter following Ctrl or Shift+Ctrl is not case sensitive.

DOCUMENT

New	Ctrl+N
Open	Ctrl+O
Save	Ctrl+S
Save as	Shift+Ctrl+S
Export	Shift+Ctrl+N
Close document	Ctrl+F4
Close all documents	Ctrl+Shift+F4
Print	Ctrl+P
Display 2 documents	Shift+Ctrl+W
Insert date & time	F5
Insert doc name	Shift+F5
Word wrap	Ctrl+W
Launch document	Ctrl+T

CURSOR MOVEMENT

L/R one character	<- ->
Word right	Ctrl+>
Word left	Ctrl+<
End of line	End
Beginning of line	Home
Previous paragraph	Ctrl+Up
Next paragraph	Ctrl+Down
Next screen up	Page Up
Next screen down	Page Down
Top of screen	Ctrl+Page Up
Bottom of screen	Ctrl+Page Down
End of document	Ctrl+End
Top of document	Ctrl+Home
Move text downwards (NTP)	Shift+Ctrl+Up
Move text upwards (NTP)	Shift+Ctrl+Down

TYPING MODE

Insert/type over	Insert
------------------	--------

EDIT

Undo	Ctrl+Z
Undo	Alt+BkSp
Redo	Shift+Ctrl+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Paste new	Shift+Ctrl+V
Use as paste board	Shift+Ctrl+P
Select all	Ctrl+A
Delete current line	Ctrl+Y
Delete word before cursor	Ctrl+Backspace

Page char. break	Ctrl+Enter
Remove HTML tags	Shift+Ctrl+T
Char. to HTML tags	Shift+Ctrl+H

SEARCH

Find	Ctrl+F
Replace	Ctrl+R
Search again	F3
Search reverse	Shift+F3
Quick find	Ctrl+F3
Search disk	Ctrl+D
Go to hyperlink	Ctrl+H
Go to line	Shift+Ctrl+L
Match brackets	Ctrl+M

TABS, INDENT, BULLET

Tab right	Tab
Tab back	<- Backspace
Tab next	Ctrl+Tab
Previous tab	Shift+Ctrl+Tab
Indent	Ctrl+I
Unindent	Shift+Ctrl+I
Join lines	Ctrl+J
Split lines	Shift+Ctrl+J
Insert bullet	Ctrl+B

OUTLINE

Add heading	Shift+Ctrl+A
Copy heading	Shift+Ctrl+B
Remove heading	Shift+Ctrl+R
Move next	Ctrl+Alt+Down
Move previous	Ctrl+Alt+Up
Move first	Ctrl+Alt+Home
Move last	Ctrl+Alt+End

CHANGE CASE (Must select text)

Invert case	Ctrl+K
Upper case	Ctrl+U
Lower case	Ctrl+L
Capitalize	Shift+Ctrl+K
Sentence	Shift+Ctrl+U

FUNCTION KEYS

F1	Help contents
F2	Expand text
F3	Search again
F4	Clipbook
F5	Date and time
F6	File list
F7	Open file at caret
F8	View in browser

F9	Open link
F10	Move the focus to the menu
F11	Spelling
F12	Thesaurus

BOOKMARKS (NTP)

Toggle bookmark #1	Shift+Ctrl+1
Toggle bookmark #2	Shift+Ctrl+2
...	
Toggle bookmark #0	Shift+Ctrl+0
Move to bookmark #1	Ctrl+1
Move to bookmark #2	Ctrl+2
...	
Move to bookmark #0	Ctrl+0
Next bookmark	Alt+Ctrl+F3
Previous bookmark	Shift+Ctrl+F3

CHANGE FOCUS

To document	Ctrl+G
document <-> Clipbook	Shift+F4
document <-> list	Shift+F6
combo box <-> list	Ctrl+F6
document <-> QuickList	Shift+F7
Left <-> right document	Shift+Ctrl+G

CLIPBOOK

In-Context Clipbook	Esc
Open/Close Clipbook	F4
Add new Clip	Ctrl+F2
Add from Clipboard	Shift+Ctrl+F2
Clip editor	Shift+Ctrl+E
Copy selection	Shift+Ctrl+C
Cut selection	Shift+Ctrl+X
Paste current Clip	Ctrl+Spacebar
Open library #1	Alt+1
Open library #2	Alt+2
...	
Open library #0	Alt+0

FAVORITES

Add favorite document	Shift+Ctrl+D
Add favorite folder	Shift+Ctrl+F
Go to favorites	Shift+F2

E-MAIL

Quote text	Ctrl+Q
Remove Quotes	Shift+Ctrl+Q

TOOLS

Open link	Ctrl+double-click
-----------	-------------------

Tidy HTML Code (<i>requires HTML Tidy</i>)	Ctrl+F7
Validate HTML (<i>requires CSE HTML Validator</i>)	Shift+Ctrl+F7
Other browser	Shift+F8
Launch document	Ctrl+T
Calculate	Ctrl+E
HTMLib Help	Ctrl+F11
HTMLib Color Wizard	Shift+Ctrl+O

Based on QuickRef.txt, written by Edward S. Wozniak

Command-line Parameters

You can specify as many parameters as the command line can take. Each parameter must be separated by a space.

- /NS** Don't show splash screen (must be first parameter). There is also an option in the [Options dialog box](#).
- FileName*** Opens specified file in the editor. File name can also include wildcards. If you want to specify several file names that contain spaces, enclose each name between double quotes, eg.
NoteTab.exe "c:\program files\MyFile1.txt" "c:\program files\MyFile2.txt"
- /F=FavoritesFileName*** Specify an alternative [Favorites](#) list file name (without extension or path). The file name must follow the equal sign. Do not use any spaces before or after the equal. Favorites lists must be in the same folder as the editor. If the file does not exist, it will be created. If you want the files in the list to open automatically on startup, make sure you check the Auto Load Favorites option in the [Options dialog box](#).
- /L=LibraryName:RunClip*** Changes the default Clip library to the one indicated by LibraryName (without extension or path). Optionally use RunClip to specify a Clip to execute once NoteTab is open.
- /C=ClipbarName*** Opens a specific Clipbar when launching NoteTab.
- /T=FileName*** Loads the specified file containing the EBCDIC character tables for conversion to and from Windows ANSI. The NoteTab program path is added if you do not specify a fully qualified path name. The extension .ebc is added if you do not specify a filename extension. See the file Default.ebc for an example of the table format.
- /P** Print the files listed in the command-line parameter.
- /R** Open the files following this parameter as Read-Only.
- /B** Browse mode: the editor will not allow editing of documents.
- /J=nnn*** Jumps to specified line number (nnn) when the editor is opened. This argument can be specified after every file name. If J=-1, the cursor will be placed at the beginning of the last line.
- /INI=FileName*** Specify an alternative INI file. If you don't indicate a path, the program's directory is assumed. If you indicate a path name, there should not be any spaces in it (if this is the case, use the short DOS name instead). You do not need to specify an extension (it is always INI).
- /INST=InstanceID*** Opens a new instance of NoteTab (even if multiple instances is disabled) or activates the instance matching the InstanceID text. Use quotes around InstanceID if it contains spaces. Multiple instances are disabled in instances opened through this method, and documents are not automatically reopened. Example: NoteTab /INST="HTML Editing"

Date and Time filters

The editor has a command that lets you easily insert the current date and time in your documents. You can do so with the Insert Date command (in the Document menu) or automatically if you are using the [log feature](#). By default, it uses the format specified by the *ShortDateFormat* defined by Windows. You can, however, customize the date by creating a Date Filter – the field is available in the [Options dialog box](#) on the Tools tab. There is also a similar field in the [Print dialog box](#). Filters can also be used in text that is added to the [Clipbook](#) (see topic for more information). The following format specifiers are supported:

Specifier	Displays
c	Displays the date using the format given by the ShortDateFormat global variable, followed by the time using the format given by the LongTimeFormat global variable. The time is not displayed if the fractional part of the DateTime value is zero.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames global variable.
dddd	Displays the day as a full name (Sunday-Saturday) using the strings given by the LongDayNames global variable.
dddddd	Displays the date using the format given by the ShortDateFormat global variable.
dddddd	Displays the date using the format given by the LongDateFormat global variable.
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec) using the strings given by the ShortMonthNames global variable.
mmmm	Displays the month as a full name (January-December) using the strings given by the LongMonthNames global variable.
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).
h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (00-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
t	Displays the time using the format given by the ShortTimeFormat global variable.
tt	Displays the time using the format given by the LongTimeFormat global variable.
am/pm	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
ampm	Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.
/	Displays the date separator character given by the DateSeparator global variable.
:	Displays the time separator character given by the TimeSeparator global variable.

'xx' Characters enclosed in single quotes are displayed as-is, and do not affect formatting.

Format specifiers may be written in upper case as well as in lower case letters – both produce the same result.

Support Files and Updates

To keep the size of the commercial distribution package down, dictionaries and the thesaurus file are not included. However, you can download all these files for free by visiting one of the NoteTab Web sites listed below.

<http://www.notetab.com/dictionaries.htm> (main site)

or

<http://www.notetab.ch/dictionaries.htm> (Swiss site)

Please note that NoteTab Light and the trial version of NoteTab Pro do not support the spell checker and thesaurus.

The Web site also provides a large collection of Clipbook libraries created by NoteTab users. Other resources include sounds for the toolbar, "Made with NoteTab" Web page banners, and a list of answers to frequently asked questions.

See the topic [Ordering Information](#) to find out how you can purchase the commercial versions of NoteTab.

Contacting Fookes Software

We have several addresses that you can use for different purposes. Please use the one that is most appropriate for your type of message. Thank you in advance for not sending large attachments.

Note that some e-mails do get lost (the Internet is not perfect), so if you feel you sent an important message and did not get a response, please do not hesitate to send it again.

Sales

For information about pricing, special licensing arrangements, or to inform us of a problem you are having with your purchase, please write to [<sales@notetab.com>](mailto:sales@notetab.com). If you are having problems ordering NoteTab from one of the online distributors, please contact their customer support directly (Digital River: [<custserv@DigitalRiver.com>](mailto:custserv@DigitalRiver.com)).

Marketing

Contact [<marketing@fookes.com>](mailto:marketing@fookes.com) to discuss eventual marketing opportunities with us.

Customer/Technical Support

If you have problems installing NoteTab or using it, please see the topic [Technical Support](#) for details on the various support resources we have to offer.

Feedback

If you like NoteTab and want to encourage us, or if you don't like it and would like to share the reasons with us, or if you have some suggestions to improve it, or some related resources that you would like us to make available to others, then please write to [<feedback@notetab.com>](mailto:feedback@notetab.com).

Bug Reports

Although we do our best to test our software rigorously before releasing any public version, there is always a possibility that an incompatibility problem or a bug will surface when the program is used under certain conditions. If you think you have encountered a bug, please see the topic [Report a Bug](#) for more details.

Author/Owner

The author and owner of Fookes Software, Eric G.V. Fookes, receives a lot of e-mail every day, which he appreciates very much. He reads all messages sent to the [<feedback@notetab.com>](mailto:feedback@notetab.com) and [<bugs@notetab.com>](mailto:bugs@notetab.com) addresses, as well as most of the mailing-list messages. Messages sent to the other addresses are usually forwarded to him if they are considered important for him to see. Regrettably, he only has time to answer a small number of the messages he receives. Those that require a response will usually be handled by technical/customer support. If you want to write directly to the author and understand that he may not be able to respond to you personally, then please send your message to [<author@fookes.com>](mailto:author@fookes.com).

Snail mail at the following address:

Fookes Software
Av. Eugène-Pittard 22 Ter
CH - 1206 Geneva
Switzerland

Fax/Tel.: +41/22-789 58 44

Technical Support

Free technical support is available to **registered** users of NoteTab. However, we reserve the right to terminate support to users who in our judgement are unreasonable or abusive, or who have problems which transcend the scope of this software. If you have a problem installing or running your registered copy of the software, you can ask for help by writing to [<support@notetab.com>](mailto:support@notetab.com). If you have problems ordering or getting your registered copy of NoteTab, please contact the online distributor's customer support (Digital River: [<custserv@DigitalRiver.com>](mailto:custserv@DigitalRiver.com)), or write to [<sales@notetab.com>](mailto:sales@notetab.com). Please do not use these e-mail addresses for questions about using NoteTab. We provide a free [mailing list](#) to help you with all non-customer support issues.

If you have questions about using NoteTab or would like to learn some time-saving tricks from other users, then please join the NoteTab mailing list (even if it is just for one question). It is free for everyone (registered and non-registered users). NoteTab's author frequently answers questions and also uses the list to announce updates and changes to the software. You will find more information about this list on the following Web page:

<http://www.notetab.com/maillist.htm> (main site)

or

<http://www.notetab.ch/maillist.htm> (Swiss site)

You will also find useful information, links, and other NoteTab-related resources (upgrade programs, answers to common questions, etc.) by visiting one of the NoteTab Web sites at:

<http://www.notetab.com> (main site)

or

<http://www.notetab.ch> (Swiss site)

NoteTab Mailing List

The NoteTab mailing list is a great place to meet other NoteTab users – friendly people from all walks of life, and ranging from complete beginners to experts. If you are a "newbie", ask some of the "old timers" some basic questions to help get you started. Also use this list to exchange ideas, share neat tips and tricks, ask questions, suggest new features, etc.

Although the list is not "owned" by the author of NoteTab (it was started and is maintained by Mr Jody Adair, a NoteTab "old timer"), he is an active participant and always tries to answer questions when he finds the time. He frequently sends out information about upgrades, fixes, and future plans for NoteTab. Features that you would like incorporated into NoteTab can be discussed on the list. He does take your suggestions into consideration and has already implemented quite a few made by list members. Also, many times he receives "problem reports" which are not problems at all with NoteTab; sending these to the list increases your chances of receiving a solution from other list members who have dealt with them before you.

You will find more information about the mailing lists and how to subscribe/sign-off on the following Web page:

<http://www.notetab.com/maillist.htm> (main site)

or

<http://www.notetab.ch/maillist.htm> (Swiss site)

Report a Bug

If you think you have found a bug in NoteTab, please do let us know about it as soon as possible. It is thanks to such reports that NoteTab has become so reliable. When sending a bug report, please describe the problem in as much detail as possible. The clearer your feedback, the more likely the author will be able to find the bug and fix it.

Note that not all bugs that you may encounter while using NoteTab are faults in the program. There are many bugs in Windows itself as well as in third-party programs or drivers that can cause your computer to generate errors or work unreliably. Unfortunately, if you encounter such a bug (external origin), there is usually nothing the author can do to fix it, but we will try to inform you if we suspect the problem is not in NoteTab.

Often, though, you can fix external bugs yourself by updating old system files and drivers. Go to the "Product Updates for Windows" Web page to see which system files have been fixed and updated by Microsoft:

<http://www.microsoft.com/windows/downloads/>

Another good source of information on updates can be found at the "Windows 95 Update Center":

<http://www.winmag.com/win95/update95.htm>

Please first check the list of known problems before submitting a bug report. Also verify, if possible, whether the bug has been fixed in the latest update. **Very important:** Remember to indicate which program variant and version number you are using (for example: NoteTab Pro v4.83). If you feel you need to send screen shots of error messages, please save them in the GIF format (not JPG) to reduce their size as much as possible.

Send your bug report to the following e-mail address:

bugs@notetab.com
(no big attachments please!)

Example of a Helpful Bug Report

Subject:

Bug in NoteTab Pro v4.83: file does not open

Message body:

System: Windows 98

Processor: Pentium III 500Mhz

RAM: 64Mb

Bug: File does not open under the following conditions...

Known Issues

If you are having a problem with NoteTab or planning to submit a bug report, please first read through the following list of known problems. Also check, if possible, whether the bug has been fixed in the latest update.

Regular Expressions with Tagged Matches

There is a glitch in the regular expression engine of NoteTab 4.82 and 4.83. It causes tagged matches higher than "\1" in a replace operation to fail. We hope to have this fixed in the next maintenance release of NoteTab.

File Open/Save Dialog Box does not Open

File Open/Save dialog box does not open: The open and save commands use the standard Windows dialog boxes. On some rare computers, Windows does not always respond to NoteTab's request to display these dialog boxes. Several users say they started experiencing this problem after installing IExplorer. Although it never seems to happen in Microsoft's own applications, this problem has been seen in Netscape, Forte Agent, WordPerfect, and other programs. It is due to a bug in some of the DLLs installed by IExplorer. Updating some of the system DLLs seems to have solved the problem for most users. Please visit the Microsoft Web site for such updates:

<http://www.microsoft.com/windows/downloads/>

Toolbar Buttons

The toolbar buttons may not display correctly if you don't have the latest version of a Windows system file. If this is the case on your system, you will have to install a newer version of the ComCtl32.DLL file. You can do this by downloading and installing the following update file (about 500KB) from Microsoft:

<http://www.fookes.com/ftp/50comupd.exe>

Some users have also reported display problems with Windows 98. This had nothing to do with the ComCtl32.DLL file. It turned out that the problem was due to bugs in their display driver and was corrected when they installed a newer update of their driver.

Explorer "Open" Command Limits

If you attempt to open multiple files associated with NoteTab with the Windows Explorer "Open" command, you may encounter problems if NoteTab is not yet running and is configured to disable multiple instances. Usually, in this case, some of the files will not be opened in NoteTab. The solution is to use the "Send To" command instead. The Utilities Clipbook library includes a command to create such a shortcut to NoteTab. The "Send To" command has the advantage of being much faster and more reliable than the "Open" command when you have several files selected for opening.

"Send To" File Count Limit

If you attempt to drag-and-drop or use "Send To" with a many files (for example 10 or more), you may get an error message such as "Access to the specified device, path, or file is denied." This message is triggered by Windows and occurs when the command line exceeds 512 characters. This is a limitation in Windows.

Printer Error on AS/400

If you are getting the 'No Default Printer Selected' error, and you have installed AS/400 Client Access software, you need to get the latest version of the AS/400 Client Access software. Microsoft's applications have some other way of using printers and bypassing the normal procedure, which explains why they don't exhibit this problem.

Default File Extension

If you use a default extension with more than three characters, the Windows File Open/Save dialog box ignores the extra characters. This is an oversight in Windows, not NoteTab. Enter the full extension to

avoid this problem.

Bad Listbox Colors

Systems configured to display only 256 colors will not allow certain color combinations for the editor text and list boxes. Incompatible colors are noticeable because Windows will use a different color for the text background; where there is no text, the correct color is used. This is not a bug in NoteTab. You must try other color combinations to correct the problem.

Undo Paste Operations

The Rich-edit control (a Windows component) used in NoteTab Std/Light cannot undo individual paste operations if these are done one after the other. The control incorrectly assumes this represents a single operation and will undo all the consecutive pastes in one go. To solve this problem, you must enter some text between the paste operations.

OEM Fonts in NoteTab Std/Light

NoteTab Std/Light cannot display fonts with OEM character sets (Terminal, MS Line Draw,...). This limitation is due to the Rich-edit control provided with Windows. NoteTab Pro, however, uses another type of input control that can correctly display such fonts.

NoteTab Std/Light Glitch when Word Wrap Changes

The Rich-edit control (a Windows component) used in Notetab Std/Light has a bug that causes the following four characters: `´` (ALT-0145 ALT-0146 ALT-0147 ALT-0148) to be converted to: ` ' ' '` when you toggle word wrap. Solution: don't change word wrap if you want to use those characters, or use NoteTab Pro (it uses a different kind of input control).

Year 2000 Compliancy

All versions of NoteTab are Year 2000 compliant (Y2K). These products are designed to be used prior to, during, and after calendar year 2000 A.D., and during each such time period will accurately receive, provide, and process data/time data (including, but not limited to, calculating, comparing, and sequencing) from, into and between the twentieth and twenty-first centuries, including the years 1999 and 2000, and leap year calculations, and will not malfunction, cease to function, or provide invalid or incorrect results as a result of data/time data, to the extent that other information technology used in combination with NoteTab properly exchanges data/time data with it.

How to Uninstall

If your NoteTab package came with an installation program (Setup.exe, or something like NTP48.exe or NTS48.exe), you can uninstall it by using the Control Panel's Add/Remove Programs dialog box or the Uninstall NoteTab icon in the Start menu. It is not recommended to use this feature if you have installed NoteTab Std and NoteTab Pro in the same directory; in this case, use the /UNINSTALL parameter as described below and just delete the main program file (NoteTab.exe for NoteTab Light/Std and NotePro.exe for NoteTab Pro).

The following information is only relevant to users of the distribution package that did not include an installation program:

If you have associated file extensions, or set the option to use the registry instead of an INI file, or if you have used the command provided to replace MS Notepad, you can easily restore the registry and MS Notepad by executing NoteTab Std with the following command-line parameter (use the Run command in the Windows Start menu and click on the browse button to find the program; NoteTab should not be running when you use this command):

C:\Program Files\NoteTab Std\NoteTab.exe /UNINSTALL
or
C:\Program Files\NoteTab Pro\NotePro.exe /UNINSTALL

The first example above is for NoteTab Light/Std and the second for NoteTab Pro. Note that the program may be stored in a different location from the example shown above.

If you have dedicated a folder for the program, just delete that folder and all its underlying subdirectories and files (except documents you may have saved there and want to keep!).

Otherwise, just delete all files with the name NOTETAB or NOTEPRO (NOTETAB.* or NOTEPRO.*), all files with the extensions ".FVR", ".OTL", ".TPL", ".CLB", and ".CTB", CLIPHELP.CLH, WHATSNEW.TXT, and this README.TXT file. All these files are located in the same directory as NoteTab, or their corresponding subdirectories.

If you have executed NoteTab from a CD-Rom, you will find the NOTETAB.INI and NOTETAB.FPR (or NOTEPRO.INI and NOTEPRO.FPR respectively) files in the Windows directory.

Fookes Software Products

You probably already know our www.notetab.com Web site, but have you ever been to the main **Fookes Software** Web site?

<http://www.fookes.com/>

You'll find several interesting new products of ours, including freeware, and a portfolio showing beautiful landscape photographs of various places in Europe.

Our Latest Products:

[Mailbag Assistant](#)

[Album Express](#)

[Screen Savers](#)

Mailbag Assistant

If you're buried alive in email, Mailbag Assistant, a powerful and user-friendly message organizer, can come to your rescue with tools to search, organize and archive your overflowing mailboxes -- tools that you've probably found missing or inadequate in your mail program. Mailbag Assistant complements your mail program without interfering with it, leaving your native mailboxes absolutely intact. It reads and processes your current and archived mailboxes with ease, whether they are stored on your computer, on a networked drive, or on removable media.

The search tools are particularly powerful, providing advanced features like Boolean operators (And, Or, Not), along with easy-to-use wizards. In addition to searching for exact matches in your emails, Mailbag Assistant lets you search for approximate matches and Soundex (sound-based) matches. These tools make it easy to define, group and find relevant messages quickly and accurately.

Novice users will enjoy Mailbag Assistant's easy-to-use grid layout, ability to sort emails simply by clicking a column heading, and Quick-Match searches. Power users will enjoy Mailbag Assistant's tools for creating subsets of messages, extracting addresses and attachments, and compiling traffic and word statistics. There's also a full-featured scripting language that lets you automate repetitive tasks like opening mailboxes, searching messages, and archiving emails. Advanced users will appreciate its powerful command-line interface support.

Whether you're maintaining mailboxes with thousands of emails, or just want to save time hunting down a few old ones, Mailbag Assistant makes it much easier to search your mailboxes, organize your emails efficiently, and archive them in an easy-to-retrieve way.

Mailbag Assistant requires Windows 95/98/ME/NT4/2000 (or higher), and supports mailboxes for Outlook Express (including v5.x), Eudora, Netscape Messenger (including v6), Pegasus, Forte Agent, The Bat!, Calypso, Poco, and FoxMail.

For more information and to download a free evaluation version, please visit the following Web page:

<http://www.fookes.com/mailbag/>

Album Express

Album Express is a great new Windows tool for creating slick, professional-looking picture albums for home, friends, and the Web. Its wizard-like interface and built-in templates make it easy to present your pictures in style: you can produce a slideshow, create an index, catalog your pictures, and build a fully searchable image database -- all with just a few clicks of the mouse.

If you're a computer novice, you'll love this program's ease of use. Just select the pictures you want, select a template and watch Album Express assemble your creation in no time --editing Web pages was never like this! Your picture collection comes complete with standard navigation tools, and you can even include a background image for added impact. Whether you want to view your albums at home, using your favourite browser, or upload them to your home page on the Internet, it's as easy as clicking on a mouse button.

Album Express is also an ideal tool for power users. Its full-featured scripting language will enable you to build your own album templates, allowing almost unlimited creativity in developing display styles and techniques. Demonstration templates are provided to show you some of the possibilities: you can build framed pages, allowing pictures to be selected from thumbnail images, or take advantage of the multimedia potential of HTML by associating pictures with MIDI tracks or sound clips -- the sky's the limit.

Remember that if a picture can convey a thousand words, your picture albums can build a virtual library!

For more information and to download a free evaluation version, please visit the following Web page:

<http://www.fookes.com/album/>

Screen Savers

Fookes Software is proud to offer a whole range of high-quality landscape screen savers. We favor quality over quantity. Unlike many pictorial screen savers, the included photographs look good even under high-resolution screen settings (up to 1600x1200). Each screen saver contains 30 pictures, and free time-limited trial versions are available from our Web site. The following screen savers are available:

Absolute Europe

High-resolution photographs of Holland, Ireland, Scandinavia, Scotland, and Switzerland.

Absolute Finland

High-resolution photographs of Southern Finland and Lapland.

Absolute Norway

High-resolution photographs of various areas, including the Lofoten Islands.

Absolute Scandinavia

High-resolution photographs of Denmark, Finland, Norway, and Sweden.

Absolute Switzerland

High-resolution photographs of the Alps and Pre-Alps areas.

For more information and to download free sample copies, please visit the following Web page:

<http://www.fookes.com/screensavers/>

Third-Party Products

There are several products that we believe will appeal to NoteTab users. Anyone using NoteTab to create Web pages should get a copy of [HTML Tidy](#). It's free and does a great job of cleaning up your HTML code right from within NoteTab. If you develop Web pages with Cascading Style Sheets (CSS), then you owe it to yourself to try the free copy of [TopStyle](#). This program works seamlessly from NoteTab. To ensure that you write syntactically correct HTML, nothing beats the powerful [CSE HTML Validator](#). The Pro version of this program also works seamlessly from NoteTab. And for those of you who appreciate NoteTab's tabbed interface, you'll love surfing the Web with [NetCaptor](#), which uses a similar interface.

For those of you who are interested in developing NoteTab libraries with Clip code, make sure you take a look at [ClipWriter](#). This program, written by an expert Clip programmer, is a powerful alternative to the NoteTab Clipbook editor. A free trial version is available for evaluation.

[HTML Tidy](#)
[TopStyle CSS editor](#)
[CSE HTML Validator](#)
[NetCaptor](#)

[ClipWriter](#)

HTML Tidy

NoteTab supports HTML Tidy -- a free utility that cleans HTML code and fixes errors. Once it is installed, you can invoke it through the "Tools|Tidy HTML Code" menu command. HTML Tidy accepts a variety of switches to control the way it processes HTML tags. NoteTab will try to determine the best switches based on the active document type and the HTML Tag option settings. If you want more control over the settings, you can create a configuration file for Tidy, which you should save in the NoteTab folder with the name Tidy.cfg (see the HTML Tidy instructions for details). You can learn more about HTML Tidy and download the latest version by visiting the following Web page:

<http://www.w3.org/People/Raggett/tidy/>

The easiest setup is to copy the Tidy.exe program file to the NoteTab folder, but you may place it anywhere else if you prefer.

TopStyle CSS editor

TopStyle provides powerful tools to create cross-browser style sheets. Features such as the property inspector and style checker ensure that your style sheets work in all major browsers, while site-wide style management features make applying styles across multiple pages a breeze. TopStyle includes full support for CSS1 and CSS2, as well as for CSS implementations from Microsoft, Netscape, WebTV Plus and Opera Software. TopStyle validates your style sheets against any (or all) of these implementations, warning you not only about errors in your style sheets, but also about bugs in popular browsers that may affect their display.

NoteTab integrates seamlessly with TopStyle version 1.51 and above. It offers access to TopStyle commands through the TopStyle library and Clipbar.

To learn more about this useful program and download a free version, please visit the following Web page:

<http://www.bradsoft.com/redir.asp?id=130>

CSE HTML Validator

CSE HTML Validator Professional is a fast, powerful, inexpensive, highly user configurable, and easy to use professional HTML development tool that will help you to write syntactically correct HTML. You'll be surprised how many problems may exist in your HTML, even if you made it with a WYSIWYG editor. HTML Validator will help you locate and fix HTML syntax problems so that your documents are easily viewed by a variety of Web browsers. In addition, HTML Validator will give you valuable and customizable advice based on your HTML document. CSE is great for both HTML beginners and professionals who want to make sure that their Website visitors do not run into problems caused by poorly written HTML. Version 4.0 now includes link checking, a powerful, easy to use, integrated HTML/text editor, and a faster, more reliable batch wizard that lets you easily validate multiple HTML documents and generate a single validation report in HTML. CSE is a must for any business with a Website.

NoteTab integrates seamlessly with CSE HTML Validator Professional (v2.00A and higher). Once the program is correctly installed, you will see a new command under NoteTab's Tools menu called "Validate HTML". When you click on this command, it will send the current document text to CSE HTML Validator. Once the checking is completed, NoteTab will create a Clipbook library containing all the messages produced by the CSE program. Executing the Clips in the library will select the offending line and display the corresponding error message. You can learn more about CSE HTML Validator and download the latest version by visiting the following Web page:

<http://www.htmlvalidator.com/>

NetCaptor

If you like the tabbed interface in NoteTab, you will love Netcaptor...

NetCaptor turns the browser world upside down. Unlike other browsers which only view one site at a time or crowd multiple windows together, NetCaptor opens sites on separate browser tabs. In addition to browsing multiple Web sites at the same time, NetCaptor allows you to customize toolbars; open groups of sites (CaptorGroups) with a single click; drag and drop URLs on other browser tabs; automatically close pop-up windows; turn the loading of images, sounds, videos, and animations on and off; and limit the displayed title length. Other features include the ability to name tabs, a configurable search button that supports multiple search engines, keyboard shortcuts, and more. NetCaptor requires Microsoft Internet Explorer 4.x or newer for operation.

To learn more about this program and download the free version, please visit the following Web page:

<http://www.netcaptor.com/>

ClipWriter

ClipWriter is a powerful alternative to the NoteTab Clipbook Editor. ClipWriter makes it easy to insert keywords with signs like "^", "!", "\$", and "%", and it allows you to define the shortcut keys that best suit your style. It comes already with more than 530 predefined shortcut keys covering over 400 Clip commands and functions.

ClipWriter supports permanent bookmarks and allows inspecting specific parts of your code such as variables, labels and instructions with direct access to them at design time. It fully integrates with NoteTab from where you can launch it using a Toolbar or Clipbar button! And you can run your Clips directly from ClipWriter.

Visit the following link for more information...

<http://www.notetab.com/clipwriter/>

Acknowledgments

The author would like to thank his colleague and friend, **Christophe Nussbaumer**, who was one of the first NoteTab users. It is thanks to him and his encouragements that NoteTab became available on the Internet.

Since then, many NoteTab users – unfortunately too many to mention here – have provided helpful feedback and interesting suggestions to improve the software. Thanks to their comments, NoteTab has been enriched with many significant improvements.

Many thanks to the following people who provided an abundance of helpful feedback and/or resources while NoteTab 4.8 was undergoing development and beta testing:

Jody Adair
Robert Binkley
Jelks Cabaniss
Cees
Dennis Cummins
Dave (aka Doc)
Stefan Elssner
Ernie
Raymond Fitzgerald
Grant
Jim Hall
Larry Hamilton, Jr.
Dr. Paul Harder
Leonard O. Hargrove, Jr.
Peter G. Henry
David Just
Bill Kemper
Lloyd
Mark Pulver
Charles M. Raine
Eric Richards
Stephen Riddle
Kay Roath
RS
Erich Ruff
Harvey R. Savage
Carl Swann
Brett Taylor
Larry Thomas
Wayne VanWeerthuisen

David Nourse was a great help in writing short program descriptions that convey clearly what NoteTab is about. His writing skills are much appreciated. The Press Release was produced by **Al Harberg**, a real expert in this field (www.dpdirectory.com).

For versions 4.5 and 4.6, the Fookes Software is particularly indebted to the abundant feedback it has received from members of the NoteTab mailing lists. Users have provided many useful suggestions and helped to fix many bugs and small glitches. Unfortunately, it is impossible to name all those who contributed to these versions, so please forgive us for just listing the most active participants:

Jody Adair
Sami Järvinen
Kay Roath
Len
R.S.
Nicole Simon
Patrick Steele
Larry Thomas
Wayne VanWeerthuisen

Also, a special thank you to **Raymond Fitzgerald** for reading through all the software documentation and detailing the errors that needed correcting.

Version 4.0 is one of the most significant updates. It has also undergone some of the most intensive beta testing since NoteTab was introduced two years ago. Fookes Software is therefore much indebted to the following people who have spent much time testing the software and offering valuable suggestions:

Jody Adair
José María Alín
Mike Arst
Paul Arzul
Paul Harder
Greg R. Haynes
Staffan Höijer
Judith
Helmut Lange
Tom Lidstone
J. Michael Mollohan
Helmut Reimer
Kurt Rosenfeld
Edward S. Wozniak

Several application icons are available if you do not like the default one (see View tab in Options dialog box). Thanks to **Jon Aske**, **Scott Klafke**, **Paul Arzul**, and **Slava Shlyaptsev** for their contributions (not to mention **Arthur Smith**, who had the idea for the main program icon – the Swiss flag). To keep the file size down, Fookes Software had to limit the number of icons included in the program. So many thanks to the other artists who sent icons too.

The following people have provided much valuable feedback on earlier versions of the program:

Mat Davis
Bob Dolan
Dennis Gallagher
Chris Glockler
Tobias Graening
Leonardo Herzenberg
Christoph Juengling
Dr. Jürgen Kehrel
Rob McGee
Mark Paciga
David Pottier
Ken Stuart
Frank Thurber

The mathematics feature was made possible thanks to the excellent freeware parsing component

developed by the following people:

Renate Schaaf
Alin Flaider
Stefan Hoffmeister

The spell checker and thesaurus components used in the commercial versions of NoteTab Standard and NoteTab Pro are produced by *Addictive Software* (www.flinthills.com/~addict).

NoteTab Pro uses a modified version of the Orpheus input control developed by *TurboPower* (www.turbopower.com).

www.notetab.ch is hosted courtesy of **Deckpoint** at www.deckpoint.ch

