



Version 4.4.0 — 15 October 2025

Published by Just Great Software Co. Ltd.

Copyright © 2004–2025 Jan Goyvaerts. All rights reserved.

“AceText” and “Just Great Software” are trademarks of Jan Goyvaerts

Table of Contents

How to Use AceText.....	1
1. Getting Started with AceText.....	3
2. Capture, Enter and Store Text	8
3. Add Emphasis or Syntax Coloring	10
4. Manage and Arrange Text.....	12
5. Manage RSS Feeds	14
6. RSS Feed Writer Settings.....	17
7. Editing Clips and Text.....	19
8. Editing Text with The Keyboard.....	21
9. Editing Text with The Mouse.....	24
10. Look up and Reuse Text	25
11. Share Experiences and Get Help on The User Forums	27
AceText Reference	33
1. AceText Tab Pages.....	35
2. Main Toolbar Reference.....	44
3. Collection Toolbar Reference.....	59
4. Collection Menu.....	75
5. Operation Menu	84
6. Clip Menu	85
7. Edit Menu.....	88
8. Text Menu.....	90
9. Convert Menu	95
10. Format Menu	108
11. Options Menu.....	117
12. Help Menu.....	119
13. AceText Tray Icon	121
14. Searching and Replacing.....	126
15. Parameters	135
16. Keyboard Shortcuts	138
17. Appearance Preferences	141
18. Syntax Preferences.....	153
19. Cursors Preferences	166
20. Tabs Preferences.....	168
21. Files Preferences	171
22. Hotkeys Preferences	175
23. Operation Preferences.....	180
24. Applications Preferences.....	184
25. Windows Preferences	190
26. AceText-Aware Applications.....	193

Part 1

How to Use AceText

1. Getting Started with AceText

AceText is a versatile text manager. A companion that eases and speeds up your everyday computer activities, whether that is writing, text editing, programming, searching and collecting information, sending and responding to email, messaging and chatting, or any or all of those.

What these activities have in common is that you are working with text. AceText simplifies working with text by enabling you to store, manage, edit and reuse text from a central location. For maximum convenience and ease of use, AceText enables you to accomplish these four basic tasks in many different ways. Which ways you follow depends on your habits, and the software you are working with. As you become more experienced with AceText, you will find that all these abilities make AceText a very transparent and helpful tool, indispensable when working with text.

To get started with AceText, click the New Collection button. Click the  New Clip button to jot down notes, prepare a message, etc. Click the  Paste as New Clip button to store text you copied from another application. Or drag and drop text directly from other applications into AceText. Reuse messages and text snippets by dragging and dropping them from AceText into other applications. Or press Win+Alt+V while working with another application and then double-click a clip to paste it directly into the application.

Extending the Windows Clipboard

AceText extends the Windows clipboard, enabling you to copy and paste like never before.

The Windows clipboard holds only one piece of text at a time. AceText holds as many as you want (a thousand by default) in the ClipHistory. Whenever you cut or copy text, AceText captures it into its ClipHistory while  Automatic Capture is enabled. AceText saves the ClipHistory even when you reboot your computer. Never again lose text you copied to the clipboard.

Double-click any clip in the ClipHistory to quickly paste it anywhere you want. Turn on SequencePaste to quickly paste many clips in succession. Rapidly transfer data from one application to another by first copying all snippets in the first application, and then pasting everything into the second with SequencePaste. Or use  AcePaste to quickly paste anything you copied in the past day, week or month.

Configure the ClipHistory on the Operation page in the Preferences screen.

Write Down and Store Notes and Ideas

When a thought hits your mind, write it down in AceText. When you come across useful information on a web site or in a document, copy and paste or drag and drop it into AceText. Never again forget a good idea or lose a scrap of paper with valuable notes.

Use AceText's full-featured text editor to take notes and write down ideas at any time. Quickly jot down a brief thought, or hold a long brainstorming session writing down, organizing and sorting out ideas. Easily store web site addresses, text snippets or even whole documents. Drag and drop or paste them directly into an AceText collection. Move or copy previously captured text from the ClipHistory and other AceText collections.

Recall and dismiss AceText at your whim with a single key press or mouse click, without worrying about saving. AceText automatically saves anything you collect. Instantly search or filter through one or more collection of notes to quickly dig up the information you need.

Quickly Respond to Email with Informative Replies

Send courteous and informative reply emails faster than you used to send out brief responses. Impress people with complete sentences without cryptic abbreviations when chatting or instant messaging, without slowing down or getting your fingers in a bind. The trick is to keep email and messaging templates in AceText.

Next time you receive an email with a question you may get again later on, write a polite and comprehensive reply. Before sending it off, copy your response into AceText, and label it with a recap of the question. If you're busy, simply copy the response to the clipboard. Later, retrieve the message from the ClipHistory, and file it with your collection of email templates.

Now, whenever you get the same question, send out the same helpful reply with just a few mouse clicks, or a quick search in AceText. Use AcePaste to directly paste the message into your email software or web-based email. Never again type in the same message twice. Reduce follow-up questions and impress your correspondents with clear and complete answers.

There are no limits or restrictions to the number of boilerplate messages you can create. Easily organize them in collections and folders. The instant full-text search helps you to quickly find the exact message you need.

Send personalized responses by creating a variety of templates with greetings, signatures, etc. Create separate templates for each question, containing only the answer to that question. Then when you reply to a message, you can quickly compose an individual response by selecting several message parts in AceText, providing exactly the information your customer was looking for. If you paste your reply in-line with the email you received, rather than bottom-quoting the entire message, people won't even know they got a standard reply.

Make people feel valued by personalizing your replies with parameters such as %FIRSTNAME%. AceText will substitute them as often as you like. Type in frequently changing parameters just once, or permanently store rarely changing parameters for automatic substitution.

Use Colorful Messages Instead of Cryptic Abbreviations when Chatting

Send courteous and informative reply emails faster than you used to send out brief responses. Impress people with complete sentences without cryptic abbreviations when chatting or instant messaging, without slowing down or getting your fingers in a bind. The trick is to keep email and messaging templates in AceText.

When instant messaging or chatting, prepare colorful messages such as "I'll be back in a minute. Don't be a stranger now!" and assign them AceType abbreviations such as "brb". When chatting, press the AceType hotkey and type "brb" to send out the colorful note instead of the boring acronym, using only one additional key press. Impress people with your courteous or funny messages, with little or no extra effort.

Use multiple AceText collections with the same abbreviations but different messages, for different days of the week, moods or weather conditions. Bring variety and brighten up the day of your correspondents!

Make people feel valued by personalizing your messages, such as frequently mentioning their name or company. Enter a few personal details in parameters before or during the start of the chat, and instantly reuse them during the chat, even as part of longer messages. Never embarrass yourself by mistyping your contact's name or getting names mixed up.

Speed Up Writing and Developing

Write emails, reports and documents faster, or develop web sites and software more quickly by storing common blocks of text, formatting tags or source code snippets in AceText. Press the AcePaste hotkey to pop up AceText while you're working with your text editor (or any other application). Double-click a clip to paste it directly into your editor, and continue working with your editor right away.

Assign abbreviations or mnemonics to your text snippets, and use AceType to quickly insert the snippets into your editor by typing in the abbreviations. AceText shows a list of clips with abbreviations, filtering them as you type, so forgetting an abbreviation won't slow you down. All without lifting your hands off the keyboard or interrupting your flow while writing or editing.

If you have a large monitor, make the AceText Tower permanently visible next to your editor. Reuse text snippets or copy snippets into AceText simply by dragging and dropping them with the mouse.

Edit Complex Documents Quickly and Easily

Use AceText as a temporary storage pad during complex text editing tasks. Never again lose an important chunk of your document because you copied something to the Windows clipboard, deleting the text you copied a minute ago which you forgot to paste first. AceText keeps everything you copy to the clipboard in its ClipHistory, transparently extending the Windows clipboard. Use AcePaste to quickly select and paste any text clipping from the ClipHistory.

AceText's Convert menu has many commands for converting between uppercase and lowercase, between straight and smart quotes, tabs and spaces, and various character escapes. The Format menu has many commands for formatting, sorting, and rearranging lines. You can easily copy text from another application, edit it in AceText, and then paste it back into the same application. With  AceEdit you can invoke AceText for this purpose while working with the other application.

If your computer has a large monitor, you can split AceText into the AceText Tower and AceText Editor, and tile them on the screen with the word processor or other application you're working with. Then you can easily move text about between AceText and the other software with copy and paste or drag and drop. Store text as a new clip in AceText, or insert it in the middle of another clip. AceText has a full-featured text editor built-in, enabling you to conveniently prepare or edit snippets before adding them to your document.

Keep a Digital Scrapbook or Diary

Keep a diary in AceText simply by typing your thoughts into AceText's full-featured text editor. Keep a scrapbook by copying and pasting or dragging and dropping text from a web site or any application. AceText automatically tags your notes with the date you first created or last updated them, as you prefer. Web snippets are automatically tagged with their source URL, so you can instantly revisit the web page.

Sort your entries by date, and organize them into folders as you like. AceText's handy calendar marks days with events, and lets you travel through time freely by filtering your collections on any date range and/or keywords.

Protect your private journal with AceText's unbreakable "blowfish" encryption. Then you can even hide particularly sensitive notes as secrets. But don't forget your password!

Gather, Research and Compile Information

Gather information by copying and pasting or dragging and dropping text from a web site or any application. AceText automatically tags your clips with the date you captured them. Web snippets are automatically tagged with their source URL, so you can instantly revisit the web page. AceText's ClipHistory automatically captures anything you copy to the clipboard. Easily file items from the ClipHistory into your collections, even hours or days after you found the information. Add RSS feeds from news sources to your collections to automatically download and save the information for later reference, or immediate reading.

Research the information you've collected by sorting and filtering your collections on words, regular expressions and dates. Search through a single clip, all clips in a collection, or all collections you've opened. Organize relevant information in folders or hierarchical trees.

Create and Publish a Knowledge Base

If you provide technical support, or you're simply an expert on a particular topic, store your knowledge in AceText. Write down tips and answers in AceText's full-featured text editor. Copy and paste or drag and drop information from web sites and other sources directly into AceText. Web snippets automatically point back to their online sources. Organize your knowledge base into collections and folders separated by topic.

Respond to inquiries quickly by pasting common answers directly into your email and online forum messages. Find the right information with AceText's instant searching and filtering abilities.

Easily publish an FAQ or a newsletter as an RSS newsfeed that people can easily track and read in a web browser or news reader. Follow other people's news feeds and save their knowledge directly into your AceText collections.

Keep Your Secrets Safe

If you protect a collection with a password then you can designate clips as being secret. AceText does not show the contents of secret clips unless you click the Reveal button. The secret is hidden automatically after a configurable number of seconds, unless you're busy editing the secret. When you use  AcePaste to transfer a secret to another application, AceText always does so by simulating keyboard shortcuts. AceText does not transfer secrets via the clipboard as that would make it all too easy for other applications to snoop on your secrets or even for them to be accidentally pasted more than once.

If you have other applications like password managers that do place sensitive data on the clipboard, you can configure those so that AceText won't automatically show or capture their data on the Clipboard or

ClipHistory tabs. If you then manually paste data from such an application into a password-protected collection then AceText automatically treats the new clip as a secret.

Or you could use AceText itself as your password manager. The  Generate Random Clip lets you generate new secrets with random characters strings that must or mustn't contain certain characters, as is often required of passwords. When a collection is protected with a password, AceText uses a slow algorithm (PBKDF2) to derive a 384-bit encryption key to make guessing the password infeasible. It then uses a hereto unbroken encryption algorithm (blowfish) to encrypt the collection. Without the password, the contents of the collection cannot be recovered.

2. Capture, Enter and Store Text

AceText provides you with a central location to store snippets of text. A piece of text is called a “clip” in AceText parlance. A single clip can contain a word, a sentence, a page of text, or the complete works of Shakespeare.

AceText supports several different kinds of clips. Most of the time, you will work with plain text clips. The other kinds are useful in conjunction with AceText-aware applications.

- Plain text: An ordinary piece of text, compatible with all software applications.
- Rectangular text block: A rectangular block of text, or a column selection. AceText-aware applications recognize AceText’s rectangular format. AceText also supports the clipboard formats for rectangular blocks used by the Visual Studio and Embarcadero IDEs. This allows you to copy and paste rectangular blocks between AceText and applications that support these formats. Other applications will treat the clip as several lines of plain text.
- Before and after text: Two pieces of plain text, where the “before” part is intended to be inserted in front of another block of text, and the “after” part behind it.
- Text with emphasis: Text that allows you to change font styles and highlight colors to add emphasis.
- Binary data: Text containing non-printable characters, displayed in hexadecimal format in AceText. AceText-aware applications can paste the bytes directly. Other applications will see a hexadecimal representation of the bytes on the clipboard. Most hex editors will interpret this correctly, allowing you to paste the actual bytes.
- Secret: A secret clip stores plain text. But AceText doesn’t show you the text unless you click the Reveal button. Only collections protected with a password can store secret clips. The actual contents of the clip are plain text. You can send secret clips directly to another application without revealing the secret if the other application can accept keystrokes without revealing them. An input field for a password, for example, fits that bill.

Clips are stored in AceText Collections. To start with a new collection, click the New button on AceText’s main toolbar. To continue working with an existing collection, click the Open button. AceText will automatically and regularly save your collection files (unless you turned this off in the Files preferences). This way you never have to worry about losing any clips.

There are many ways to add text as a new clip into AceText.

Directly Type In Text

If the text you want to store in AceText is not stored anywhere on your computer yet, you can simply type it in directly into AceText. Click the  New Clip button in the Collection toolbar, or press F4 on the keyboard. Then click inside the large text box, or press Alt+T on the keyboard, and start typing.

Copy And Paste

Copy and paste is the traditional way to transfer text between different applications, and is supported by all software. Simply copy the text to the clipboard. To paste it into AceText as a new clip, click the  Paste as New Clip button in the Collection toolbar, or press Shift+Ctrl+V on the keyboard. To paste the text into an existing clip, select the clip, click inside the text box or press Alt+T on the keyboard, and then press Ctrl+V to paste.

Copy Without Paste, Via The Cliphistory

By default, AceText will automatically grab all text that you copy to the clipboard, regardless of the application you copied it from, while AceText is running. AceText is running when the AceText icon is visible next to the system clock. Windows may hide the icon even when AceText is running; make sure to make it visible. Each time you copy some text, a new clip is added to the top of the ClipHistory. You can add one or more of those clips to a collection by clicking on the ClipHistory tab, and then clicking the  Duplicate Clips or  Move Clips button, or pressing Ctrl+D (duplicate) or Ctrl+M (move) on the keyboard. Finally, select the collection to move or duplicate the clips into.

Using the ClipHistory is much handier and quicker than manual copy and paste if you want to store several snippets of text in quick succession into AceText. The ClipHistory also enables you to retrieve or paste a clip that you copied to the clipboard a long time ago, and was already removed from the clipboard by subsequent copy operations. By default, the ClipHistory stores the last ten thousand clips. You can configure the ClipHistory's automatic capture in the Operation Preferences, and manual capture in the Hotkeys Preferences.

You can configure the automatic capture in the Operation Preferences, and manual capture in the Hotkeys Preferences.

Drag And Drop

Drag and drop is a very easy and direct way to transfer text using the mouse only. Unfortunately, quite a lot of software does not support drag and drop between itself and other software. This method is well-suited to grab text from a web page while surfing, or from a PDF file while reading in Acrobat. First, select the text. Then click inside the selection, hold the mouse button down, and move the mouse pointer over the AceText window. If AceText is invisible, move the mouse pointer over AceText's taskbar button (the button, not the icon next to the system clock), and AceText will pop up after a short delay. If you move the mouse pointer on top of a tab while dragging, that tab will become active.

To drop the text as a new clip into AceText, either drop it on the collection's tab, or on the tree with clips at the left hand side in the AceText Editor, or in the AceText Tower. If you drop it inside the text box, the text will be inserted into the current clip, at the position where you dropped it.

Opening Files

If the text is already stored in a text file on your computer, click the  Create Clip from Text File button in the Collection toolbar, or press Shift+Ctrl+O on the keyboard. Select the file, and its contents will be added to the collection as a new clip, using the file's name as the label. If you select more than one file, they will all be added in separate clips.

If you want to add all the files from a folder then use  Create Folder from Disk.

3. Add Emphasis or Syntax Coloring

If you use the drop-down list on the AceText Editor to change a clip's kind to "text with emphasis" then you can add emphasis or highlighting to the clip's text. The drop-down list to the right of the one with clip kinds changes to show a list of named colors. The first color is "plain text" which does not have any effect. To add emphasis, select the text you want to emphasize and then select one of the other named colors from "Emphasis" to "Fountain highlight 10" from the drop-down list. The color of the selected text changes immediately. You can repeat this as many times as you like to emphasize as many different parts of your clip in any of the available colors as you like.

For all other kinds of clips, the second drop-down list shows a list of syntax coloring schemes. The first item is "no syntax coloring". This means the clip does not have a syntax coloring scheme of its own. All the other items are the names of the syntax coloring schemes available to AceText. Selecting one immediately applies its colors to the clip's contents. The colors themselves do not become part of the clip. AceText only remembers which syntax coloring scheme you selected for the clip. If you share your AceText collection with others then they will need to have all the syntax coloring schemes that you've used in your collection to see the highlighting. You can download syntax coloring schemes shared by other AceText users. You can also download the scheme editor to edit any of the provided syntax coloring schemes or to create your own.

AceText may still highlight URLs and email addresses in clips without a syntax coloring scheme. That includes clips with emphasis. You can enable or disable this for all clips without syntax coloring in the Appearance Preferences.

Named Colors to Actual Colors

When you add emphasis to a clip, you select named colors such as "Comment" or "Hint Message". You don't select actual colors like "green" or "yellow". The syntax coloring schemes work the same way. They apply different named colors to different parts of the clip. The schemes do have a longer list of named colors to pick from than clips to which you apply your own emphasis.

The named colors are translated into actual colors by the color palette. The color palette that you select in the Appearance Preferences determines the actual colors for each named color when AceText displays your clip.

Though this may seem cumbersome or restrictive at first, it makes emphasis and syntax coloring much more flexible. If you share your collections with others, you won't know if they will be using AceText's light theme or dark theme. You may even switch between themes yourself depending on the time of day and how well lit your room is. A color that provides nice contrast and emphasis on a white background may be hard to see on a black background, or vice versa. Basing both the editor's background color and the emphasis colors on the same palette ensures the colors always provide the desired amount of contrast. AceText comes with a choice of color palettes that provide more or less contrast.

While you might think that green and red would clearly convey the intended meaning (good and bad) of the emphasis you added to a clip, your color-blind colleague may have a hard time seeing any difference. AceText includes several color palettes for the most common types of color-blindness. They allow color-blind people to get the full benefits from syntax coloring with the reduced set of colors they can distinguish. AceText even comes with monochrome palettes that use brightness and various underlining styles for emphasis instead of colors for those who think that modern editors use too many colors.

Even colors with obvious names like “underline” or “strong” aren’t hardwired to make the text underlined or bold. Some people think underlining is bad style. Someone else may prefer a monospaced font that is wider when it’s bold, meaning its not perfectly monospaced when mixing plain and bold text. They may prefer to customize their palette to make sure none of the text is made bold.

When you print a collection you can choose a palette for printing in the print preview. You’ll probably want to choose one with a pure white background to avoid wasting ink. AceText includes several printing palettes for this purpose. When you export a collection to HTML you can choose a color palette in the export dialog. You can pick a palette that matches the colors of the rest of your website if you plan to publish the HTML file.

When you send a clip with emphasis or syntax coloring to another application AceText can copy the clip in HTML or in RTF format to the clipboard. You can enable this for each application in the Applications Preferences. You can also select a color palette that should be used to generate the HTML or RTF separately for each application. You could select a printing palette when sending clips to a word processor in which you prepare documents that will be printed. You could select a dark palette when sending clips to a web editor in which you edit a website with a dark theme.

In all of these places where you can select a palette you’ll have a button that opens the Customize Editor Palette dialog. This dialog lets you pick one of AceText’s many predefined palettes. It also lets you use any of those as the basis for a custom palette. All the colors and font styles used by AceText’s editor control can be fully customized.

4. Manage and Arrange Text

Once you've added a sizable number of clips to a collection, arranging the clips inside the collection or moving them between collections will enable you to access and the clips much faster. When clips are arranged logically, you can select them directly from the list, without having to search for them (even though AceText makes searching quick and flexible).

Group Clips Into Folders

If a collection contains a large number of clips, grouping the clips into folders makes it easier to look up clips later. Just like arranging files in folders makes it easy to find the correct file later. To place clips in a folder, select those and click the  New Folder button. Then type in the name of the folder. If you do not select two or more clips first, an empty folder will be created. Either way, you can add more clips to the folder later.

Move Clips Between Folders

To move one or more clips from one folder into another, select the clips in the list. Then click on the clips, hold the mouse button down, and drag them to the other folder. Either drop them onto the folder itself, or on one of the clips already inside that folder. You can also place folders inside other folders this way.

Change The Order Of Clips

Clips are easier to find if they are stored in a logical order. The easiest way is to just click the  Sort Collection button. You can then choose whether to sort clips and folders alphabetically or by date. If you prefer to manually order the clips, use the  Move Up and  Move Down buttons, or drag and drop them with the mouse.

Move Clips Between Collections

Click the  Move Clips button to move the selected clips and/or folders into another collection. If the selected clips and folders are inside one or more folders, the same folder structure will be created in the collection you are moving the clips into. You can also move clips with drag and drop. If you drop them onto a tab, the clips will be moved into the other collection just like the Move Clips button does. Or, you can move the mouse pointer over a tab while dragging, wait a brief moment for the tab to become active, and then drop the clips in a specific location in that collection.

Combine Clips

To combine multiple clips into a single clip, select them and click the  Merge Clips button. You will be given a choice of delimiters to separate the original clips in the combined clip. Another way is to copy a clip to the clipboard, delete it, and then paste its text inside another clip.

Split Clips

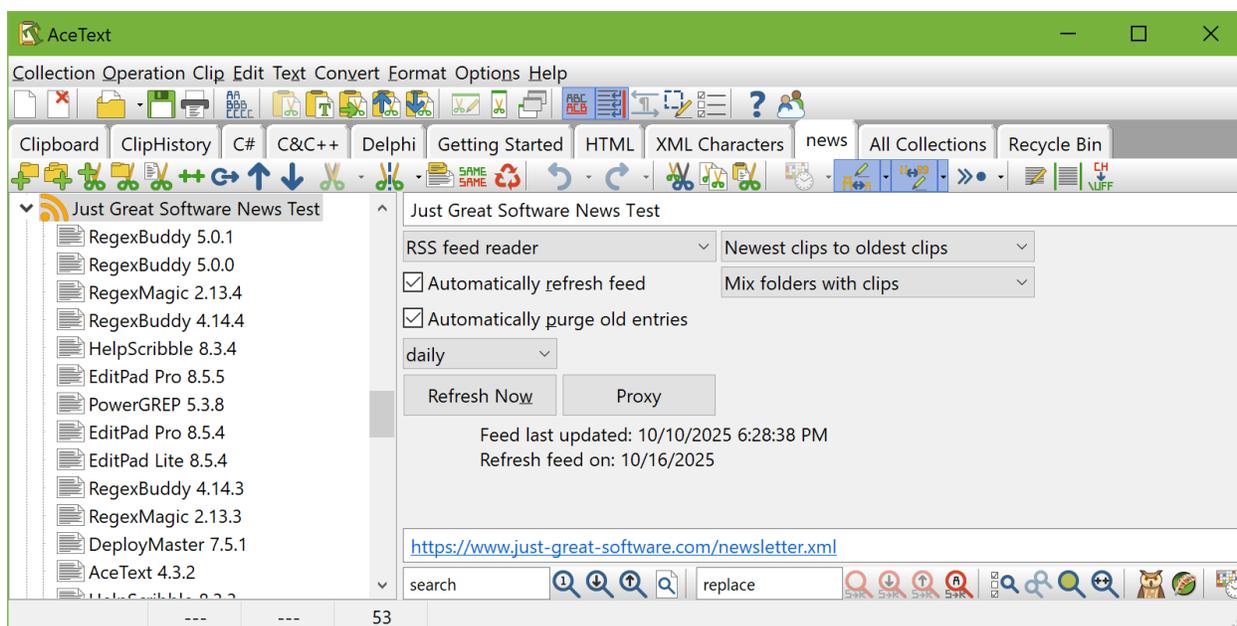
If a clip contains several sections of text that would be useful on their own, you can split those off into new clips. Select part of the clip's text, and select the Move Text into New Clip command. Repeat to split off several parts. Or use the Split Clips button to split clips along specific delimiters or regular expressions.

5. Manage RSS Feeds

RSS feeds are a great way to stay up to date about all sorts of information and news on the Internet, and to keep other people up to date about your company, your web site, or whatever people would like to keep track of. While RSS feeds originally became popular as an easy way to read blogs, today they're used to distribute and gather all kinds of information.

At Just Great Software (developers of AceText), we currently publish an RSS feed at <https://www.just-great-software.com/newsletter.xml> which announces new releases and updates to our products. It also has regular “tips and tricks sections” explaining how to get the most out of our products.

Downloading and Storing RSS Feeds



To retrieve an RSS feed with AceText, click the  New Folder button. Select “RSS feed reader” in the “Kind” drop-down list in the folder’s properties. Several new options will appear. At the bottom of the AceText Editor, enter or paste the URL of the RSS feed, e.g. <https://www.just-great-software.com/newsletter.xml>. AceText can read both RSS and Atom feeds, and supports all revisions of these two feed formats.

To retrieve the feed’s contents, click the Refresh Now button. The button will become disabled, indicating AceText is busy retrieving the feed. You can continue working with AceText while the feed is being retrieved. When AceText has downloaded the feed, it will add all of the feed’s entries as clips inside the RSS reader folder. These clips are just like all other clips, and you can copy or move them into other folders or collections for safekeeping if you like.

If your computer connects to the Internet through a proxy server, click the Proxy button to inform AceText of the details of your proxy server. You only need to do this once. This setting is stored with AceText’s preferences rather than in your clip collection.

When a publisher adds a new item to an RSS feed, the oldest item is usually dropped from the feed. If you turn on “automatically purge old entries” for the RSS reader folder in AceText, AceText will delete all clips in the RSS reader when retrieving an updated feed. This way only entries still present in the feed will be stored. If you turn off “automatically purge old entries”, AceText doesn’t remove any clips from the RSS reader folder. AceText will use the unique ID that RSS feeds assign to their entries to prevent adding duplicate entries.

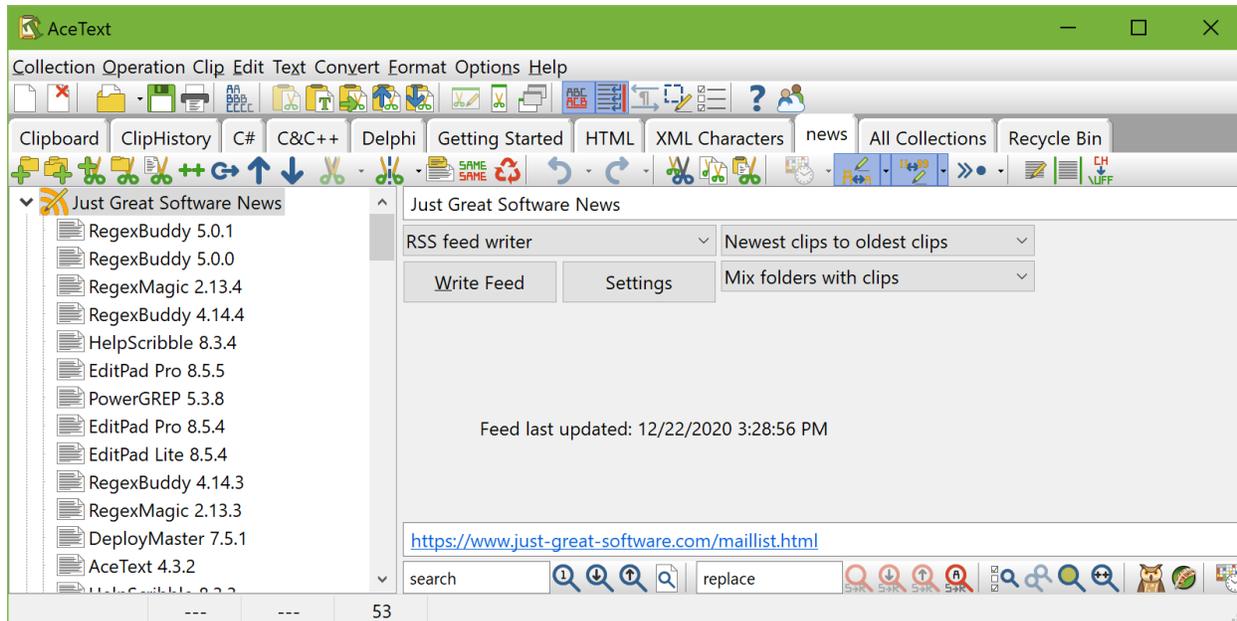
If a feed is updated regularly, you can tell AceText to automatically refresh the feed. If the feed itself doesn’t specify a “time to live”, you can choose how often AceText should refresh the feed yourself, ranging from hourly to daily.

You can create subfolders inside an RSS reader folder. These subfolders will not be affected by refreshing the RSS feed. A single AceText collection can contain as many RSS reader folders as you want.

Reading RSS Feeds

Once you’ve added and refreshed a feed, you can read it right within AceText. Each clip in the RSS reader folder stores the text of one entry. AceText will show you the raw contents, HTML tags and all. Some RSS feeds may contain only summaries rather than complete articles. To read the entry in your web browser, double-click its URL, press Ctrl+E on the keyboard, or select Clip|Browse to URL in the main menu.

Publishing RSS Feeds



Publishing your AceText clips as an RSS feed is a handy way to share clips with other people, even if they don’t use AceText. AceText generates standard RSS 2.0 feeds that can be read by all RSS readers.

To create a feed, first put all the items you want to include in the feed into a single folder in your AceText collection. Then click on the folder. Select “RSS writer” in the “Kind” drop-down list for the folder. Two

new buttons will appear. Click the Settings button to configure your RSS feed. Click the Write Feed button to save or upload your feed as you specified in the settings. Uploading a feed via FTP is done in the background. You can continue using AceText in the mean time. When AceText is done, the Write Feed button will become enabled again, and AceText will indicate the date and time it last successfully uploaded the feed.

You have to put all the clips you want to include in the RSS feed directly under the RSS writer folder. Clips in subfolders will not be included in the feed. RSS feeds do not support folders. AceText does allow you to put subfolders under RSS writers. This way you can create subfolders like “drafts” or “old entries” to store clips that are related to the feed, but which you don’t want to publish.

Writing RSS Feed Entries

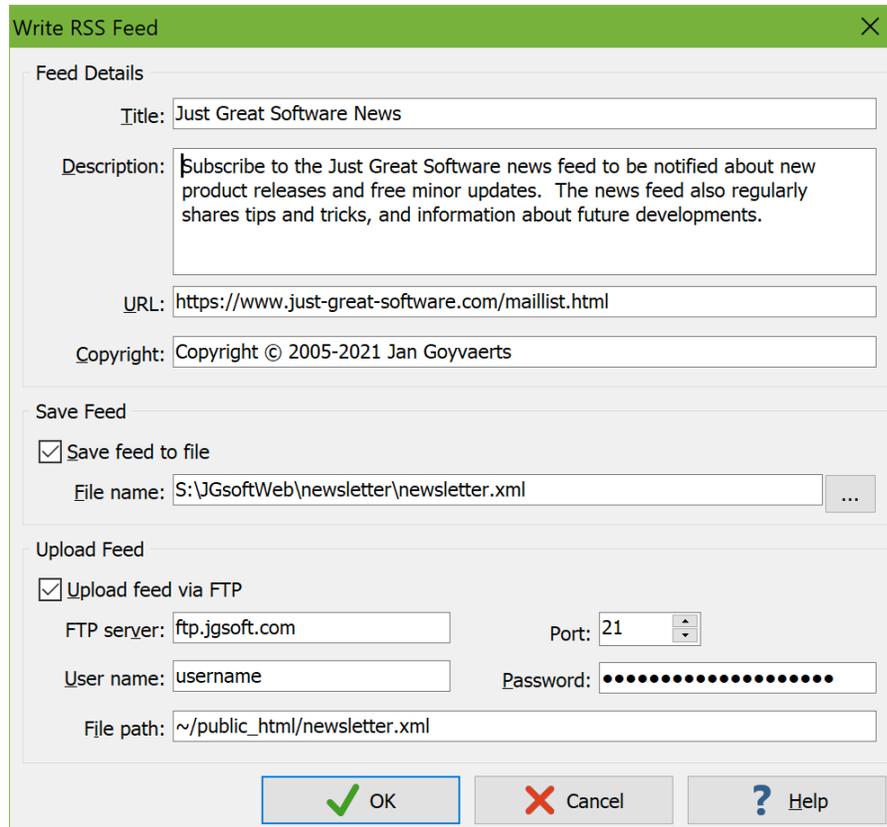
When preparing clips to be published in an RSS feed, set the clip’s URL to a web page with more information about the entry. Some RSS readers, such as the one built into the Firefox web browser, will only display the clip’s label, and open the associated URL when the user clicks on it instead of displaying the clip’s text.

When AceText writes your RSS feed, it assigns a unique ID to each clip. This ID is not visible in AceText, but will be saved in your AceText collection. If you edit the clip and publish your feed again, the clip’s entry in the feed will have the same ID. Feed readers, including AceText, will see the entry as an updated version of the entry in the previous feed, because the ID is still the same, and replace the old entry with the new one. In AceText, this makes no difference when “automatically purge old entries” is on in the RSS feed reader. It does make a difference when the option is off: the old version of the clip will be replaced rather than kept.

What the previous paragraph really means is that if you want to add a new entry to your RSS feed, you should create a new clip, instead of editing an old clip. If you want to base a new entry on an old one, make a copy of the old clip, and edit and publish the copy. The clip’s unique ID is not copied over when you copy clips (otherwise it wouldn’t be a unique ID).

6. RSS Feed Writer Settings

To publish an RSS feed, put some clips into a folder, and set the folder’s “kind” to “RSS feed writer”. Two new buttons will appear. Click the Settings button to configure your feed. A new window will appear.



Feed Details

In this section, you can enter some information to be included in the feed itself. The Title is what feed readers will display to identify your feed. In the Description field you can tell what your feed is about. Not all readers will display the description.

The feed’s URL should point to a web page with more information about the feed, or the feed’s contents formatted in HTML. If you don’t have such a web page, enter the URL where you will publish the RSS feed itself.

The Copyright field is optional. Enter “Copyright © 2021 Your Name” to assert your legal rights.

Save Feed

Turn on the “save feed to file” option and enter a file name if you want AceText to save your feed on your computer. Use this option if you want to upload your RSS feed to your web server by yourself.

Upload Feed

Turn on the “upload feed via FTP” option if you want AceText to upload your feed directly to your web server via FTP. Enter the FTP server’s domain name, and the user name and password you use to log in. As the file path, enter the full path where AceText should save the file on the FTP server.

7. Editing Clips and Text

Often, a clip will need editing before storing it permanently, or before sending it to another application. AceText can perform many common transformations with a single command:

Case Conversion: Convert casing of the text of a clip, or the selected part of the text. You can convert all characters to uppercase or lower case, capitalize the first letter of each word (title case), or invert the casing when you've been typing with Caps Lock on. Some of the syntax coloring schemes that you can select for a clip let you automatically adjust the case to the rules of the syntax coloring scheme.

Quote Conversion: Convert between the straight quotes that you can type on your keyboard, the smart quotes that are used in print, and primes that indicate units such as inches. Some of the syntax coloring schemes that you can select for a clip specify rules for straight and smart quotes so that the conversion doesn't break the syntax of the clip.

Tab Conversion: Convert between tabs and spaces.

Character Conversion: Convert between characters and character escapes or character entities representing those characters.

Reflow Paragraphs: If you want to edit text that uses fixed line lengths, use the Reflow Paragraphs command to remove the line breaks that were used in place of word wrapping. The paragraphs will then rewrap nicely when you edit the text in AceText or send it to an application that supports word wrapping (most Windows applications do). Email messages, newsgroups postings, and text files from old DOS systems or mainframes, often use fixed line lengths.

Limit Line Length: Before sending text to an application that expects fixed line lengths, use the Limit Line Length command. This does the opposite of the Reflow Paragraphs command, inserting line breaks to limit the maximum line length. AceText will only break lines between words, and not in the middle of a word, so some lines will end up shorter than the maximum length you specify. If you want, AceText can compensate for that by inserting additional spaces. When the text is viewed with a fixed pitch font such as Courier, the text will appear to be aligned to both the left and right margins.

Comment Block: Add or remove characters from the start of the line that turn the line into a comment. This is mainly useful for clips containing source code snippets.

Quote Block: In email messages and newsgroup postings, it is common to quote previous messages by prefixing each line of the previous message with a special character, typically >. AceText can automatically add or remove those characters from the start of each line to quote or unquote blocks. After unquoting a block you can use the Reflow Paragraphs command to make the text automatically rewrap when you edit it.

Sort: Sort the lines in a clip alphabetically or alphanumerically.

Trim Whitespace: When selecting text in a document or a web page, it takes quite a bit of effort to make sure no superfluous whitespace characters end up in the selection. When using AceText, you don't have to worry about this. Just capture the text, and use Trim Whitespace to remove extraneous spaces and blank lines at the start and the end of the clip.

Trim Whitespace from Paragraphs: If a clip has superfluous white space at the start and/or end of the lines or paragraphs in the text, use Trim Whitespace from Paragraphs to delete it. E.g. when copying a table from a web page, the resulting text is often needlessly indented by the web browser. Trim Whitespace from Paragraphs takes care of this.

Delete Duplicate Lines: Delete duplicate items from clips containing lists.

Delete Blank Lines: Delete all lines that are totally blank from a clip.

Consolidate Blank Lines: Replace each block of consecutive blank lines with a single blank line in a clip.

Basic Text Editing

The box that displays the text of the active clip is a full-featured edit control. In fact, it is the same edit control that powers EditPad, one of the most flexible and convenient text editors available today (see <https://www.editpadpro.com>).

- Editing Text with The Keyboard
- Editing Text with The Mouse

8. Editing Text with The Keyboard

The box that displays the text of the active clip is a full-featured edit control. In fact, it is the same edit control that powers EditPad, one of the most flexible and convenient text editors available today (see <https://www.editpadpro.com>).

Beyond typing in text, the editor box recognizes the following keyboard shortcuts:

Cursor movement keys

Arrow key	Moves the text cursor (blinking vertical bar).
Ctrl+Left arrow	Moves the text cursor to the start of the previous word or the end of the previous line, whichever is closer.
Ctrl+Right arrow	Moves the text cursor to the start of the next line or the end of the current line, whichever is closer.
Page up/down	Moves the text cursor up or down an entire screen.
Ctrl+Page up/down	Scrolls the text one screen up or down.
Home	Moves the text cursor to the beginning of the line.
Ctrl+Home	Moves the text cursor to the start of the entire text.
End	Moves the text cursor to the end of the line.
Ctrl+End	Moves the text cursor to the end of the entire text.
Shift+Movement key	Moves the text cursor and expand or shrink the selection towards the new text cursor position.
	If there was no selection, one is started.
	Pressing Ctrl as well, will move the text cursor correspondingly.

Clip selection keys

Ctrl+Up/Down arrow	Selects the clip listed above or below the active clip in the collection. This is a fast way to browse through clips and keeps input focus on the editor box.
--------------------	---

Editing commands

Delete	Deletes the current selection if there is one and selections are not persistent. Otherwise, the character to the right of the caret is deleted.
Ctrl+Delete	Deletes the current selection if there is one. Otherwise, the part of the current word to the right of the text cursor is deleted.
Shift+Ctrl+Delete	All the text on the current line to the right of the text cursor is deleted.
Backspace	Deletes the current selection if there. Otherwise, the character to the left of the text cursor is deleted.
Ctrl+Backspace	Deletes the current selection if there. Otherwise, the part of the current word to the left of the text cursor is deleted.

Shift+Ctrl+Backspace	Deletes the current selection if there is one. Otherwise, all the text on the current line to the left of the text cursor is deleted.
Alt+Backspace	Undo
Alt+Shift+Backspace	Redo
Ctrl+Z	Undo
Ctrl+Y	Redo
Insert	Toggles between insert and overwrite mode.
Tab	If there is a selection that spans multiple lines then the selected lines are indented.
	Otherwise, a tab is inserted (possibly replacing a single-line selection if it is not persistent).
Shift+Tab	If there is a selection that spans multiple lines then the selected lines are unindented (outdented).
	Otherwise, if there is a tab (or a series of spaces the size of a tab) to the left of the text cursor then that tab or spaces are deleted.
Ctrl+A	Select All
Ctrl+Alt+Y	Delete Line
Shift+Ctrl+Alt+Y	Duplicate Line

Clipboard commands

Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+V	Paste
Shift+Delete	Cut
Ctrl+Insert	Copy
Shift+Insert	Paste

Characters with Diacritics

You can type many characters with diacritics in AceText even if they don't appear on your keyboard. First, hold down the Ctrl key and press a punctuation key. If your keyboard uses the Shift key to type a particular punctuation character then hold down the Shift key too. Release all keys. Then type in a letter from A to Z, holding down Shift if you want a capital letter.

Punctuation	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
Ctrl+'	á	á	ć	č	é	é	ğ	í	í	ķ	ĺ	ĺ	ń	ń	ó	ó	ř	ř	ś	ś	ú	ú	w	w	ý	ý	z
Ctrl+`	à			è				ì					ò	ò							ù	ù	w	w	ÿ		
Ctrl+:	ä			ë			ĥ	ĩ					ö							ț	ü	ü	w	ÿ	ÿ		
Ctrl+^	â	â	ç	ê	ê		ĝ	ĥ	î	ĵ	ķ	ĺ		ň	ô				ř	ŝ	ť	û	w	ÿ	ÿ	ž	

9. Editing Text with The Mouse

The box that displays the text of the active clip is a full-featured edit control. In fact, it is the same edit control that powers EditPad, one of the most flexible and convenient text editors available today (see <https://www.editpadpro.com>).

Below you can find a list of action you can carry out in the editor using the mouse.

Dragging means to move the mouse before releasing the mouse button you pressed. If you move the mouse pointer to the edge of the editor space while dragging, the text will start to scroll automatically.

Modifier keys like shift or control must be pressed before pressing the mouse button and kept depressed until the mouse button is released.

Left click	Moves the text cursor to the spot where you clicked.
Shift+Left click	Moves the text cursor and expands or shrinks the selection. If there is no selection, the text between the old and new cursor positions becomes selected.
	If you click outside of the selection, the selection plus the text between the selection and the new cursor position becomes selected.
	If you click inside the selection, the new selection is the text between the original start of the selection and the new cursor position.
Left click+drag	When clicking outside the selection, a new selection is created from the point where you press the mouse button until the point where you release it. When clicking inside the selection, the selected text deleted and inserted again at the spot (outside the selection) where you release the mouse button.
Shift+Left click+drag	Expands or shrinks the selection like Shift+Left click, but then the text cursor is moved and the selection adjusted until you release the mouse button.
Rotate wheel	Scrolls the text a single line up or down.
Shift+Wheel keys on the keyboard.	Moves the text cursor a line up or down, like pressing the up or down arrow
Ctrl+Wheel	Scrolls the text an entire screen up or down.
Shift+Ctrl+Wheel	Moves the text cursor a screen up or down, like pressing page up or down on the keyboard.

10. Look up and Reuse Text

When you have captured text into AceText, and arranged it into collections, you can unleash AceText's true power: the ability to instantly locate any piece of text, and transfer it into any Windows application.

Selecting Clips

If your clips have descriptive labels and are neatly organized into folders, you can easily pick the clip you want by clicking on it with the mouse in the tree view. When you have your hands on the keyboard, press Ctrl+Up Arrow or Ctrl+Down Arrow on the keyboard to walk through the clips. Collapsed folders will be expanded.

Depending on the choice you made in the Operation Preferences, double-clicking the selected clip or pressing Ctrl+Enter on the keyboard will place the clip onto the Windows clipboard, ready to be pasted into any application, or directly send it to the application you were last working with.

If the software you use supports inter-application drag and drop, you can click on a clip's label in the list, hold the mouse button down, and move the mouse pointer into the document you are working with. Release the mouse button, and the clip's text will be inserted into the document.

When working with a limited set of clips, this method is quick and easy. E.g. when creating a document that has many identical elements, such as HTML tags in a web page, you can open AceText and your document editor side by side. Just double-click items to send them into the editor software, as you create the document. The AceText Tower is well-suited in this situation, enabling you to see the maximum number of clips with AceText taking up little screen space.

Looking Up Clips With AceType

When you add a clip to a collection, you will notice it has a field labeled "AceType". In this field, you can enter a memorable abbreviation or mnemonic. E.g. you could assign "ty" to a clip with an elaborate "thank you for your business" message, and "pl" for a clip with your company's price list and purchasing options.

To instantly insert a clip into a document, press the AceType hotkey (Win+Alt+A by default) while working with the document in another application. AceText will pop up with an extra AceType field shown above the list of clips. The list of clips will show the clips along with their AceType abbreviations. Clips that do not have abbreviations will be hidden. Type in the abbreviation of the clip you want, and press Enter. E.g. when composing an email response to a customer's inquiry, press Win+Alt+A, T, Y and Enter, to insert your thank you message into the email. Writing polite and elaborate responses to common questions has never been easier.

If you forgot the abbreviation you assigned to the clip you want, type in the first few letters of what you guess to be the abbreviation. The list of clips will reduce to clips with AceType abbreviations starting with those letters. Press Backspace on the keyboard to back up and try again. Of course, all the other methods of locating a clip still work while in AceType mode. And you can still double-click a clip to insert it into your document, email message, etc.

Pressing the AceType hotkey while working with AceText also shows the AceType field above the list of clips. It will let you look up a clip using its AceType abbreviation in the same way. But the clip won't be sent to another application.

Searching And Filtering

Use the search box and buttons at the bottom of the AceText Editor and AceText Tower to search through and filter a collection. Type in a word or phrase, and click the Find Previous or Find Next button to search for a clip containing the word. You can also press F3 and Shift+F3 on the keyboard to search. The label, abbreviation and text of each clip are all searched through. If no clip can be found, nothing will happen when you click one of the search buttons.

When you click the Filter button or press Ctrl+F3, the list will only show those clips that contain the word or phrase you entered in the search box. If you change the text in the search box, the list of clips is automatically updated to show only those clips containing the new search text. Click the Filter button again to turn off the filter and make the list show all clips again.

If you opened several collections, you can search all of them at once by turning on the “all clips in all collections” search option.

Quickly Reuse Clips

AceType, discussed above, is a particularly powerful way to look up and reuse a clip in one go. You do need to assign abbreviations to your clips beforehand. If you did not do so, use AcePaste instead of AceType. The default hotkey is Win+Alt+V. Press it while working with another application, and AceText will pop up. Look up the clip you want, and then double-click it or press Ctrl+Enter on the keyboard. AceText will hide itself, and insert the clip into the document you are editing with the other application.

Reuse Lots Of Clips

If you want to sequentially insert a large number of clips into a document, look up those clips in AceText and activate SequencePaste by clicking the SequencePaste Up or Down button in the toolbar. Then switch to the document you are editing, and press the AcePaste hotkey to paste the clips in the sequence one after the other. This way you do not have to switch back and forth between your application and AceText.

SequencePaste is particularly useful when you need to move several pieces of text from one application to another. First, copy all the items one by one in the first application. AceText will capture them automatically. Then select the items on the ClipHistory Tab and turn on SequencePaste. Now, you can paste all the items one by one in the target application. This saves you having to switch back and forth between both applications, and reorienting yourself each time.

To grab somebody’s contact details off a web page and store them into your address book, first select and copy the person’s name, address, phone number, etc. on the web page, one by one. Then turn on SequencePaste selecting those items. Now, switch to your contact management software, create a new contact, and paste the items one by one pressing Win+Alt+V.

11. Share Experiences and Get Help on The User Forums

When you click the Login button you will be asked for a name and email address. The name you enter is what others see when you post a message to the forum. It is polite to enter your real, full name. The forums are private, friendly and spam-free, so there's no need to hide behind a pseudonym. While you can use an anonymous handle, you'll find that people (other AceText users) are more willing to help you if you let them know who you are. Support staff from Just Great Software will answer technical support questions anyhow.

The email address you enter is used to email you whenever others participate in one of your discussions. The email address is never displayed to anyone, and is never used for anything other than the automatic notifications. AceText's forum system does not have a function to respond privately to a message. If you don't want to receive automatic email notifications, there's no need to enter an email address.

If you select "never email replies", you'll never get any email. If you select "email replies to conversations you start", you'll get an email whenever somebody replies to a conversation that you started. If you select "email replies to conversations that you participate in", you'll get an email whenever somebody replies to a conversation that you started or replied to. The From address on the email notifications is forums@jgsoft.com. You can filter the messages based on this address in your email software.

AceText's forum system uses the standard HTTP protocol which is also used for regular web browsing. If your computer is behind an HTTP proxy, click the Proxy button to configure the proxy connection.

If you prefer to be notified of new messages via an RSS feed instead of email, log in first. After AceText has connected to the forums, you can click the Feeds button to select RSS feeds that you can add to your favorite feed reader.

Various Forums

Below the Login button, there is a list where you can select which particular forum you want to participate in. The "AceText" forum is for discussing anything related to the AceText software itself. This is the place for technical support questions, feature requests and other feedback regarding the functionality and use of AceText.

The "regular expressions" forum is for discussing regular expressions in general. Here you can talk about creating regular expressions for particular tasks, and exchange ideas on how to implement regular expressions with whatever application or programming language you work with.

Searching The Forums

Before starting a new conversation, please check first if there's already a conversation going on about your topic. If you find one, start with reading all the messages in that conversation. If you have any further comments or questions on that conversation, reply to the existing conversation instead of starting a new one. That way, the thread of the conversation stays together, and others can instantly see what you're talking about. It doesn't matter whether the conversation is many years old. If you reply to it, it becomes an active conversation that moves to the top automatically.

In the top right corner of the Forum window, there is a box on the toolbar that you can use to search for messages. When you enter something into that box, only conversations that include at least one message containing the word or phrase you entered are shown.

The filtering happens in real time as you type in your word or phrase. It includes all conversation subjects, all message summaries, and all author names. It also includes the message bodies of conversations that have been downloaded. AceText automatically downloads the 20 most recent conversations when you connect to the forum. Other conversations are downloaded only if you click on them to view them. Downloaded messages are cached between AceText sessions.

This means that the real time filtering does not search through the body text of older messages that you've never viewed. To search through those messages, click the Search Forum button that sits immediately to the left of the search box. AceText then shows all conversations with messages containing the search term in their summaries, author names, or body texts, including messages that haven't been downloaded yet.

You can enter only one search term, which is searched for literally. If you enter "find me", only conversations containing the two words "find me" next to each other and in that order are shown. You cannot use boolean operators like "or" or "and". Since the filtering is instant, you can quickly try various keywords.

If your search term can be found in the subject of a conversation, then all messages in that conversation are always shown. If the search term cannot be found in the subject of a conversation, but it can be found in the summary or body text of a message in that conversation, then the Show Complete Conversations button determines which messages are shown. When this button is up, only the messages in which the search term was found are shown for that conversation. When this button is down, all messages are shown for that conversation.

If you have previously participated in the forums, you can use the Show My Conversations button to show only conversations that you have participated in and conversations that have a message that you gave a +1. If you did not enter a search term, this shows all messages in all those conversations. If you did enter a search term, this reduces the search results to conversations that you participated in.

Conversations and Messages

The left-hand half of the Forum window shows two lists. The one at the top shows conversations. The bottom one shows the messages in the selected conversation. You can change the order of the conversations and messages by clicking on the column headers in the lists. A conversation talks about one specific topic. In other forums, a conversation is sometimes called a thread.

If you want to talk about a topic that doesn't have a conversation yet, click the New button to start a new conversation. A new entry appears in the list of conversations with an edit box. Type in a brief subject for your conversation (up to 100 characters) and press Enter. Please write a clear subject such as "scraping an HTML table in Perl" rather than "need help with HTML" or just "help". A clear subject significantly increases the odds that somebody who knows the answer will actually click on your conversation, read your question and reply. A generic scream for help only gives the impression you're too lazy to type in a clear subject, and most forum users don't like helping lazy people.

After typing in your subject and pressing Enter, the keyboard focus moves to the empty box where you can enter the body text of your message. Please try to be as clear and descriptive as you can. The more information you provide, the more likely you'll get a timely and accurate answer. If your question is about a

particular regular expression, don't forget to attach your regular expression or test data. Use the forum's attachment system rather than copying and pasting stuff into your message text.

If you want to reply to an existing conversation, select the conversation and click the Reply button. It doesn't matter which message in the conversation you selected. Replies are always to the whole conversation rather than to a particular message in a conversation. AceText doesn't thread messages like newsgroup software tends to do. This prevents conversations from veering off-topic. If you want to respond to somebody and bring up a different subject, you can start a new conversation, and mention the new conversation in a short reply to the old one.

When starting a reply, a new entry appears in the list of messages. Type in a summary of your reply (up to 100 characters) and press Enter. Then you can type in the full text of your reply, just like when you start a new conversation. However, doing so is optional. If your reply is very brief, simply leave the message body blank. When you send a reply without any body text, the forum system uses the summary as the body text, and automatically prepends [nt] to your summary. The [nt] is an abbreviation for "no text", meaning the summary is all there is. If you see [nt] on a reply, you don't need to click on it to see the rest of the message. This way you can quickly respond with "Thank you" or "You're welcome" and other brief courtesy messages that are often sadly absent from online communication.

When you're done with your message, click the Send button to publish it. There's no need to hurry clicking the Send button. AceText forever keeps all your messages in progress, even if you close and restart AceText, or refresh the forums. Sometimes it's a good idea to sleep on a reply if the discussion gets a little heated. You can have as many draft conversations and replies as you want. You can read other messages while composing your reply. If you're replying to a long question, you can switch between the message with the question and your reply while you're writing.

Dates and Times

The Started column indicates how long ago each conversation was started. The Last Reply column indicates how long ago the last reply was made, if any. For older conversations, it indicates how much later that reply came after the conversation was started. If you sort conversations by last reply, conversations without replies are sorted by their starting date. The sort order is always based on absolute dates.

The Date Posted column indicates how long ago each message was posted. For replies to older conversations, this date indicates how much later that message was posted after the conversation was started. The Date Edited column indicates how long ago the message was edited, if at all. For older messages, it indicates how much later it was edited after it was posted. If you sort messages by the date they were edited, messages that weren't edited are sorted by their posting date. The sort order is always based on absolute dates.

Directly Attach AceText Clips and Collections

One of the greatest benefits of AceText's built-in forums is that you can directly attach the AceText collections or any other files you're working with. Simply click the Attach button and select the item you want to attach.

To attach a screen shot, press the Print Screen button on the keyboard to capture your whole desktop. Or, press Alt+Print Screen to just capture the active window (e.g. AceText's window). Then switch to the Forum

panel, click the Attach button, and select Clipboard. You can also attach text you copied to the clipboard this way.

It's best to add your attachments while you're still composing your message. The attachments appear with the message, but won't be uploaded until you click the Send button to post your message. If you add an attachment to a message you've written previously, it is uploaded immediately. If you send a message and later notice you forgot an attachment then you can attach it directly. You shouldn't click the Edit button unless you want to edit the body text of the message.

You cannot attach anything to messages written by others. Write your own reply, and attach your data to that.

To check out an attachment uploaded by somebody else, click the Use or Save button. The Use button loads AceText collections directly into AceText. Other files are opened in their associated applications. If you click the Save button, AceText prompts for a location to save the attachment. AceText does not automatically open attachments you save.

AceText automatically compresses attachments in memory before uploading them. So if you want to attach an external file, there's no need to compress it using a zip program first. If you compress the file manually, everybody who wants to open it has to decompress it manually. If you let AceText compress it automatically, decompression is also automatic.

Taking Back Your Words

If you regret anything you wrote, simply delete it. There are three Delete buttons. The one above the list of conversations deletes the whole conversation. You can only delete a conversation if nobody else participated in it. The Delete button above the edit box for the message body deletes that message, if you wrote it. It is labeled Cancel if you have not yet sent your message. The Delete button above the list of attachments deletes the selected attachment, if it belongs to a message that you wrote.

If somebody already downloaded your message before you got around to deleting it, it won't vanish instantly. The message will disappear from their view of the forums the next time they log onto the forums or click Refresh. If you see messages disappear when you refresh your own view of the forums, that means the author of the message deleted it. If you replied to a conversation and the original question disappears, leaving your reply as the only message, you should delete your reply too. Otherwise, your reply will look silly all by itself. When you delete the last reply to a conversation, the conversation itself is also deleted, whether you started it or not.

Changing Your Opinion

If you think you could better phrase something you wrote earlier, select the message and then click the Edit button above the message text. You can then edit the subject and/or body text of the message. Click the Send button to publish the edited message. It will replace the original. If you change your mind about editing the message, click the Cancel button. Make sure to click it only once! When editing a message, the Delete button changes its caption to Cancel and when clicked reverts the message to what it was before you started editing it. If you click Delete a second time (i.e. while the message is no longer being edited), you'll delete the message from the forum.

If other people have already downloaded your message, their view of the message will magically change when they click Refresh or log in again. Since things may get confusing if people respond to your original message before they see the edited message, it's best to restrict your edits to minor errors like spelling mistakes. If you change your opinion, click the Reply button to add a new message to the same conversation.

Updating Your View

When you click the Login button, AceText automatically downloads all new conversations and message summaries. Message bodies are downloaded one conversation at a time as you click on the conversations. Attachments are downloaded individually when you click the Use or Save button.

AceText keeps a cache of conversations and messages that persists when you close AceText. Attachments are cached while AceText is running, and discarded when you close AceText. By caching conversations and messages, AceText improves the responsiveness of the forum while reducing the stress on the forum server.

If you keep AceText running for a long time, AceText does not automatically check for new conversations and messages. To do so, click the Refresh button.

Whenever you click Login or Refresh, all conversations and messages are marked as “read”. They won't have any special indicator in the list of conversations or messages. If the refresh downloads new conversation and message summaries, those are marked as “unread”. This is indicated with the same “people in the cloud” icon as shown next to the Login button.

Forum RSS Feeds

When you're connected to the user forum, you can click the Feeds button to select RSS feeds that you can add to your favorite feed reader. This way, you can follow AceText's discussion forums as part of your regular reading, without having to start AceText. To participate in the discussions, simply click on a link in the RSS feed. All links in AceText's RSS feeds will start AceText and present the forum login screen. After you log in, wait a few moments for AceText to download the latest conversations. AceText will automatically select the conversation or message that the link points to. If AceText was already running and you were already logged onto the forums, the conversation or message that the link points to is selected immediately.

You can choose which conversations should be included in the RSS feed:

- All conversations in all groups: show all conversations in all the discussion groups that you can access in AceText.
- All conversations in the selected group: show the list of conversations that AceText is presently showing in the Forum window.
- All conversations you participated in: show all conversations that you started or replied to in all the discussion groups that you can access in AceText.
- All conversations you started: show all conversations that you started in all the discussion groups that you can access in AceText.
- Only the selected conversation: show only the conversation that you are presently reading in the Forum window in AceText.

In addition, you can choose how the conversations that you want in your RSS feed should be arranged into items or entries in the feed:

- One item per group, with a list of conversations: Entries link to groups as a whole. The entry titles show the names of the groups. The text of each entry shows a list of conversation subjects and dates. You can click the subjects to participate in the conversation in AceText. Choose this option if you prefer to read discussions in AceText itself (with instant access to attachments), and you only want your RSS reader to tell you if there's anything new to be read.
- One item per conversation, without messages: Entries link to conversations. The entry titles show the subjects of the conversations. The entry text shows the date the conversation was started and last replied to. If your feed has conversations from multiple groups, those will be mixed among each other.
- One item per conversation, with a list of messages: Entries link to conversations. The entry titles show the subjects of the conversations. The entry text shows the list of replies with their summaries, author names, and dates. You can click a message summary to read the message in AceText. If your feed has conversations from multiple groups, those will be mixed among each other.
- One item per conversation, with a list of messages: Entries link to conversations. The entry titles show the subjects of the conversations. The entry text shows the list of replies, each with their full text. If your feed has conversations from multiple groups, those will be mixed among each other. This is the best option if you want to read full discussions in your RSS reader.
- One item per message with its full text: Entries link to messages (responses to conversations). The entry titles show the message summary. The entry text shows the full text of the reply, and the conversation subject that you can click on to open the conversation in AceText. If your feed lists multiple conversations, replies to different conversations are mixed among each other. Choose this option if you want to read full discussions in your RSS reader, but your RSS reader does not mark old entries as unread when they are updated in the RSS feed.

Part 2

AceText Reference

1. AceText Tab Pages

The row of tabs in the AceText Editor enables you to switch between different AceText collections. When you open a collection, or create a new one, a new tab appears for that collection. You can rearrange the tabs by dragging them with the mouse. If you open many collections then the tabs may no longer fit on one row. You can scroll the tabs with the double chevron buttons at the right hand edge of the row of tabs. The menu button next to them lets you select the tab you want to activate. On Windows 10 you can also scroll them with the mouse wheel. You can scroll the tabs with the left and right arrow buttons.

You can quickly switch tabs with the keyboard. Press Ctrl+Tab or Shift+Ctrl+Tab to cycle through the tabs. In the Tabs Preferences you can choose whether Ctrl+Tab cycles from left to right or in most recently used order. You can directly access a tab by pressing Alt+1 to activate the leftmost tab, Alt+2 to activate the second tab, Alt+3 for the third, etc. Alt+0 activates the tenth tab.

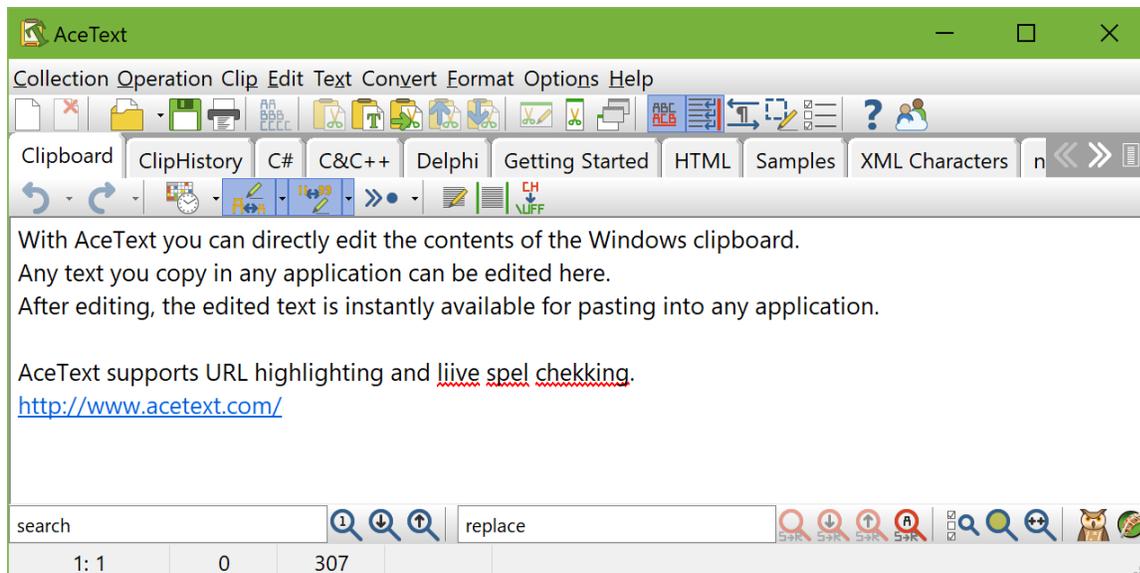
There are four standard tabs, that fulfill special functions in AceText:

- Clipboard
- ClipHistory
- All Collections
- Recycle Bin

All the other tabs are AceText Collection tabs.

You can configure the looks and behavior of the tabs in the Tabs Preferences.

Clipboard Tab



On the Clipboard tab, you can directly edit the contents of the Windows clipboard. When you copy something to the Windows clipboard, AceText automatically updates the text on the Clipboard tab. When you edit the text on the clipboard tab, AceText automatically updates the Windows clipboard.

Use the Clipboard tab when you want to prepare some text to be pasted into another application, and you do not want to keep a copy of that text in AceText for later reuse.

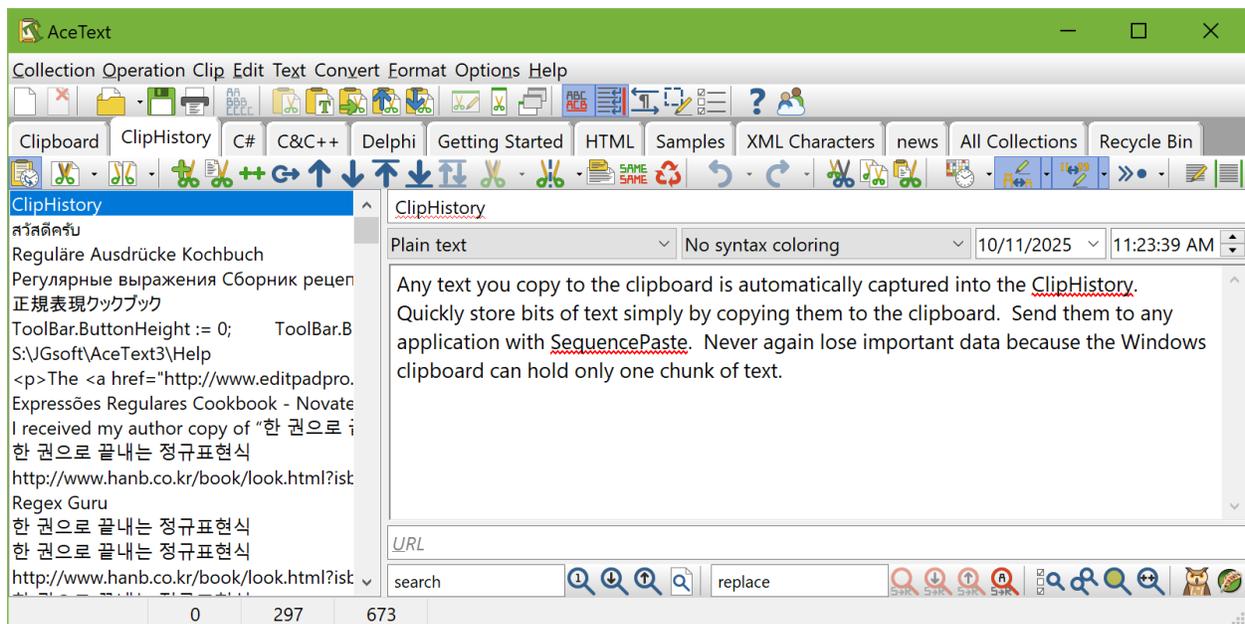
The toolbar on the Clipboard tab page shows the following buttons:

 Undo
 Redo

 Insert Date & Time
 Auto Adjust Case
 Auto Adjust Quotes
 Visualize Spaces

 Insert and Extract Text
 Reformat Text
 Convert Characters

ClipHistory Tab



By default, AceText automatically grabs all text that you copy to the clipboard, regardless of the application you copied it from. Each time you copy some text, a new clip is added to the top of the ClipHistory. You can toggle this on and off via the  Automatic Capture button on the toolbar. If you prefer to disable automatic capture, you can enable manual capture in the Hotkeys Preferences. When using manual capture, you need to press a hotkey combination on the keyboard to make AceText capture the text on the clipboard into the ClipHistory.

The Windows clipboard can only hold one piece of information, which is often insufficient. AceText's ClipHistory, however, can hold as many clips as you want. By automatically grabbing all text you copy to the clipboard, AceText enables you to easily work with many bits of text. If you accidentally overwrite the text on the clipboard by copying something else, you can easily put it back on the clipboard again by double-clicking on the clip in the ClipHistory.

To copy and paste many pieces of text from one application into another, first copy all the items in the first application, without bothering to paste them. Then switch to AceText, select the copied items in the ClipHistory, and activate SequencePaste. Then switch to the target application and paste all the items, one after the other. This way of working is much handier than switching back and forth between the two applications, copying and pasting the items one by one.

Sometimes you can have a mismatch between applications, where one expects you to copy or paste items in a list separately, while the other expects you to copy or paste the list as a whole. You can enable  Auto Append to have the ClipHistory automatically combine separately copied items into one clip. Or you can enable  Auto Split instead to have the ClipHistory automatically split the text that was copied into many smaller clips.

If you forgot to save some text that you worked with (using copy and paste) over the past days or weeks, you can retrieve it from the ClipHistory. See Look up and Reuse Text to learn how to do this.

By default, the ClipHistory holds up to 1,000 items and ignores clipboard data larger than 64 KB. You can change these limits in the Operation Preferences. You can also configure whether the ClipHistory grows at the top or at the bottom and how it should respond when you copy the same text to the clipboard more than once.

The clips captured by the ClipHistory are fully editable. You can change the text the clip holds, as well as the label that is displayed in the list of clips. You can change the order of the clips by drag-and-drop, or with the  Move Up and  Move Down buttons. Use the  Duplicate Clips and  Move Clips buttons to permanently store clips into an AceText collection.

The search box and buttons at the bottom of the ClipHistory tab page enable you to search through and filter the ClipHistory. Enter a search term and click the Find Next or Find Previous button to search for it. You can click the Search Options button to choose which parts of each clip to search through, and whether to search through only the active clip or all clips. If no clip can be found, the button you clicked briefly flashes a “stop” icon.

When you click the Filter button, the list only shows those clips that contain your search term in the clip parts you told AceText to search through. If you change the text in the search box, the list of clips is automatically updated to show only those clips containing the new search text. Click the Filter button again to turn off the filter and make the list show all clips again.

The toolbar on the ClipHistory tab page shows the following buttons:

 Automatic Capture

 Auto Append

 Auto Split

 New Clip

 Create Clip from Text File

- Duplicate Clips
- Move Clips
- Move Up
- Move Down
- Move to Top
- Move to Bottom
- Reverse Clips
- Merge Clips
- Split Clips
- Extract Label
- Delete Duplicate Clips

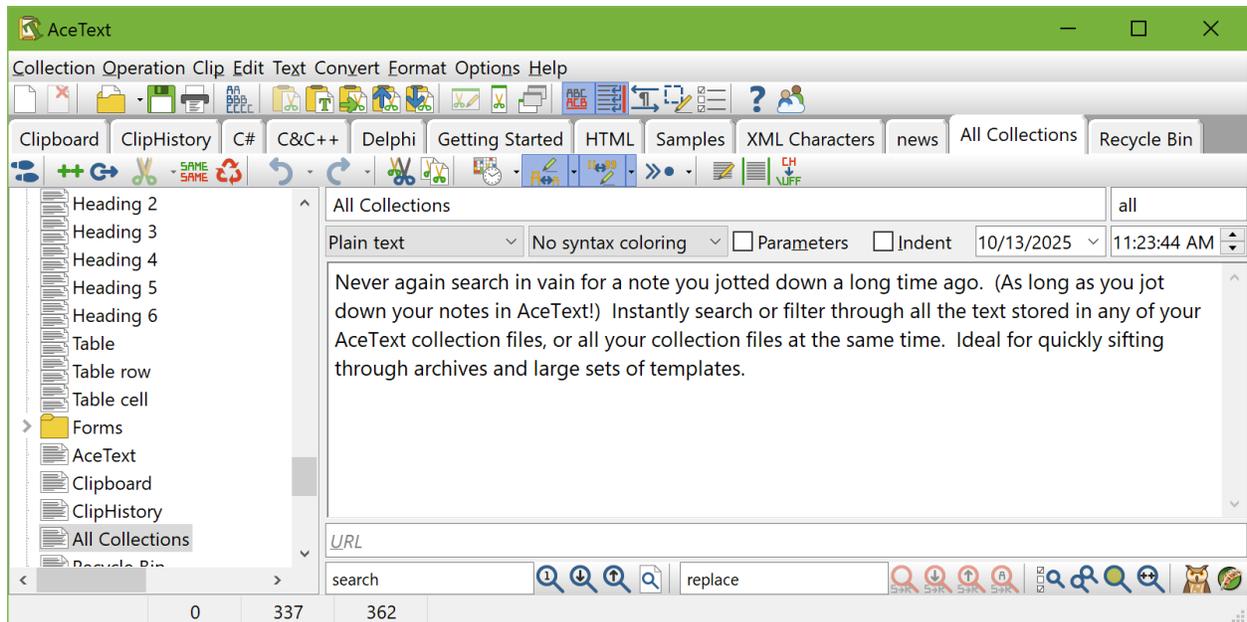
- Undo
- Redo

- Cut Clip to Clipboard
- Copy Clip to Clipboard
- Paste as New Clip

- Insert Date & Time
- Auto Adjust Case
- Auto Adjust Quotes
- Visualize Spaces

- Insert and Extract Text
- Reformat Text
- Convert Characters

All Collections Tab



The All Collections tab lists all the clips in all AceText collections that you have opened. You can use the  Sort Collection button on the main toolbar to determine the order of the clips on the All Collections tab. Note that clips from the ClipHistory and recycle bin are not listed on the All Collections tab. Only clips stored in AceText collections are.

The primary purpose of the All Collections tab is to make it easy to find a clip, even if you do not remember into which collection you stored it. The All Collections tab is also the best place to use AceType. It enables you to store clips in separate collections, for easier editing and updating, yet use AceType with all of the clips, without having to switch between the collections.

Use the AceText Tower if you want to see more clips than the tree view on the All Collections tab page can show in its limited space. See Look up and Reuse Text to learn how to use AceText.

The toolbar on the All Collections tab page shows the following buttons:

 Go To Collection

 Duplicate Clips

 Move Clips

 Merge Clips

 Delete Duplicate Clips

 Undo

 Redo

 Cut Clip to Clipboard

 Copy Clip to Clipboard

 Insert Date & Time

 Auto Adjust Case

 Auto Adjust Quotes

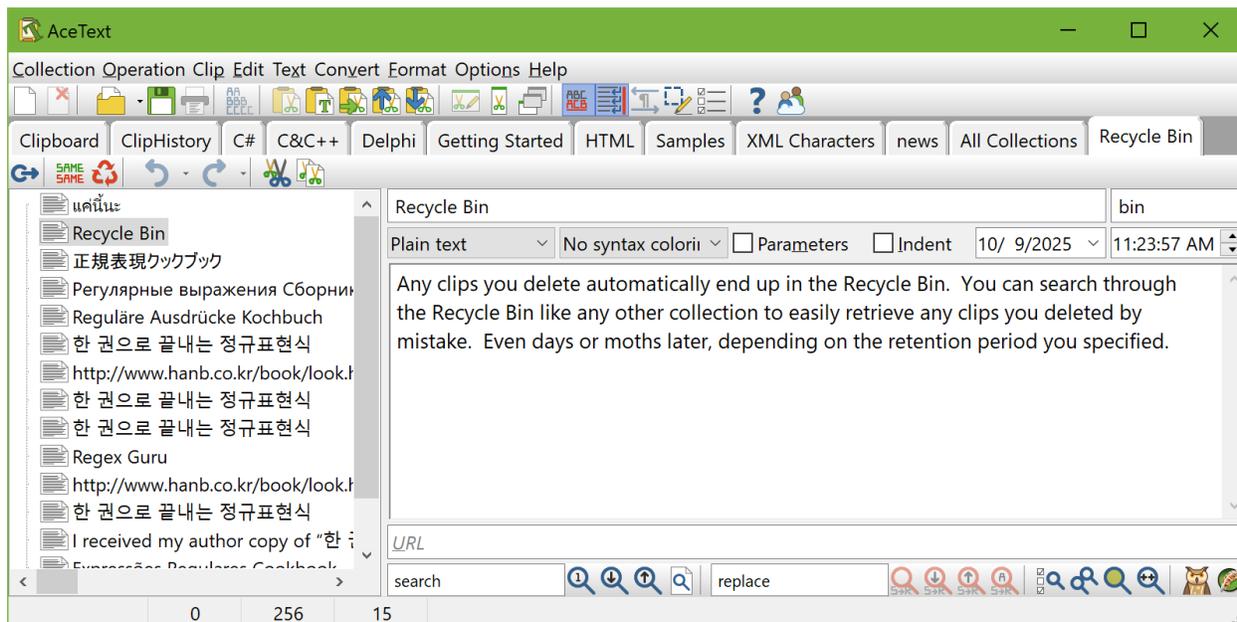
 Visualize Spaces

 Insert and Extract Text

 Reformat Text

 Convert Characters

Recycle Bin Tab



When you delete a clip, it is not really deleted but merely moved into the recycle bin. You can restore the clip from the recycle bin, until it is purged from the recycle bin. You can configure AceText to purge old clips from the recycle bin in the Files Preferences.

To restore a clip from the recycle bin, use the  Move Clips button to move it back into a collection. You can restore the clip into any collection, not just the one you deleted it from.

If you click the  Delete Clips button in the recycle bin, the selected clips are permanently deleted once you close AceText. While AceText is still running, the deleted clips may linger in the undo history. The recycle bin has its own undo history and you can undo deleting clips from the recycle bin like you can undo deleting clips from a collection.

The undo history for each open collection in AceText keeps its own copies of all the clips you edited or deleted. The  Undo and  Redo commands for the ClipHistory and your collections do not depend on the clips being in the recycle bin. But restoring a clip with the Undo command requires you to first undo all later changes you made to the clip's collection. Restoring a clip from the recycle bin doesn't require you to undo anything. A collection's undo history is discarded when you close the collection. The recycle bin can be configured to persist.

The toolbar on the Recycle Bin tab page shows the following buttons:

 Move Clips

 Delete Duplicate Clips

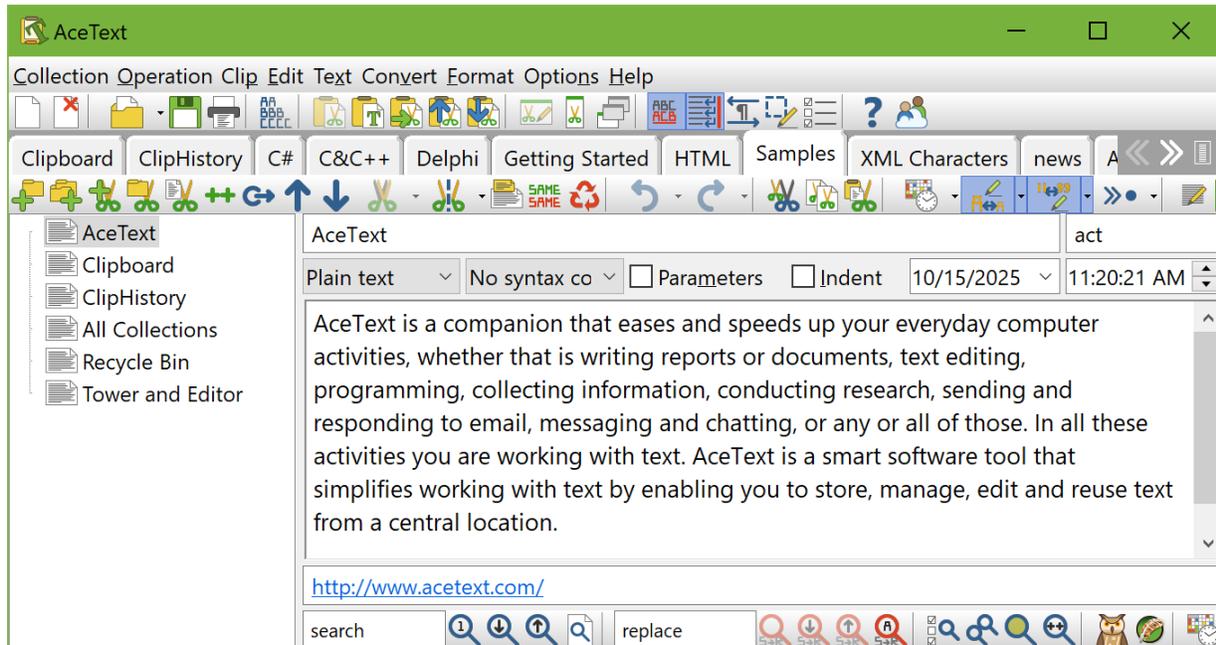
 Delete Clips

 Undo

 Redo

-  Cut Clip to Clipboard
-  Copy Clip to Clipboard

Collection Tabs



When you open an AceText Collection, or create a new one, a new tab appears for that collection.

At the left-hand side, tree view shows the list of folders and clips in the collection. You can use drag and drop to arrange items in the collection, or use the  Move Up and  Move Down buttons or their keyboard shortcuts. To add new items, use the  New Folder,  New Subfolder,  New Clip,  Paste as New Clip or  Create Clip from Text File buttons. You can also drag and drop to transfer text from another application into AceText. Use the AceText Tower if you want to see more clips than the tree view can show in its limited space.

Press the Control+Arrow Up or Control+Arrow Down keys on the keyboard to quickly walk through the list of clips. These hotkeys work even when the tree view does not have keyboard focus.

You can select several items in the tree, with the restriction that all selected items need to be inside the same folder. To select a range of items, click on the first item you want to select, then press and hold the Shift key on the keyboard, and click on the last item in the range. To add a single item to the selection, or to deselect it, press the Control key on the keyboard, and click on the item. If you selected all the items in a collection, the only way to deselect them is to press the Control key while clicking on an item. Then you can click again on that item without pressing the Control key, to select that item and deselect all others.

When you select multiple items, the last one you clicked on is the active item. This is the item that you can edit. When a folder is active, the right hand side of the tab page shows an edit box for the folder's label. When a clip is active, the right hand side allows you to edit the clip.

In the Label box, enter the label that should identify the clip in the list of clips. If you do not enter a label, the first few dozen characters of the clip’s text are used instead. In the AceType box, enter the AceType abbreviation for the clip. Assign a memorable abbreviation to be able to quickly paste the clip using AceType. Abbreviations are case insensitive.

AceText supports several different kinds of clips. Most of the time, you will work with plain text clips. The other kinds are useful in conjunction with AceText-aware applications.

- Plain text: An ordinary piece of text, compatible with all software applications.
- Rectangular text block: A rectangular block of text, or a column selection. AceText-aware applications recognize AceText’s rectangular format. AceText also supports the clipboard formats for rectangular blocks used by the Visual Studio and Embarcadero IDEs. This allows you to copy and paste rectangular blocks between AceText and applications that support these formats. Other applications will treat the clip as several lines of plain text.
- Before and after text: Two pieces of plain text, where the “before” part is intended to be inserted in front of another block of text, and the “after” part behind it.
- Text with emphasis: Text that allows you to change font styles and highlight colors to add emphasis.
- Binary data: Text containing non-printable characters, displayed in hexadecimal format in AceText. AceText-aware applications can paste the bytes directly. Other applications will see a hexadecimal representation of the bytes on the clipboard. Most hex editors will interpret this correctly, allowing you to paste the actual bytes.
- Secret: A secret clip stores plain text. But AceText doesn’t show you the text unless you click the Reveal button. Only collections protected with a password can store secret clips. The actual contents of the clip are plain text. You can send secret clips directly to another application without revealing the secret if the other application can accept keystrokes without revealing them. An input field for a password, for example, fits that bill.

If you change a clip’s kind into “before and after text”, the clip’s text is split at the position of the text cursor in the edit box.

Clips other than “text with emphasis” can use a syntax coloring scheme to automatically highlight the structure of the text in AceText. The highlighting itself does not become part of the clip. It is still only plain text.

Mark the option “indent” if the clip consists of multiple lines, and you want the second and following lines to be lined up with the first line, when the clip is pasted into an AceText-Aware application. When pasting into an application that is not AceText-aware, the “indent” option has no effect. The “indent” option for a clip also turns on automatic indent for the editors to edit the clip’s contents.

The search box and buttons at the bottom of the ClipHistory tab page enable you to search through and filter the ClipHistory. Enter a search term and click the Find Next or Find Previous button to search for it. You can click the Search Options button to choose which parts of each clip to search through, and whether to search through only the active clip or all clips. If no clip can be found, the button you clicked briefly flashes a “stop” icon.

When you click the Filter button, the clip tree only shows those clips that contain your search term in the clip parts you told AceText to search through. If you change the text in the search box, the clip tree is automatically updated to show only those clips containing the new search text. Click the Filter button again to turn off the filter and make the list show all clips again.

The toolbar on an AceText collection tab page shows the following buttons:

-  New Folder
-  New Subfolder
-  New Clip
-  Create Folder from Disk
-  Create Clip from Text File
-  Duplicate Clips
-  Move Clips
-  Move Up
-  Move Down
-  Move to Top
-  Move to Bottom
-  Reverse Clips
-  Merge Clips
-  Split Clips
-  Extract Label
-  Delete Duplicate Clips

-  Undo
-  Redo

-  Cut Clip to Clipboard
-  Copy Clip to Clipboard
-  Paste as New Clip

-  Insert Date & Time
-  Auto Adjust Case
-  Auto Adjust Quotes
-  Visualize Spaces

-  Insert and Extract Text
-  Reformat Text
-  Convert Characters

2. Main Toolbar Reference

You can find AceText's main toolbar at the top of the AceText window. The AceText Tower has a shorter version of this toolbar, with fewer commands. You can choose between small, medium, and large toolbar icons in the Appearance Preferences. From left to right, the main toolbar offers the following commands:

 New Collection
 Close Collection

 Open Collection
 Save Collection
 Print Collection

 Sort Collection

 AcePaste Now
 AceType Lookup
 Double-Click Sends Clip
 SequencePaste Up
 SequencePaste Down

 AceText Editor
 AceText Tower
 Always Visible

 Live Spell Check
 Word Wrap
 Right-to-Left
 Persistent Selections
 Preferences

 Help
 User Forum

New Collection

Click the New AceText Collection button on the main toolbar, press Ctrl+N on the keyboard, or select Collection|New Collection in the menu to start with a fresh, untitled AceText Collection. You can also find this button on the AceText Tower.

See Capture, Enter and Store Text to learn how to add clips to a collection.

Close Collection

Click the Close AceText Collection button on the main toolbar, press Ctrl+W on the keyboard, or select Collection|Close Collection in the menu to close the active collection. If the collection was previously saved

and has unsaved changes, AceText will silently and automatically save those changes. You can turn off the automatic save in the Files Preferences. In that case, AceText will ask if you want the changes to be saved or not.

If the collection is untitled, AceText will ask you if you want to save it. If you do, AceText will ask you for the location and file name.

Open Collection

Click the Open AceText Collection button on the main toolbar, press Ctrl+O on the keyboard, or select Collection|Open Collection in the main menu to open a previously saved AceText collection. You can also find this button on the AceText Tower.

To quickly open a collection you recently worked with, click the downward pointing arrow next to the Open button on the main toolbar. A menu will pop up listing the last 16 collections that you worked with. Click the Maintain List item in the menu to access a list of the last 100 collections. You can rearrange the list and remove collections from the list as well. Select Remove Obsolete to remove collections that no longer exist from the list of recently used collections. Select Remove All to clear the menu.

Save Collection

By default, AceText automatically saves your AceText collections. It saves them not only when you close AceText, but every five minutes. This way, you never lose any data, and you never have to worry about clicking the Save button every so often.

When you click the Save Collection button on the main toolbar, press Ctrl+S on the keyboard, or select Collection|Save Collection in the menu, AceText will ask you for a new file name for the collection. You can either choose the same name, overwriting the existing collection, or a new name. If you pick a new name, the automatic save will work with the new file from then on. The original file is no longer used by AceText, until you decide to open the original again.

Basically, the “Save AceText Collection” button works like the “File|Save As” menu item in many applications that do not automatically save your data.

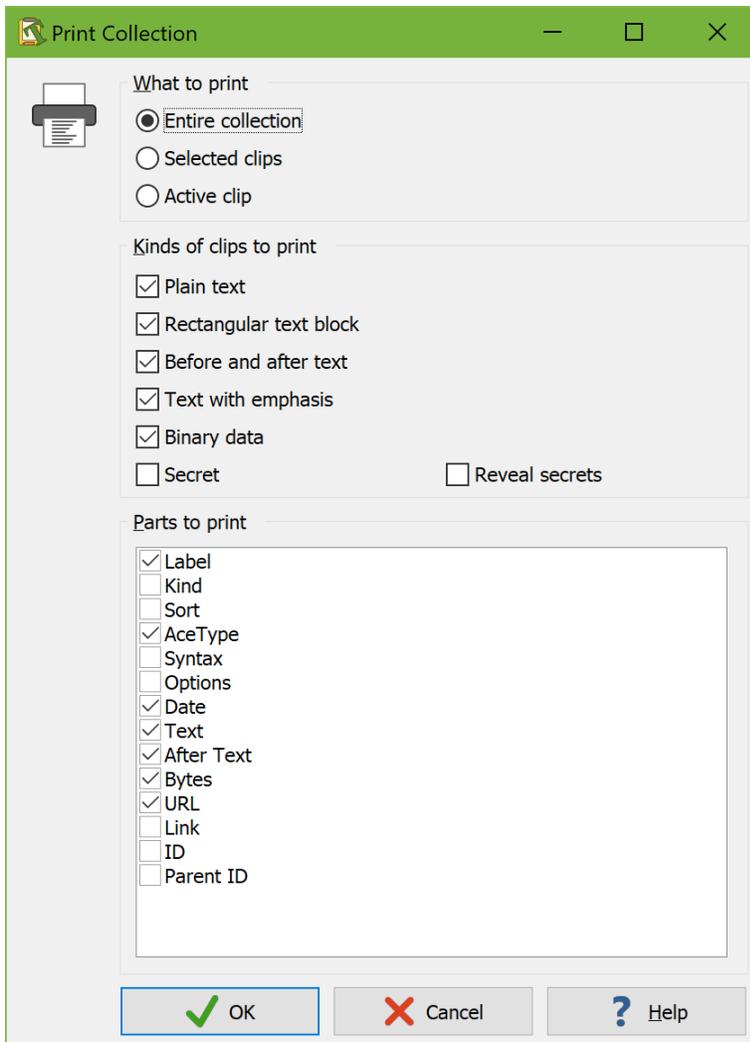
If you do not like the automatic save, you can turn it off in the Files Preferences.

Print Collection

Click the Print Collection button on the main toolbar, press Ctrl+P on the keyboard, or select Collection|Print in the menu to print the entire collection. The first time you do this it shows you the Print Collection dialog where you can choose which kinds of clips and which parts of them should be printed. When you OK the Print Collection dialog you see the Print Preview. The printout starts when you click the Print button in the Print Preview.

When you make a printout, AceText remembers your settings. If you've already printed a collection then the Collection|Print command immediately shows the Print Preview. You can click the What to Print button on the Print Preview to show the Print Collection dialog again to change which kinds of clips and which parts of them to print. If the current collection or selections doesn't contain any clips of the kinds you included in the preview printout, then the Collection|Print command does show the Print Collection dialog before the Print Preview dialog.

What to Print



You can choose to print all the clips on the active collection tab, or the clips you've selected in the clip tree, or just the clip being shown in the AceText Editor. If you want to print all clips in all open collections, switch to the All Collections tab before printing. If you only want to print the body text of the active clip without anything else, use the  Print Text item in the Text menu instead of the Print item in the Collection menu.

Kinds of Clips to Print

You can also choose which kinds of clips to print. If you deselect a certain kind of clip then no parts of clips of that kind are printed at all. If you deselect all clip kinds then only folders are printed. This can be useful to print an outline of a large collection.

For secret clips there is an extra option whether the printout may reveal secrets. If you tick "reveal secrets" and also tick the "text" part then the printout will reveal the text of all the secret clips that you're printing. If you tick the "secret" kind but untick "reveal secrets" then the printout includes all the parts of secret clips except for the "text" part. This can indirectly reveal that your collection contains secret clips as the "text" part will be printed for other kinds of clips but not for secret clips. But the secrets themselves won't be revealed.

Parts to Print

Finally, you can choose which parts of the clips that you want to print. The printout will arrange the parts similarly to how they are arranged in the AceText Editor. The "label", "kind", and "URL" parts apply to both clips and clip folders. The "sort" part only applies to folders. The "AceType", "syntax", "options", "date", "text", "after text", and "bytes" parts are the various clip parts that you can edit in the AceText Editor. The "text" part also covers the "before" text of "before & after text" clips. The "link" part is the link that is generated by the  Edit|Copy Link command. The "ID" part is the item's unique ID used by the Copy Link command. The "parent ID" parts is the ID of the folder that contains the item. Links and IDs are printed for both folders and clips if you select those parts.

Print Preview

With the media player buttons at the top, you can browse through the pages that will be printed. You can also press the Page Up and Page Down keys on the keyboard. If it shows that zero pages will be printed, click the What to Print button to select clip kinds and clip parts to print.

The magnifying glass buttons zoom the preview in and out. You can also press the plus and minus keys on the numeric keypad on your keyboard. Zooming does not affect the printout. It only affects the preview. At low magnification, the preview may not be very accurate. Colored backgrounds may have small gaps and edges of characters may be clipped. That is because the printout is based on the printer resolution. Printer resolution is generally many times higher than the screen resolution. If you zoom out the screen doesn't have enough resolution to accurately show the preview. The actual printout will be pixel perfect regardless.

The margins can be changed by moving the mouse over one of the dotted blue margin lines. The mouse pointer will change to indicate that you can move the margin. A hint box will indicate the size of the margin in centimeters and inches. Click on the line, hold the mouse button down and drag the line until it indicates the desired margin position.

If you want to print with a different font or different text spacing, click the Font & Layout button. This button shows the same text layout configuration screen as the Edit buttons for the text layout configuration in the Appearance Preferences. The difference is that it shows the text layouts you have configured for printing rather than the ones you've configured for editing. Another difference is that if you turn on the "allow bitmapped fonts" checkbox, the font list will include printer fonts rather than screen fonts in addition to the TrueType and OpenType fonts that work everywhere. A "printer font" is a font built into your

printer's hardware. If you select a printer font then you should set "text layout and direction" to "left to right only" for best results.

Press the Setup button to access your printer's setup screen or to select a different printer than the default printer.

"From page" and "to page" determine the range of pages that will be printed. The value of "from page" must be smaller than or equal to that of "to page", even if you want to print in reversed order (see below). "Copies" is the number of copies that should be printed of each page.

Activating "All pages" will print all pages in the range specified by "from page" and "to page", while "odd pages" will print only the odd-numbered (1, 3, 5, ...) pages in the range, and "even pages" only the even-numbered (2, 4, 6, ...) ones. If the page range is only one page ("from page" equals "to page") and that page is odd-numbered while "even pages" is activated, or the other way around, nothing will happen when you click the Print button. You will not get an error message.

If you tick "Reversed" then the pages will be printed from last to first instead of from first to last. This can be useful if you have an inkjet printer that puts pages in reversed order in the out tray.

If you have a monochrome printer or if you do not want to waste expensive color ink cartridges, tick "Black and white" to print everything as black text without any background colors. Bold, italic, underline, and strikethrough styles selected in the color palette will remain. Underline and strikethrough colors will be blank.

You can click the Color Palette button to select a different set of colors. You should select a palette that uses a white background. If you print on colored paper then you should still select a white background in AceText. AceText cannot determine the color of your paper. White is assumed. If the print preview shows a colored or even a black background then the actual printout will also have the same colored or black background. This can waste large amounts of ink. If you click the Print button and the background isn't white then AceText will warn you about this (if you didn't previously tell AceText not to warn about ink usage again).

The Color Palette button shows the same color selection dialog as the corresponding button in the Appearance Preferences. If you edit any palettes, those changes also affect the palettes used for editing. If you want to use different palettes for editing and for printing, create new palettes for printing instead of editing existing ones.

Click the "Print" button to start printing. If you've printed all pages then this also closes the print preview.

Click the "Close" button or press Escape on the keyboard to close the print preview. If you close without printing then AceText does not remember your text layout or color palette choices.

Print Headers

Click the "Headers" button in the Print Preview window to change the headers and footers to be printed, if any.

The headers and footers are printed inside the margins specified on the print preview. If no headers or footers are specified, their space will be claimed by the text body.

A few special placeholders are available in the headers and footers:

%P	Page number of the current page.
%N	Total number of pages.
%D	Current date, printed using the short date format specified in the regional settings in Windows.
%T	Current time, printed using the short time format specified in the regional settings in Windows.
%FD	Date the collection was last modified, again printed in short date format.
%FT	Time of the day the collection was last modified, again printed in short time format.
%FP	Full path to the collection being printed: folder + file name.
%FN	File name (without folder name) of the collection being printed.

Sort Collection

Click the Sort button on the main toolbar or select Sort in the Collection menu to change the order of the clips in the collection.

AceText can sort clips in various ways. The “A to Z” options sort alphabetically, character by character. “A10” sorts before “A4” because 1 sorts before 4. The “A to Z and 0 to 9” options sort alphanumerically. They sort numbers logically. When the text contains numbers that consist of multiple digits, the numbers are compared as a whole. So “A4” sorts before “A10” because 4 is less than 10. This sort order is also known as “natural sort”.

When comparing folders, alphabetic and alphanumeric sort always use the folder label. When comparing folders with clips, they use the folder label or clip label. If a clip doesn’t have a label, then the first 250 characters of the clip’s contents are used. When comparing clips, you can choose whether to compare the label or the AceType abbreviation. If you select to compare the AceType abbreviation and two clips in the same folder don’t have an abbreviation or they have the same abbreviation, then those two clips are compared by their labels instead.

You can also choose to sort clips by their dates. For clips, this compares the date that you can set in the AceText editor. For folders, sorting from older to newer compares the dates on the oldest clip in each of the folders. Sorting folders from newer to older compares the dates on the newest clip in each of the folders.

When the collection is sorted, you can choose whether folders should be mixed with clips. If you do, folders are compared with clips as described above. If a folder contains clips and subfolders, then subfolders may appear before, between, and after clips, depending on how they compare. Alternatively, you can choose to sort all the subfolders before or after all the clips in a folder. Then the “sort clips” option is only applied to clips. You can then make a separate “sort folders” choice that determines how folders are compared with each other.

You can choose to apply the sort order to the entire collection. The Sort Collection button is the only way to apply a sort order to an entire collection. If you turn this off then the sort order is only applied to the selected folders. You can change the sort order of a single folder directly by using the drop-down lists in the AceText Editor. This way you can sort one folder differently from the rest of the collection (or not at all).

Tick “keep clips and folders sorted in this order when they are edited” to save your chosen sort order in the collection or in the selected folders. Untick this if you want the sort order to be applied just once. If you

previously ticked this option and then change the sort order again without this option, then the previously saved sort order is removed.

Tick “set all subfolders to sort like their parent folders” if you want to remove any sorting options from the folders in your collection. This ensures that the entire collection is sorted in the same way.

AcePaste Now

After activating  AcePaste,  AceType or  AceEdit, you can click the AcePaste Now button on the main toolbar to send the selected clip to the application you were working with when you activated AcePaste, AceType or AceEdit. You can achieve the same by double-clicking the clip, pressing Ctrl+Enter on the keyboard, or invoking the AcePaste/AceType/AceEdit keyboard shortcut again.

When the  Double-Click Sends Clip mode is active, you can also click the AcePaste Now button to send the active clip to the previously active application.

AceType Lookup

You can click the AceType Lookup button on the main toolbar or select AceType Lookup in the Operation menu to search for a clip using its AceType abbreviation.

A highlighted AceType field appears above the list with clips in the collection. Enter the AceType abbreviation of the clip you want to find. As you enter the abbreviation, the list of clips in the collection is filtered. It will only show clips that have an abbreviation starting with the same characters as the ones you typed. As soon as you activate AceType Lookup, the labels of the clips in the list are prefixed with their abbreviations. Clips that do not have an abbreviation are hidden. If you forgot the exact abbreviation of a clip, you can easily look it up. While typing the abbreviation, you can press Ctrl+Arrow Up or Ctrl+Arrow Down on the keyboard to select one of the clips in the list, without typing in (the remainder of) its abbreviation. If none of the clips have an abbreviation starting with the characters you entered then AceText ignores as many characters at the end of the abbreviation you’re looking up as needed to be able to show at least one clip that starts with the remaining characters.

Press the Enter key on the keyboard to confirm the selection. This hides the AceType Lookup field. The clip you just found remains active.

If you want to look up a clip using its AceType abbreviation and immediately paste it into another application, then press the  AceType hotkey on the keyboard while working with the other application. You can configure it in the Hotkeys Preferences. If you like to use the mouse, right-click on the AceText tray icon and select AceType from the menu. This starts the same AceType lookup process. But when you press Enter, the clip is immediately sent to the application that was active when you pressed the hotkey or right-clicked the tray icon.

Double-Click Sends Clip

Click the Double-Click Sends Clip button on the main toolbar or select Double-Click Sends Clip in the Operation menu to toggle this mode on or off. When this mode is on, double-clicking a clip in the clip tree

directly pastes it into the last application that was active before you switched to AceText. With this mode, pressing Ctrl+Enter or clicking the  AcePaste Now button also sends the active clip to the last active application.

SequencePaste Up And Down

Click the SequencePaste Up or Down button on the main toolbar, or press Shift+F5 (up) or F5 (down) on the keyboard, to activate SequencePaste mode. You can also find these buttons on the AceText Tower. The mode is indicated by the SequencePaste button on the toolbar, which remains depressed. The AceText icon next to the system clock also indicates SequencePaste mode.

In SequencePaste mode, AcePaste works differently. Rather than popping up AceText for you to select the clip, AcePaste automatically sends the active clip to the application you are working with. The active clip is the one being displayed in AceText's window (regardless whether the window is actually visible). After the clip is sent to the application, either the clip above it in the collection, or the one below it is made active, depending on the direction of SequencePaste.

If you press the AcePaste hotkey while SequencePaste is not active, you can activate SequencePaste while AceText is waiting for you to select the clip to be pasted. Doing so immediately pastes the selected clip, and activate the next clip in the sequence.

If only a single clip is selected in the list, SequencePaste cycles through all the clips in the collection. When the end is reached, AceText activates the first clip again. If two or more clips are selected, SequencePaste cycles through the selected clips only.

While SequencePaste is active, you can make a different selection to change which clips SequencePaste cycles through, without having to deactivate and reactivate SequencePaste.

To deactivate SequencePaste, click the SequencePaste Up or Down button again. If you click the button that is not the one you activated SequencePaste with, the direction of SequencePaste is changed.

SequencePaste makes it easy to paste many pieces of text in quick succession. E.g. when filling out an order form, you usually have to enter your name, address, etc. in the same order. If you prepare an AceText collection with those items, you can select them and activate SequencePaste to quickly paste each item in the corresponding field on the order form.

On the ClipHistory tab, SequencePaste has an extra feature: if you copy one or more items to the clipboard while SequencePaste is active, AceText selects those items as the new sequence to be pasted. This way you can copy and paste series of items between two applications, without switching to AceText. With SequencePaste active, switch to the first application, copy a bunch of items, switch to the second application, and paste those items. Switch back to the first application to copy some more items. If you then start pasting again in the second application, only the second batch of items is pasted.

In the Operation Preferences you can choose whether the ClipHistory adds newly captured clips at the top or at the bottom. The default is at the top. In this case, if you use SequencePaste Up then the items are pasted in the order that you copied them. If you use SequencePaste Down then the items are pasted in reverse order. If you change the preference to add new clips to the bottom of the ClipHistory, then SequencePaste Down pastes items in the order you copy them, while SequencePaste Up pastes them in reverse order.

AceText Editor

If you clicked the AceText Tower button on the main toolbar to show the AceText Tower, you can hide the AceText Editor by clicking the AceText Editor button in the main toolbar. Click the AceText Editor button again to show the AceText Editor window again. Another way to toggle the visibility of the AceText Editor is to press the F8 key on the keyboard. If the editor is already visible, press the F9 key to move the keyboard focus to the editor.

The AceText Editor is AceText's main window. When you start AceText the very first time, only the AceText Editor appears.

The AceText Editor shows the main toolbar, a row of tabs, the active tab page, and a status bar. When the AceText Tower is hidden, as it is by default, all tab pages except the Clipboard tab page, show a list of clips at the left hand side, and the contents of the clip at the right hand side. When the AceText Tower is visible, the AceText Editor only shows the contents of the clip.

AceText Tower

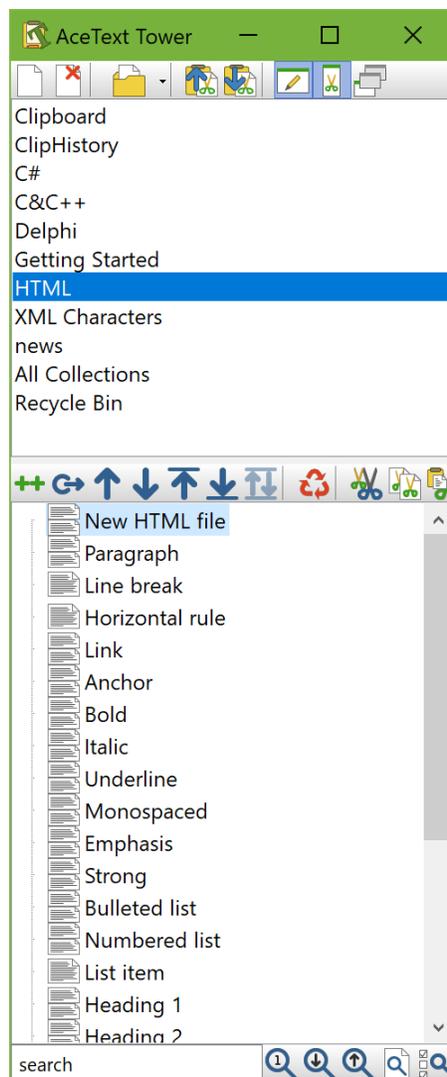
Click the AceText Tower button in the main toolbar to show the AceText Tower. Click that button again to hide the AceText Tower. Clicking the X button on the AceText Tower's caption bar closes both the AceText Tower and the AceText Editor.

You can also toggle the tower's visibility by pressing Ctrl+F8 on the keyboard. If the tower is already visible, press Ctrl+F9 to move keyboard focus to the tower.

When the AceText Tower is visible, the list of clips that is normally shown at the left hand side of the AceText Editor is then shown by the AceText Tower instead. The advantage is that you can make the AceText Tower very tall, so you can see a large number of clips in the list, without having to make the AceText Editor large as well. This way you could dock the AceText Tower to the left hand side of your computer's monitor, and the AceText Editor to the bottom. This maximizes the amount of information AceText shows you, without covering the entire screen.

When reusing clips without editing them, show the AceText Tower and then hide the AceText Editor by clicking the AceText Editor button on the main toolbar. You can then position the AceText Tower and the application you are working with side by side. This way, you can see the maximum number of clips while AceText still takes up relatively little screen space.

At the top, the AceText Tower shows an abbreviated main toolbar. Below it, a list shows the names of the available collections. Click on a name in the list to switch to that collection. If the list is too short, move the splitter just below the list (above the second toolbar) to make it taller. The second toolbar is a shorter version of the



collection toolbar, with the following commands. If you use the AceText Editor and AceText Tower at the same time, then the buttons for duplicating, moving, and deleting clips are only available on the AceText Tower.

Edit Clip

-  Duplicate Clips
-  Move Clips
-  Move Up
-  Move Down
-  Move to Top
-  Move to Bottom
-  Delete Clips

 Copy to Clipboard

 Paste as New Clip

Use the drop-down list and buttons at the bottom to search through and filter the collection, just like you can do with the Search box and buttons on the Collection Tabs.

Focus Editor

Press F9 on the keyboard to instantly switch keyboard focus to the AceText Editor. If the AceText editor is not visible, pressing F9 will make it visible. Press F8 if you want to hide the editor.

If a clip is selected when you focus the editor then the edit box to edit the clip's contents receives keyboard focus. When a folder is selected, the edit field for the folder's label receives keyboard focus. After pressing F9, you can instantly edit the text of the clip or the label of the folder.

Focus Clip Tree

Press Ctrl+F9 on the keyboard to switch keyboard focus to the tree or list with the clips in the active collection. You can then navigate through the list with the keyboard. Press the arrow keys to move around the tree. Press the + (plus) and - (minus) keys to expand or collapse folders. Press Home or End to go to the first or last item in the collection. Keep the Shift key depressed while pressing the arrow keys to select multiple clips.

Pressing Ctrl+F9 does not change the visibility of the AceText Tower at any time. If the AceText Tower is hidden then the tree is visible inside the AceText Editor. Ctrl+F9 focuses the tree wherever it is visible. Press Ctrl+F8 to show the AceText Tower.

Note that you can walk through the list of clips even when the tree does not have keyboard focus. Simply press Ctrl+Shift+Arrow Up or Ctrl+Shift+Arrow Down. One key difference is that the Ctrl key method always walks through clips inside folders. Using the arrow keys without Ctrl when the tree has input focus, on the other hand, only selects clips inside folders when those folders have been expanded.

 **Always Visible**

Click the Always Visible button on the main toolbar, or turn on Options | Always Visible in the menu, to keep AceText permanently visible, on top of all other applications. This is handy if you want to transfer several pieces of text between AceText and another application, when the other application fills the entire screen. That way, AceText remains visible even when you switch to the other application, so you can quickly switch back and forth. Click the Always Visible button again to turn off the feature.

 **Live Spell Check**

Click the Live Spell Check button on the main toolbar, press F12 on the keyboard, or turn on Options | Live Spell Check in the menu to make AceText mark misspelled words while you type. Click the button or press F12 again to stop marking misspelled words.

You need to have downloaded at least one spell check dictionary for the live spell check to work. If you have not done so yet then clicking the Live Spell Check button offers to download one or more dictionaries. You can download dictionaries at any time through the Appearance Preferences.

When you double-click a word that is marked as misspelled, you get a list of correctly spelled words that are similar to the word you typed. Double-click on the correct alternative in the list to replace the misspelled word. If the word you typed is correctly spelled, click the Learn button to permanently add it to the list of correctly spelled words. Or, click the Always Ignore button to accept the word as correctly spelled until you close AceText.

 **Word Wrap**

Click the Word Wrap button on the main toolbar, press F11 on the keyboard, or select Word Wrap in the Options Menu to toggle word wrapping on or off. When you turn it on, paragraphs that do not fit on a single line will span across multiple lines.

Word wrapping only affects how AceText displays the text of a clip. No line breaks are inserted into the text. If you resize AceText, all paragraphs are re-wrapped to fit the new width. When you paste the clip into another application, it is up to that application to apply its own word wrapping, or not. If you want to permanently limit the length of the lines in a clip, click on the Reformat Text button on the collection toolbar, and select the Limit Line Length item.

Turn on word wrapping when working with text documents. It makes paragraphs easy to read. Turn off word wrapping when working with line-based data.

When working with text like source code where indentation is important, you can turn on  Indent Wrapped Lines in the Options menu to maintain indentation when word wrap is on.

Right-to-Left

Select Right-to-Left in the Options menu to quickly toggle text clips between left-to-right and right-to-left editing. What this command really does is switch the multi-line editor controls on the AceText Editor between the “default text layout for clip contents” and the “layout for opposite text direction” that you selected on the Appearance tab in the Preferences. The single-line editor control for the clip’s label is similarly switched between the “default text layout for clip labels” and its corresponding “layout for opposite text direction”. Because you’re switching between text layouts, the switch may change more settings than just the text direction. Other controls on the AceText Editor, including the URL field, do not switch text directions.

If you have at least one right-to-left keyboard layout installed in Windows then this command has a button on the main toolbar. You can then also switch text direction with the Ctrl and Shift keys on the keyboard. To switch to left-to-right, hold down either Ctrl key while pressing and releasing the left hand Shift key. To switch to right-to-left, hold down either Ctrl key while pressing the right hand Shift key. These key combinations too switch the editor controls between the “default text layout for clip contents/labels” and the “layout for opposite text direction”.

If you want right-to-left text to always be displayed legibly then both the default and opposite text layouts should be “complex script” text layouts, one right-to-left and the other left-to-right. Both support bidirectional text editing. The only difference is whether the text is left-aligned or right-aligned. The other left-to-right text layouts, however, do not support bidirectional text editing at all.

Persistent Selections

By default, selections in AceText behave like in most other Windows applications. When you move the text cursor, by pressing a key on the keyboard or clicking somewhere with the mouse, the selection disappears. When you enter some text or paste from the clipboard, the selected text is replaced by the text you entered or pasted. Pressing the Backspace or Delete key deletes the selection.

You can turn on Persistent Selections in the Options menu or via the corresponding button on the main toolbar. If you do so then moving the text cursor does not cause the selection to disappear. Also, entering or pasting text does not delete the selected text. The typed or pasted text is inserted at the position of the text cursor, whether that is inside or outside of the selection. If it is inside the selection, the selection is expanded to include both the text originally selected and the newly entered text.

Pressing the Backspace or Delete key does not delete a persistent selection. It only deletes one character to the left or right of the cursor. Press Ctrl+Delete to delete a persistent selection.

Rectangular Selections

If you use the  Word Wrap button to turn off word wrap and select a monospaced left-to-right text layout as the default text layout in the Appearance Preferences then you can make rectangular selections within the body text of your clips. These settings ensure that the characters in your text line up neatly in columns. The clip itself does not need to be of the “rectangular text block” kind. It will work with any clip kind other than “binary data”. With word wrap off and a monospaced text layout, you can make a rectangular selection by holding down the Alt key while making a selection like you normally do with the keyboard or the mouse.

Making a rectangular selection automatically puts AceText into column editing mode while the cursor is at the left- or right-hand edge of the selection. Column editing often results in a new rectangular selection that is zero columns wide. Such a zero-column selection appears as a vertical line between two columns. You can think of it as a multi-line text cursor. You can make such a selection simply by creating a rectangular selection that starts and ends on the same column on different lines.

Column editing works differently depending on whether selections are persistent.

Column Editing with Non-Persistent Selections

When selections are not persistent, the cursor is always at an edge of the selection. Moving the cursor away from the selection removes the selection. So column editing is always active if there is a non-persistent rectangular selection.

Entering a character on the keyboard replaces the selected columns with that character on each line. You get a new zero-column selection after the entered character on each line. Entering a character on a zero-column selection inserts the character at the selected column on each line and shifts the zero-column selection one column to the right. Combined, entering multiple characters replaces the selected columns with those characters on each line.

When one or more columns are selected, pressing Backspace or Delete on the keyboard deletes the selected columns. It doesn't matter whether the cursor is at the left- or right-hand edge. The selection itself remains where it is but becomes zero columns wide.

Pressing Backspace on a zero-column selection deletes the character to the left of the selection on each selected line. The selection moves one column to the left along with the cursor. Pressing Delete on a zero-column selection deletes the character to the right of the selection on each selected line. The selection stays where it is.

Pasting text that does not contain any line breaks replaces the selected columns with the pasted text on each line. You get a new zero-column selection after the pasted text on each line. Pasting text without line breaks on a zero-column selection inserts the pasted text on each line and shifts the zero-column selection as many columns to the right as were pasted.

Copying a rectangular selection and pasting it over another rectangular selection deletes all the selected columns. The pasted rectangle is then inserted at what was the left-hand edge of the deleted selection. The first line on the clipboard is inserted into the first selected line, the second line on the clipboard into the second selected line, and so on. Nothing is selected after pasting. If there are fewer lines on the clipboard than were selected then the whole selection is still deleted. Nothing is inserted on the extra lines that were selected. If there are more lines on the clipboard than were selected, those lines are inserted at the same column into the lines below the selection. If there are not enough lines in the file below the selection then as many lines are added as needed. They are padded with spaces until the column that you are pasting at.

Copying a linear selection and pasting it over a rectangular selection deletes all the selected columns. All of the pasted text, including its line breaks, is then inserted at what was the upper left corner of the deleted selection. Nothing is inserted on the second and following lines that columns were deleted from.

So, by turning off persistent selections, you can easily replace or fill a rectangular selection with new text.

Column Editing with Persistent Selections

When selections are persistent, moving the cursor away from the edge of the selection cancels column editing without removing the selection. Column editing becomes active again as soon as you move the cursor back to the edge of the selection. Any edits you make while the cursor is not at the selection's edge are made only at the position of the cursor, whether that is inside or outside the selection.

Entering a character on the keyboard inserts that character on each selected line at the column that the text cursor is on. You get a new zero-column selection after the entered character on each line. Nothing is deleted.

Pressing Backspace deletes the character to the left of the cursor's column on each selected line. Pressing Delete deletes the character to the right of the cursor's column on each selected line. If the deleted column was selected, the selection shrinks by one column. If the deleted column was to the left of the selection, the selection shifts one column to the left. Regardless, the cursor stays at the same edge of the selection.

Pasting text that does not contain any line breaks inserts the pasted text at the cursor's column on each selected line. You get a new rectangular selection of the pasted text on each line it was pasted into.

Pasting text that does contain line breaks clears and ignores any persistent selection and is thus unaffected by column editing. Pasting a linear block simply inserts all its text at the cursor's position.

Pasting a rectangular block inserts the first pasted line at the cursor, the second line at the cursor's column on the line below the cursor, and so on. If the cursor was at the bottom of a rectangular selection then the second and following lines are pasted below the selection. If there are not enough lines in the file below the cursor to paste all the lines on the clipboard then additional lines are added to the file. They are padded with spaces until the cursor's column.

So, by turning on persistent selections, you can easily prefix or suffix a rectangular selection with new text.

»• Visualize Spaces

Turn on Visualize Spaces in the Options menu or click the corresponding button on the collection toolbar if you want spaces and tabs to be visualized. Spaces are indicated by a vertically centered dot. Tabs are indicated by a » character. This option is useful when working with clips where extraneous whitespace can lead to problems, or where the difference between tabs and spaces matters.

To change the color of spaces and tabs, customize the color palette and change the “Editor: Whitespace” color. Its text color is only applied to visualized spaces and tabs. But its background color is applied to spaces and tabs whether they are visualized or not. So if you set a background color other than “default” for the “Editor: Whitespace” color then spaces and tabs appear as colored rectangles, with or without their symbol.

The Visualize Spaces button on the collection toolbar has a drop-down menu with several related items:

- Convert|Tabs|Tabs ⇒ Spaces
- Convert|Tabs|Spaces ⇒ Tabs
- Convert|Tabs|Indentation Spaces ⇒ Tabs
- »• Visualize Spaces
- ¶ Visualize Line Breaks

Selecting the Visualize Spaces item in the drop-down menu has the same effect as clicking the Visualize Spaces button directly.

¶ Visualize Line Breaks

Turn on Visualize Line Breaks in the Options menu or select the item in the drop-down menu of the »• Visualize Spaces button on the collection toolbar if you always want to see a ¶ character (Pilcrow symbol) at the end of each paragraph, indicating the invisible line break character(s) at the end of each line. To change the color of the ¶ character, customize the color palette and change the “Editor: Line breaks” color.

If you turn off Options|Visualize Line Breaks then line break symbols are only displayed when line break characters are selected. They are then shown with the colors for selected text in reverse.

3. Collection Toolbar Reference

You can find the collection toolbar below the row of tabs on the AceText Editor window. If you have a high resolution monitor and the toolbar buttons are too small, turn on the “large toolbar icons” option in the Appearance Preferences. The buttons available on the collection toolbar depend on the active tab. Buttons that do not apply to the active tab are invisible. If you use the AceText Editor and the AceText Tower at the same time, then the buttons for duplicating, moving, and deleting clips are only available on the AceText Tower. The order of the buttons is always the same. From left to right, the collection toolbar offers the following commands:

 Go To Collection

 Automatic Capture

 Auto Append

 Auto Split

 New Folder

 New Subfolder

 New Clip

 Create Folder from Disk

 Create Clip from Text File

 Duplicate Clips

 Move Clips

 Move Up

 Move Down

 Move to Top

 Move to Bottom

 Reverse Clips

 Merge Clips

 Split Clips

 Extract Label

 Delete Duplicate Clips

 Delete Clips

 Undo

 Redo

 Cut Clip to Clipboard

 Copy Clip to Clipboard

 Paste as New Clip

 Insert Date & Time

 Auto Adjust Case

 Auto Adjust Quotes

 Visualize Spaces

 Insert and Extract Text

 Reformat Text

 Convert Characters

Go To Collection

The Go To Collection button is only available on the collection toolbar when the All Collections tab is active.

When you click this button or select Go To Collection in the Clip menu, AceText will activate the collection that contains the active clip. Since the All Collections tab is the only tab showing clips from other collections, the command is only available on this tab.

Automatic Capture

The Windows clipboard can only hold one piece of information, which is often insufficient. AceText's ClipHistory, however, can hold as many clips as you want. Whenever you copy some text to the Windows clipboard, AceText grabs it and adds it to the ClipHistory. That way, you can easily work with many bits of text.

Whenever AceText automatically captures a clip into the ClipHistory it flashes the icon of the  New Clip command in the notification area on the Windows taskbar next to the system clock. If you don't see the icon, click the < button in the notification area and drag AceText's icon from the popout onto the taskbar. This way you can confirm with a glance whether AceText did capture the text you just copied. If it flashes the  Lock Collection icon that means AceText couldn't capture the text you just copied because the ClipHistory is locked. You'll need to unlock it first. If the icon doesn't flash at all that means AceText ignored the clip for one of the reasons explained in the following paragraphs.

You can use the Automatic Capture button on the ClipHistory toolbar or the Automatic Capture item in the Operation menu to temporarily stop AceText from automatically capturing any text you copy. This can be useful if you'll be copying and pasting some sensitive data that you don't want AceText to make copies of.

In the Operation Preferences you can configure the maximum size and maximum number of clips that AceText should capture and how to deal with duplicate clips.

If you never want AceText to automatically capture text copies by certain applications or windows then you can configure those in the Applications Preferences and the Windows Preferences. There you can permanently disable automatic capture for specific applications and windows.

An application can signal AceText and some other clipboard managers not to automatically capture text it copies by placing the `ExcludeClipboardContentFromMonitorProcessing` format onto the clipboard in addition to the actual text. AceText never automatically captures text that is copied with this flag, regardless of any settings. The clipboard history that is built into Windows 10 and 11 also respects this flag.

This method allows the application that copies the text to control whether clipboard managers (that support this flag) should capture the text. It could even do so selectively, by adding the flag only when copying certain text. AceText itself places this flag on the clipboard every time it copies something to the clipboard. This way, if you are using multiple clipboard managers (that support this flag), your clips don't get needlessly duplicated in the other clipboard managers.

If you would like an application to support this flag, you can ask the developers of that application to implement the following. The application should declare a 32-bit unsigned integer variable that we'll call

CF_ExcludeClipboardContentFromMonitorProcessing. Before the application copies any text to the clipboard, it should perform this assignment once:

```
CF_ExcludeClipboardContentFromMonitorProcessing :=
    RegisterClipboardFormat('ExcludeClipboardContentFromMonitorProcessing');
```

Then it can copy text to the clipboard and tell AceText and other compatible clipboard managers not to capture it like this:

```
if OpenClipboard(hwnd) then begin
    EmptyClipboard();
    SetClipboardData(CF_ExcludeClipboardContentFromMonitorProcessing,
        GlobalAlloc(GMEM_MOVEABLE or GMEM_DDESHARE or GMEM_ZEROINIT, 8));
    // Call SetClipboardData again to add the actual text or other data
    CloseClipboard();
end;
```

When putting multiple formats on the clipboard, it is very important that all calls to SetClipboardData() happen between a single pair of calls to OpenClipboard() and CloseClipboard(). This ensures that AceText and other clipboard managers only see a single clipboard change event. Then the order in which you add the different formats doesn't matter.

The actual data the application places on the clipboard for the CF_ExcludeClipboardContentFromMonitorProcessing format doesn't matter. AceText only checks whether the clipboard format is available. It doesn't look at the data. SetClipboardData(CF_ExcludeClipboardContentFromMonitorProcessing, 0) would also work. But then the format would be removed from the clipboard when your application terminates. Allocating a dummy buffer of 8 bytes ensures the clipboard retains the CF_ExcludeClipboardContentFromMonitorProcessing flag when the application terminates. The clipboard itself will deallocate the buffer when it is no longer needed.

AceText also recognizes the "Clipboard Viewer Ignore" flag that was invented by Chris Thornton, developer of ClipMate, a decade or two before Microsoft came up with the ExcludeClipboardContentFromMonitorProcessing flag. It works in exactly the same way. You just use a different string to register the clipboard format:

```
CF_CLIPBOARD_VIEWER_IGNORE :=
    RegisterClipboardFormat('Clipboard Viewer Ignore');
```

When AceText uses the clipboard to transfer text to another application it adds both the ExcludeClipboardContentFromMonitorProcessing and "Clipboard Viewer Ignore" formats to the clipboard to signal both modern and legacy clipboard managers to ignore the operation.

Auto Append

The Auto Append button is available on the collection toolbar when the ClipHistory tab is active.

When AceText captures text into the ClipHistory because  Automatic Capture is active or because you press the hotkey to capture text on the clipboard into the ClipHistory, Auto Append adds that text to the

most recently captured clip in the ClipHistory rather than adding a new clip. This way you can easily create a single clip from many bits of text copied from another application.

If AceText has already captured multiple bits of text into separate clips then you can append them using the  Merge Clips button on the collection toolbar.

When you click the Auto Append button, you will be asked which characters, if any, you want to use to delimit each bit of copied text in the clip. You can select one of the common delimiters, or type in whatever text you want to use as a delimiter. If you do not use a delimiter, the copied pieces of text are appended to the clip one right after the other, without any space between them.

Click the downward pointing arrow next to the Auto Append button to quickly turn on Auto Append using one of the common delimiters. The drop-down menu also has an option labeled "Insert at The Start". Selecting this makes Auto Append add the text to the start instead of to the end of the clip. That will still be the topmost or bottommost clip in the ClipHistory as chosen in the Operation Preferences.

When Auto Append is enabled, clicking the Auto Append button disables it. To change the delimiter while Auto Append is on, either turn off Auto Append and turn it on again, or click the downward pointing arrow to access the Auto Append drop-down menu to pick another delimiter.

If the hotkey to capture text on the clipboard into the ClipHistory is disabled then activating Auto Append automatically activates Automatic Capture too. Otherwise, Auto Append would have no effect. Turning off Automatic Capture then also automatically turns off Auto Append. If the hotkey is enabled then you can turn Auto Append and Automatic Capture on or off independently as Auto Append can also work with the hotkey.

Activating Auto Append automatically cancels Auto Split.

See Capture, Enter and Store Text to learn how to use AceText.

Auto Split

The Auto Split button is available on the collection toolbar when the ClipHistory tab is active.

When AceText captures text into the ClipHistory because  Automatic Capture is active or because you press the hotkey to capture text on the clipboard into the ClipHistory, Auto Split automatically splits that text into multiple clips along your chosen delimiter. This way you can easily paste separate items from a list that was copied by another application.

If AceText has already captured the clip you want to split then you can use the  Split Clips command to split it afterwards.

When you click the Auto Split button, you will be asked which characters along which you want to split the copied text. You can select one of the common delimiters, or type in whatever text you want to use as a delimiter. If you copy some text that does not contain the delimiter, it is added as a single clip.

Click the downward pointing arrow next to the Auto Split button to quickly turn on Auto Split using one of the common delimiters. The drop-down menu also has an option labeled "Reverse Split Clips". This reverses

the order of the split clips in the ClipHistory. The clips are still added to the top or bottom of the ClipHistory as chosen in the Operation Preferences.

When Auto Split is enabled, clicking the Auto Split button disables it. To change the delimiter while Auto Split is on, either turn off Auto Split and turn it on again, or click the downward pointing arrow to access the Auto Split drop-down menu to pick another delimiter.

To split more complicated clips, you can use a regular expression. AceText can apply the regular expression in two ways. If you select the “split” option, AceText deletes each regular expression match. The bits of text between two matches (as well as the bits before the first and after the last match) are captured into the ClipHistory. Splitting along the regular expression `[; ,]`, for example, treats both semicolons and commas as delimiters.

If you select the regular expression “match” option, AceText finds all regex matches in the captured text and stores each match into a new clip. Parts of the captured text not matched by the regular expression are discarded. If the regex does not match the captured text at all, nothing is added to the ClipHistory. Splitting a clip matching the regular expression `\w+`, for example, extracts all the words from the clip, putting each word into a new clip. Punctuation and whitespace are discarded.

If the hotkey to capture text on the clipboard into the ClipHistory is disabled then activating Auto Split automatically activates Automatic Capture too. Otherwise, Auto Split would have no effect. Turning off Automatic Capture then also automatically turns off Auto Split. If the hotkey is enabled then you can turn Auto Split and Automatic Capture on or off independently as Auto Split can also work with the hotkey.

Activating Auto Split automatically cancels Auto Append.

See Capture, Enter and Store Text to learn how to use AceText.

New Folder

The New Folder button is available on the collection toolbar when an AceText Collection tab is active.

Click the New Folder button, or press Shift+F4 on the keyboard to add a new folder to the collection. This moves keyboard focus to the edit box for the label. You can enter the folder’s label immediately.

If multiple clips and/or folders are selected when you add the new folder then those clips and/or folders are placed inside the new folder. The new folder is placed inside the folder that previously held the selected items. If only one clip or one folder is selected then the new folder is added as a sibling to the selected clip or folder and placed above it.

If you prefer the folder to be added below the active clip or inside the active folder then use the  New Subfolder button instead of the New Folder button.

See Capture, Enter and Store Text to learn how to use AceText.

New Subfolder

The New Subfolder button is available on the collection toolbar when an AceText Collection tab is active.

Click the New Subfolder button, or press Ctrl+F4 on the keyboard to add a new folder to the collection. This moves keyboard focus to the edit box for the label. You can enter the folder's label immediately.

The new folder is added as a subfolder to the active folder. If a clip is active then the folder is added as a sibling to the selected clip, below the selected clip. Unlike the  New Folder command, the New Subfolder command does not take into account whether multiple clips are selected. Nothing is moved automatically into the newly added subfolder.

If you want to add a folder above the active clip or if you want a newly added folder to become the parent folder of the selected items then use the  New Folder button instead of the New Subfolder button.

See Capture, Enter and Store Text to learn how to use AceText.

New Clip

The New Clip button is available on the collection toolbar when the ClipHistory tab or an AceText Collection tab is active.

Click the New Clip button or press F4 on the keyboard to create a new, blank clip. This moves keyboard focus to the edit box for the clip's contents. You can enter the clip's text immediately. If you'd rather start with entering the clip's label, there is an option in the Operation Preferences to focus the Label box instead.

If you add the new clip when a clip is selected then the new is added below the selected clip, inside the folder that contains it. If the parent folder is not sorted and you want the new clip to above the selected clip, click the  Move Up after adding the clip.

If you add the new clip when a folder is selected then the new clip is added inside the selected folder. If you want the added clip to be a sibling of the selected folder, click the  Move Up after adding the clip. If the parent folder is not sorted and you want the added folder to be a sibling below the selected folder, first move the clip above the folder. Then select the folder again and click the  Move Up button to move it above the clip.

See Capture, Enter and Store Text to learn how to use AceText.

Create Folder from Disk

The Create Folder from Disk button is available on the collection toolbar when an AceText Collection tab is active. When you click it, a dialog box pops up. Select the folder you want to add. You can enter or paste the path to the folder in the edit box at the top. Or you can click on the folder in the folder tree.

When you OK the dialog, AceText adds a new folder to your clip collection. The name of the folder on disk that you selected is used as the label of the new clip folder. The full path to the folder on disk is used as the

clip folder's URL. If you ticked "Add subfolders" in the dialog, then any subfolders of the folder on disk are added as subfolders to the new clip folder. The "Add empty folders" checkbox then determines whether clip folders are added for subfolders that don't contain any files matching the file mask.

Inside the new clip folders, AceText adds one clip for each file inside the folder on disk. It loads the files in the same way as the  Create Clip from Text File command.

If you don't specify a file mask, all files are added. If you do specify a file mask, only files matching the file mask are added.

In a file mask, the asterisk (*) represents any number (including none) of any character, similar to `.*` in a regular expression. The question mark (?) represents one single character, similar to `?` in a regular expression. The file mask `*.txt` tells AceText to add any file with a `.txt` extension.

File masks also support a simple character class notation, which matches one character from a list or a range of characters. To add all web logs from September 2003, for example, use a file mask such as `www.200309[0123][0-9].log` or `www.200309??.log`. To add a literal opening square bracket to a file mask, you need to place it into a character class. The closing square bracket has no special meaning outside of a character class. If you want to add a literal closing square bracket to a character class, place it immediately after the opening square bracket. So the file mask `*[[][0-9]].txt` matches file names like `whatever[1].txt`. In this file mask, `[[]` is a literal opening bracket, `[0-9]` is a single digit, and `]` is a literal closing bracket.

You can delimit multiple file masks with any mixture of semicolons, commas, and line breaks. A file is added if it matches at least one of the file masks in your list. To add all C source and header files, use `*.c;*.h`. To add a literal semicolon or comma to a file mask, either place the semicolon or comma between square brackets, or place the whole file mask between double quotes. The file mask `*[,]*.txt;"*;*.doc"` adds all `.txt` files that have a comma in their name, and all `.doc` files that have a semicolon in their name.

A file mask is only considered to match the file name if it matches the whole file name. The mask `*.txt` matches `joe.txt`, but not `joe.txt.doc` since the latter does not end in `txt`. You could use the mask `*.txt*` to match both.

See Capture, Enter and Store Text to learn how to use AceText.

Create Clip from Text File

The Create Clip from Text File button is available on the collection toolbar when the ClipHistory tab or an AceText Collection tab is active.

Click the Create Clip from Text File button or press Shift+Ctrl+O on the keyboard to select one or more files. Each file you select is added as a new clip to the active collection. The names of the files are used as labels for the clips. Their full paths are used as URLs for the clips. The date on the clip is set to the file's last modification date. The full path is also matched against the masks that you can customize in the Syntax Preferences to automatically select a syntax coloring scheme for the clip.

The contents of the file are stored into the AceText collection. Though the clip's URL remembers which file the clip was created from, this does not represent an active link to the file. Editing or overwriting the file does

not automatically update the clip in AceText. Editing a clip created from a file in AceText does not update that file.

AceText can only read plain text files, such as text documents (.txt), HTML files (.html), XML data (.xml), software source code, etc. AceText automatically detects Unicode files, with or without a Unicode signature (BOM). Unicode files are loaded as plain text clips. If a file does not look like it's a Unicode file then AceText checks whether the file contains NULL bytes. If it does then AceText loads the file as binary data. All other files are loaded as text files using your computer's default non-Unicode encoding.

AceText cannot decode proprietary file formats, such as Microsoft Word documents (.doc), Excel spreadsheets (*.xls) or Adobe PDF files (*.pdf). It loads them as binary data. To transfer text from such files into AceText, open the file in its corresponding application. Then use copy and paste or drag and drop between that application and AceText. The software then decodes its own proprietary file format and offers the contents as plain text to AceText.

See [Capture, Enter and Store Text](#) to learn how to use AceText.

Duplicate Clips

The Duplicate Clips button is available on the collection toolbar when the ClipHistory tab, the All Collections tab, or an AceText Collection tab is active. You can also find this button on the AceText Tower.

When you click the Duplicate Clips button, a popup menu appears below it. When you press Ctrl+D on the keyboard, the same menu appears to the right of the list of clips. The menu lists all AceText collections that you have opened. Select one of them from the menu to duplicate all selected clips and folders into that collection.

If the selected clips and folders are inside one or more folders, the same folder structure is created in the collection you are duplicating the items into. If the folder structure already exists in the target collection, the items are placed into the existing folders.

By default, read-only collections are not listed in the popup menu. They will be if you turn on “try to obtain write access when activating a read-only collection” in the Files Preferences. If you select a read-only collection to duplicate clips into then AceText will try to obtain write access before duplicating the clips. If it can the clips are duplicated. The target collection is not automatically made read-only again until one of the other options in the Files Preferences kicks in. If AceText cannot obtain write access then it pops up a message saying so.

See [Manage and Arrange Text](#) to learn how to use AceText.

Move Clips

The Move Clips button is available on the collection toolbar, except when the Clipboard tab is active. You can also find this button on the AceText Tower.

When you click the Move Clips button, a popup menu appears below it. When you press Ctrl+M on the keyboard, the same menu appears to the right of the list of clips. The menu lists all AceText collections that

you have opened, except the active collection. Select one of them from the menu to move all selected clips and folders into that collection.

If the selected clips and folders are inside one or more folders, the same folder structure is created in the collection you are moving the items into. If the folder structure already exists in the target collection, the items are placed into the existing folders.

By default, read-only collections are not listed in the popup menu. They will be if you turn on “try to obtain write access when activating a read-only collection” in the Files Preferences. If you select a read-only collection to move clips into then AceText will try to obtain write access before duplicating the clips. If it can the clips are moved. The target collection is not automatically made read-only again until one of the other options in the Files Preferences kicks in. If AceText cannot obtain write access then it pops up a message saying so.

See *Manage and Arrange Text* to learn how to use AceText.

Move Up

The Move Up button is available on the collection toolbar when the ClipHistory tab or an AceText Collection tab is active. You can also find this button on the AceText Tower.

Click the Move Up button or press Shift+Ctrl+Arrow Up on the keyboard to move the selected clips and/or folders one position upward in the list. If the selected clips are at the top of the folder that contains them, they will be moved into the folder above it.

If the selected clips are in a folder that is sorted, then moving them up immediately moves them into the folder above the folder that contains them. You cannot change the order of clips within a folder while it is sorted.

See *Manage and Arrange Text* to learn how to use AceText.

Move Down

The Move Down button is available on the collection toolbar when the ClipHistory tab or an AceText Collection tab is active. You can also find this button on the AceText Tower.

Click the Move Down button or press Shift+Ctrl+Arrow Down on the keyboard to move the selected clips and/or folders one position downward in the list. If the selected clips are at the bottom of the folder that contains them, they will be moved into the folder below it.

If the selected clips are in a folder that is sorted, then moving them down immediately moves them into the folder below the folder that contains them. You cannot change the order of clips within a folder while it is sorted.

See *Manage and Arrange Text* to learn how to use AceText.

Move to Top

The Move to Top button is available on the ClipHistory toolbar. Click it to move the selected clips to the top of the ClipHistory.

The Move to Top command is also available in the Clip menu. It is enabled when the ClipHistory tab or an AceText Collection tab is active. In a collection, it moves the selected clips and folders to the top of the collection. If the collection is sorted, then it moves the selected clips and folders out of their parent folders into the root of the collection. They are then sorted between the other clips and folders in the root of the collection.

In the Operation Preferences there are options for automatically moving clips to the top of the ClipHistory when they are copied again by another application or when AceText sends them to another application.

See Manage and Arrange Text to learn how to use AceText.

Move to Bottom

The Move to Bottom button is available on the ClipHistory toolbar. Click it to move the selected clips to the bottom of the ClipHistory.

The Move to Bottom command is also available in the Clip menu. It is enabled when the ClipHistory tab or an AceText Collection tab is active. In a collection, it moves the selected clips and folders to the bottom of the collection. If the collection is sorted, then it moves the selected clips and folders out of their parent folders into the root of the collection. They are then sorted between the other clips and folders in the root of the collection.

In the Operation Preferences there are options for automatically moving clips to the top of the ClipHistory when they are copied again by another application or when AceText sends them to another application.

See Manage and Arrange Text to learn how to use AceText.

Reverse Clips

The Reverse Clips button is available on the ClipHistory toolbar. Click it to reverse the order of the selected clips in the ClipHistory. Selected clips below the topmost selected clip are placed immediately above the topmost selected clip in reverse order.

In the Operation Preferences you can choose whether automatically captured clips are added to the top of the ClipHistory (like a stack growing upwards) or to the bottom of the ClipHistory (like a list growing downwards). With the Reverse Clips command you can switch between the two for clips that have already been captured.

Merge Clips

The Merge Clips button is only available on the collection toolbar when the ClipHistory tab, the All Collections tab, or an AceText Collection tab is active.

Click the Merge Clips button on the collection toolbar or press Ctrl+G on the keyboard to merge or group all selected clips into a single clip. You will be asked which characters, if any, you want to use to delimit the pieces of text from the original clips in the merged clip. You can select one of the common delimiters, or type in whatever text you want to use as a delimiter. If you do not use a delimiter, the pieces of text are put into the merged clip one right after the other, without any space between them.

If some of the clips that you want to merge may have extraneous line breaks or whitespace then you can choose to trim leading and trailing line breaks or whitespace from those clips before merging them. This does not affect the original clips if you choose to keep them. It only ensures that there is no whitespace between text from different clips in the merged clip other than your delimiter.

You can also choose where you want to store the merged clip. The first option, “replace the original clips”, deletes the selected clips from the collection and inserts the merged clip in their place. All the other options leave the selected clips in place. The options “before the first original clip” and “after the last original clip” insert the new clip into the collection before or after the selected clips. The option “on the clipboard” copies the merged text to the clipboard without adding it to the collection and without automatically pasting it anywhere.

The last option, “send to target application”, uses the merged clip to complete an  AcePaste or  AceType operation if you pressed either of those hotkeys in another application prior to merging the clips. Otherwise, it sends the merged clip to the last application that was active before you switched to AceText to merge the clips. Sending the merged clip to the target application may or may not put it on the clipboard, depending on whether you configured AceText to send keystrokes to that application.

You can also choose a sort order for merging the clips. This does not move the original clips in the collection. It only affects the order in which the text of the original clips is put into the merged clip. If the selection includes folders, then “sort flattened list of clips” sorts the clips in those folders as if those clips weren’t in folders at all. With the other options, clips are sorted within their folders and merged in the order of the folders that contain them.

Click the downward pointing arrow next to the Merge Clips button to quickly merge clips using one of the common delimiters. Doing so always replaces the merged clips and merges them using their present order.

See Manage and Arrange Text to learn how to use AceText.

Split Clips

The Split Clips button is available on the collection toolbar when the ClipHistory tab or an AceText Collection tab is active.

Click the Split Clips button on the collection toolbar to split one or more clips into multiple clips.

When you click the Split Clips button, you will be asked which characters AceText should interpret as delimiters in the original clips. A delimiter can be a single character or a string of characters. Those delimiters will be deleted, and the bits of text between them will be stored in new clips. E.g. to split a clip holding a comma-delimited list of items, select the comma as the delimiter. AceText will then create a new clip for each item in the comma-delimited list.

To split more complicated clips, you can use a regular expression. AceText can apply the regular expression in two ways. If you select the “split” option, AceText deletes each regular expression match. The bits of text between two matches (as well as the bits before the first and after the last match) are stored in new clips. Splitting along the regular expression `[;.]`, for example, treats both semicolons and commas as delimiters.

If you select the regular expression “match” option, AceText finds all regex matches in the clip and stores each match into a new clip. Parts of the original clip not matched by the regular expression are discarded. Splitting a clip matching the regular expression `\w+`, for example, extracts all the words from the clip, putting each word into a new clip. Punctuation and whitespace are discarded.

You can also choose how to add the split clips to the collection. If you select “replace the original clips” then each clip that you selected is replaced with the clips that it was split into. The other two options leave the original clips in place and insert the clips that are split from each clip above or below the original clips.

Click the downward pointing arrow next to the Split Clips button to quickly split clips using one of the common delimiters. Doing so always replaces the original clips with the split clips.

See Manage and Arrange Text to learn how to use AceText.

Extract Label

A clip's label is used to identify the clip in the list of clips in the ClipHistory and in the tree of clips in an AceText Collection. If a clip does not have a label then the start of the clip's text is used instead. But this can make the collection look disorganized, particularly if you have many clips starting with similar text. It also causes the de facto label to change whenever you edit the clip's contents.

The Extract Label button on the collection toolbar lets you use a specific part of the clip's text as the clip's label. A dialog box appears when you click the button.

You can choose to use the text from the start of the clip stopping at a certain delimiter. The delimiter can be a single character or a string of characters. It can even be a regular expression match. If you loaded some comma-delimited values into a clip, for example, you could first split the clip along line breaks, giving you one clip for each CSV row. Then you could use the Extract Label command with the comma as the delimiter to use the first field of each row as the label.

If you want to use text from the middle of the clip as its label, select the option "text matched by a regular expression". Then the first match of this regular expression in the clip's text is used as the label. If you want to use only part of the regex match as the label, add a capturing group named “label” to the regular expression. Then only the text matched by this named capturing group is used as the label. With the regex `<a[^\<>]*>(?'label'.*?)`, for example, you could use the text of the first link in an HTML snippet as the label.

The Extract Label command actually stores the new label in the clip's Label field. If you edit the clip's contents afterwards, the label is not automatically updated. You can use the Extract Label command again to extract a new label.

If you don't need the clip's contents to retain the label, select one of the first two options in the group "alter the clip's text". The last option leaves the clip unchanged. When using a regex match with a capturing group named "label", the option to remove the label and the delimiter removes the whole regex match, while the option to remove the label but not the delimiter removes only the text matched by the group named "label".

Delete Duplicate Clips

The Delete Duplicate Clips button is only available on the collection toolbar when the ClipHistory tab, the All Collections tab, or an AceText Collection tab is active. A dialog box appears when you click the button.

First you can choose the scope from which to delete duplicate clips. This can be the entire collection or just the clips that you have selected. If your selection includes folders, then the clips inside those folders are also considered to be selected.

Then you can choose which clips that each clip should be compared with to see whether it is a duplicate. If you select "anywhere in the scope", then any clip that is a duplicate of any other clip higher up in the entire collection or in the selection is deleted. If you select "clips in the same folder only", then any clip that is a duplicate of a sibling clip higher up in the same folder is deleted. The sibling must also be selected for it to be compared if the scope is set to "selected clips". If you set the proximity rule to "adjacent clips only", then a clip is deleted if it is a duplicate of the clip immediately before it within the same folder. The adjacent clip does not need to be selected for it to be compared.

Finally, you can choose which parts of the clips that should be compared, and whether the comparison should be case sensitive. If you select multiple parts then all of those parts must be the same between two clips for them to be treated as duplicates. Parts that you don't select are not compared. Blank parts are considered to be the same. If you decide to delete duplicate clips comparing only their labels, for example, and you have multiple clips without labels, then all but one of those will be deleted.

Delete Clips

The Delete Clips button is available on the AceText Tower and in the Edit menu. It is also available on the Recycle Bin toolbar. Pressing the Delete key on the keyboard while the list or tree of clips has keyboard focus also invokes the Delete Clips command.

Deleting clips from the ClipHistory or a collection moves those clips into the recycle bin. Deleting clips from the recycle bin removes them permanently.

See the Recycle Bin topic to learn the difference between restoring clips from the recycle bin and via the  Undo command.

Undo

The Undo button is available on the collection toolbar for all tabs. Click the button on press Ctrl+Z on the keyboard to undo the last change. Click the downward pointing arrow next to the Undo button to see the last 16 changes. If you click on an item in the list, that change and all changes above it (i.e. made after it) are undone.

While the undo button only shows 16 changes at a time, AceText remembers all changes you've made since opening the clip collection. You can undo them all by issuing the undo command repeatedly. Use the Redo button if you change your mind about undoing something.

To keep the length of the undo history manageable, AceText consolidates editing actions affecting the same clip when you activate another clip. If you type three words at different spots into the text of a clip, for example, then those three actions appear as separate actions into undo history. You can undo them one by one. Once you switch to another clip, AceText consolidates the three typing actions into a single "edit clip" action that undoes all three typing actions at once.

AceText keeps separate undo information for each tab. Clicking the Undo button only undoes changes made to the active tab. They are undone independently of other tabs. Actions that affect multiple collections, such as moving clips or search-and-replace across collections, have to be undone in both collections if you want to completely undo them. The advantage is that if you make changes in collection A, then make changes in collection B, you can undo some of the changes you made to A without having to undo anything in B.

This undo history separation even affects simple commands like  Delete Clips. When this command moves a clip from a collection into the Recycle Bin, the collection's undo history gets an item for the deletion. The recycle bin gets a separate undo history item for adding the deleted clip to the recycle bin. If you click the Undo button on the collection tab, the deleted clip is restored without removing it from the recycle bin (and without needing it to still be in the recycle bin). You can click the Undo button on the Recycle Bin tab to remove the recycled clip.

Redo

The Redo button is available on the collection toolbar for all tabs. Click the button on press Ctrl+Y on the keyboard to redo the last undone change.

The redo history is separate for each tab, just like the undo history. You can only redo changes immediately after undoing them. If you undo multiple successive changes, you can redo all of them. But as soon as you make another change that's undoable, the redo history is cleared. The only exception are trivial changes like moving the text cursor or selecting another clip. Those won't clear the redo history. If you redo an undone change after making a trivial change, the trivial change is undone before the action you wanted to redo is redone.

Cut Clip to Clipboard

The Cut Clip to Clipboard button is available on the collection toolbar, except when the Clipboard tab is active. You can also find this button on the AceText Tower.

This command places the selected clips on the clipboard in the same way as the  Copy Clip to Clipboard command. It then removes the selected clips from the collection. The removed clips are not added to the Recycle Bin. You can restore the cut clips using the  Paste as New Clip command (before the clipboard is overwritten) or by using the  Undo command (which also requires you to undo everything you did in the same collection after you cut the clips).

See Manage and Arrange Text and Look up and Reuse Text to learn how to use AceText.

Copy Clip to Clipboard

The Copy Clip to Clipboard button is available on the collection toolbar, except when the Clipboard tab is active. You can also find this button on the AceText Tower.

Click the Copy Clip to Clipboard button or press Shift+Ctrl+C on the keyboard to place the selected clip onto the Windows clipboard. You can then paste the clip into another application to use the clip's text in that application. Or, you can paste the clip back into AceText to duplicate it.

If several clips and/or folders are selected, all of them are copied onto the Windows clipboard. All of them can be pasted into AceText with the  Paste as New Clip command. When pasting into another application, however, only the active clip will be pasted. The active clip is the one of which the text is visible in the AceText Editor. If a folder is active, nothing can be pasted into another application.

Note that the shortcut for copying the entire clip is Shift+Ctrl+C. The regular copy shortcut Ctrl+C copies the selected part of the control that has keyboard focus. If you are editing the clip's label, for example, Ctrl+C copies the selected part of the label.

Instead of using copy and paste to duplicate clips in AceText, you can use the  Duplicate Clips button to duplicate the clips in one action, without using the Windows clipboard.

See Manage and Arrange Text and Look up and Reuse Text to learn how to use AceText.

Paste as New Clip

The Paste as New Clip button is available on the collection toolbar when the ClipHistory tab or an AceText Collection tab is active. You can also find this button on the AceText Tower.

Click the Paste as New Clip button or press Shift+Ctrl+V to create a new clip from the text on the Windows clipboard, regardless of which application copied the text to the clipboard. If you used  Cut Clip to Clipboard or  Copy Clip to Clipboard in AceText to place several clips and/or folders on the clipboard then all of them are pasted.

Note that the keyboard shortcut Shift+Ctrl+V pastes the text as a new clip, while the regular paste shortcut Ctrl+V pastes the text into the control that has keyboard focus. If you are editing a clip's label, for example, Ctrl+V pastes (the first line of) the text on the clipboard into the label.

If the text being pasted was copied to the window by an application or window that you configured in the Applications Preferences or Windows Preferences then the pasted text uses the syntax coloring scheme you

configured for that application or window. If you disabled both the Clipboard tab and automatic capture to the ClipHistory for that application or window, then the text is pasted as a secret clip.

See Capture, Enter and Store Text and Manage and Arrange Text to learn how to use AceText.

4. Collection Menu

The Collection menu in the main menu lists all commands that affect the active collection entirely, and that open other collections. Many of these commands have buttons on the main toolbar. Those commands are described in the main toolbar reference section of this manual.

-  New Collection
-  Open Collection
-  Save Collection
-  Rename/Move Collection

-  Export Collection
-  Print Collection

Read Only

-  Set Password
-  Lock Collection
-  Sort Collection

Archive ClipHistory
Empty Recycle Bin
Sort Collection Tabs

-  Delete Collection

-  Close Collection
-  Shut Down AceText

Rename/Move Collection

Select Rename/Move Collection in the Collection menu to save the current collection under a different name and/or a different folder. The file that previously stored the collection is deleted. Any further changes you make to the collection will be automatically saved in the new location.

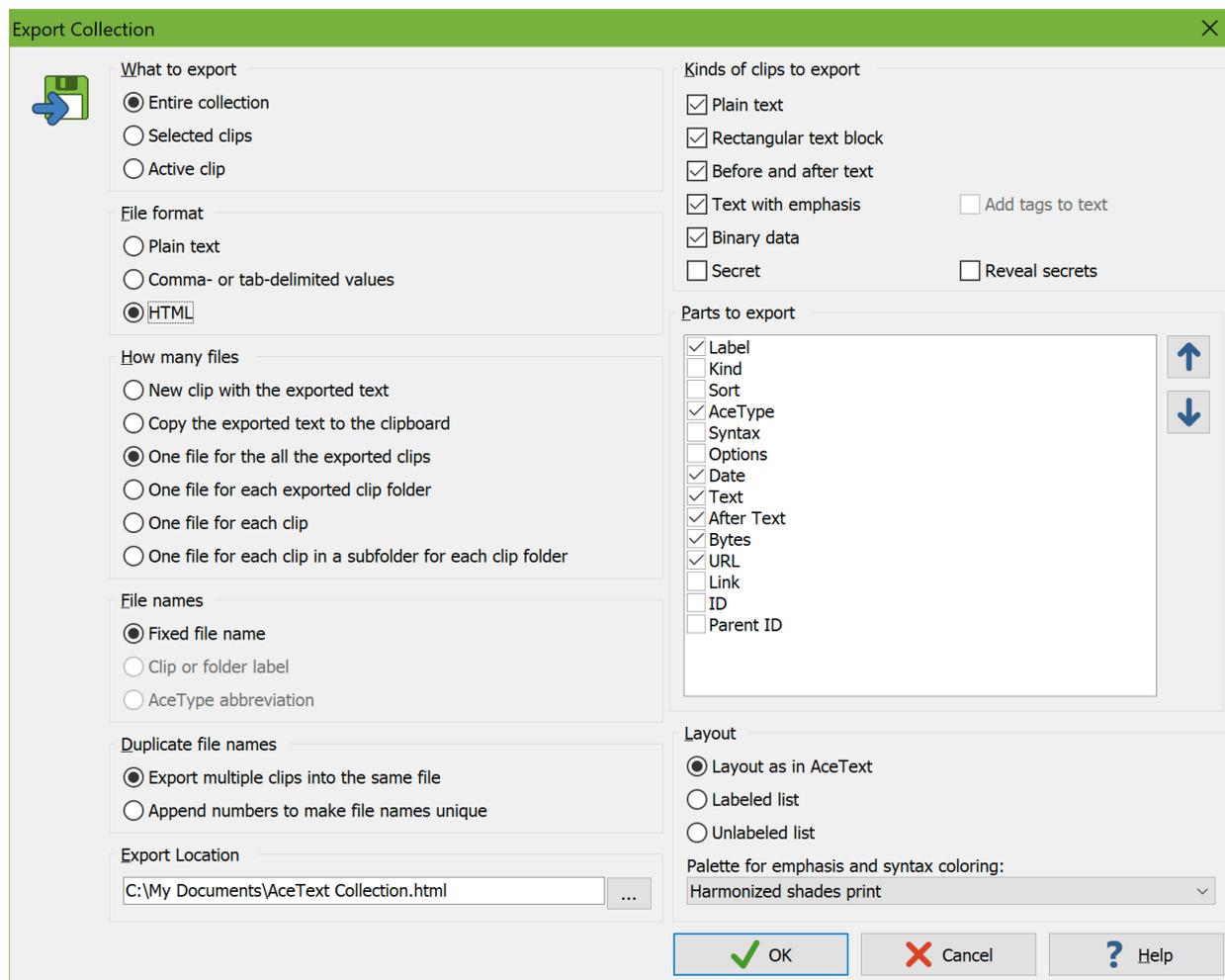
There is no toolbar button for the Rename/Move Collection command. If you've hidden the main menu, you can access the command by right-clicking on a collection's tab.

Export Collection

Select Export Collection in the Collection menu to export the current collection, the selected clips, or the active clips, into one of various file formats that can be read by other applications.

There is no toolbar button for the Export Collection command. If you've hidden the main menu, you can access the command by right-clicking on a collection's tab.

The Export Collection dialog offers a lot of options to choose what is exported and in which format. Some options are geared towards producing a document for a person (who doesn't use AceText) to read. Other options are geared towards producing files that can be processed with other applications.



What to Export

You can choose to export all the clips on the active collection tab, or the clips you've selected in the clip tree, or just the clip being shown in the AceText Editor. If you want to export all clips in all open collections, switch to the All Collections tab before exporting. If you only want to export the body text of the active clip without anything else, use the  Save Text item in the Text menu instead of the Export Collection item in the Collection menu.

File Format

The “plain text” file format exports your collection into .txt files that can be viewed in any text editor, including Microsoft Notepad.

The “comma- or tab-delimited values” choice allows you to export `.csv` or `.tsv` files. Which will be produced depends on the delimiter characters that you specify in the bottom right corner of the Export dialog. Many applications can import data from CSV and TSV files.

The “HTML” format produces files that can be used as web pages or that can be loaded into a word processor. This is the only export format for which you can choose a color palette to retain emphasis and syntax coloring. If you choose to export into a single file in the next section then AceText embeds the style sheets for emphasis and syntax coloring into the one HTML file. If you export into multiple HTML files then AceText exports the style sheets into a separate CSS file that is then referenced by all the HTML files.

If your goal is to programmatically process a collection then it may be easier to just process the `.atc` file that the collection is saved in instead of exporting your collection. The AceText Collection file format is based on XML with a schema that is published at the schema URL referenced by each `.atc` file. This gives you access to all the details of your collection.

Not all of the following options are available for all file formats. So you should choose your file format before proceeding to set the other options.

How Many Files

The first three options export everything into a single file. You can choose to add the contents of this file as a new clip to the collection you’re exporting or to copy the contents of this file to the clipboard. In those cases AceText only produces the exported file in memory. The file is only written to disk if you select “one file for all the exported clips”.

If you are exporting multiple folders then you can choose “one file for each exported clip folder” to export all the clips in each folder into a separate file for that folder. If the folder contains subfolders then the clips in those subfolders are also exported into separate files for each of those subfolders. The details of the folder itself are exported to the same file as the folder’s clips. If the folder doesn’t have any clips then the file is still created with only the details of the folder itself.

The last two options export each clip into a separate file. Folder details are not exported. You can choose to export a subfolder for each clip folder. This uses the label of each clip folder as the name of the subfolder.

File Names

When exporting to disk you can choose how AceText should determine the file name of each exported file. The “fixed file name” option is only available when exporting everything into a single file. You’ll specify the actual file name below in the Export Location section.

When exporting one file per folder or one file per clip you can use the label of each clip or folder as the name of the file that that clip or folder is exported to. If you didn’t enter a label for a folder or a clip then the file name uses the default label such as “New Folder” or the start of the clip’s contents as the folder or file name, just as the folder or clip is displayed with in the clip tree in AceText.

When exporting one file per clip you can choose to use the clip’s AceType abbreviation as the name of the file that the clip is exported to. The file name of the Export Location is used for clips that don’t have an AceType abbreviation.

Duplicate File Names

AceText allows collections and folders to contain multiple folders and multiple clips and with the same labels or AceType abbreviations. When using labels or abbreviations to determine file names this can result in duplicate file names. Since folders on disk cannot contain multiple files with the same name, you need to choose how to handle this eventuality.

If you choose “export multiple clips to the same file” then clips with the same label are all exported into one file using that label as its file name. With this option, “one file for each clip” and “one file for each clip folder” are not strictly true. Instead you get one file for each different clip label or one file for each different folder label.

If you chose “append numbers to make file names unique” then you do get one file per clip or one file per exported clip folder in all circumstances. When the same file name is needed for multiple clips or multiple folders, the second and following ones have (2), (3), etc. appended to the file name.

Export Location

Specify the full path to the file that should be exported. You can click the (...) button to select the file in Save dialog.

The export location should always include a file name with an extension that matches the file format. AceText changes the extension to match the file format when you change the file format. But it does not enforce this extension when you edit the export location, allowing you to use a custom extension.

Even when basing file names on labels or abbreviations you need to specify a file name for the export location. It will be used as a fallback when no file name can be determined based on the label or abbreviation. This can happen when a clip doesn't have an abbreviation or a label that contains only characters that are not valid in file names.

Kinds of Clips to Export

You can also choose which kinds of clips to export. If you deselect a certain kind of clip then no parts of clips of that kind are exported at all. If you deselect all clip kinds then only folders are exported. This can be useful to produce an outline of a large collection.

For secret clips there is an extra option whether the export may reveal secrets. If you tick “reveal secrets” and also tick the “text” part then the exported files will reveal the text of all the secret clips that you're exporting. If you tick the “secret” kind but untick “reveal secrets” then the exported files includes all the parts of secret clips except for the “text” part. This can indirectly reveal that your collection contains secret clips as the “text” part will be exported for other kinds of clips but not for secret clips. But the secrets themselves won't be revealed.

When exporting to plain text or a delimited text format, AceText cannot use actual colors and font styles to show emphasis. So for these formats there is another extra option to add tags to “text with emphasis” clips. If you tick this then the body text of “text with emphasis” clips has pseudo HTML tags added to the export. A clip with emphasis that looks like “label and **caption**” in AceText will be exported as “<lbl>label</lbl> and

<cap>caption</cap>”. There’s no option to add tags to preserve syntax coloring when exporting to plain text.

Parts to Export

The “label”, “kind”, and “URL” parts apply to both clips and clip folders. The “sort” part only applies to folders. The “AceType”, “syntax”, “options”, “date”, “text”, “after text”, and “bytes” parts are the various clip parts that you can edit in the AceText Editor. The “text” part also covers the “before” text of “before & after text” clips. The “link” part is the link that is generated by the  Edit|Copy Link command. The “ID” part is the item’s unique ID used by the Copy Link command. The “parent ID” parts is the ID of the folder that contains the item. Links and IDs are exported for both folders and clips if you select those parts.

You can use the Up and Down buttons to change the order of the exported parts. This determines the order in which the parts are exported for each clip each folder. This order is not used when you select “layout as in AceText” in the next section.

Layout

The “layout” section appears when exporting as plain text or as HTML. AceText can use spacing and indentation to arrange the exported parts of the clips similarly to how they are arranged in the AceText Editor. This is best if you want to present your collection for a person to read. If you want to process the export with another program then exporting the parts as a list produces a file that is easier to parse. A labeled list includes the labels from the “parts to export” list.

When exporting into HTML format, you can choose a palette for emphasis and syntax coloring. These are the same palettes that you can choose from and edit in the Appearance Preferences. If you choose a palette then clips with emphasis or syntax coloring will appear with that emphasis or syntax coloring using the colors from the selected palette when viewing the HTML file in a web browser. The palette’s background color is also used. If you intend to paste your HTML into a report that may end up being printed then you should probably choose one of the “print” palettes to give the exported HTML a pure white background.

Characters for Delimited Values

The “characters for delimited values” section appears when you set the file format to “comma- or tab-delimited values”. The value separator that you select determines whether the file will actually be comma-delimited or tab-delimited. You can also make it semicolon-delimited or pipe-delimited (vertical bar). Or you can enter any punctuation character or symbol of your choice as the value separator. In the exported file, the value separator will separate the different parts that are exported for the same clip or folder.

The row separator is the character that will separate different clips and folders. Windows files typically use CRLF line breaks while Linux or OS X files typically use LF line breaks. Some systems use the vertical tab so that line breaks within fields don’t need to be quoted or escaped. AceText also lets you use a page break (form feed character) or any punctuation character or symbol that is different from the value separator.

Since AceText clips can contain any text, there is always a possibility that one of the exported clips contains the value or row separator. To handle this, you need to specify a quote character, an escape character, or both.

If you specify only a quote character then fields that contain the value or row separator are surrounded by quote characters. If those fields also contain the quote character then those quotes are doubled. Quotes are not doubled in fields that contain the quote character but not the value or row separator.

If you specify only an escape character then that escape character is placed before the value separator, the row separator, and the escape character itself when these appear within fields.

If you specify both a quote and an escape character then fields that contain the value or row separator are surrounded by quote characters. The escape character is placed before all quote or escape characters within fields, regardless of whether those fields need to be surrounded by quote characters.

If you specify an escape character then you can also replace line breaks and tabs with escaped letters. If you tick this option and select the backslash as your escape character, then tabs are replaced with `\t`, carriage returns with `\r`, line feeds with `\n`, form feeds with `\f`, and vertical tabs with `\v`. If you enter another different escape character then you get the same letters with your chosen escape character. If you use this option and your value separator is a tab or your row separator is a line break then fields are never considered to contain that value or row separator as it will be replaced with an escaped letter, removing the need for the field to be quoted.

By default, AceText uses a comma as the value separator, a CRLF line break as the row separator, a double quote as the quote character, and no escape character. This is the format that Microsoft Excel uses for CSV files. If your computer is configured to use the comma as a decimal separator, then you should select the semicolon as your value separator for compatibility with Excel.

Read Only

Select Read Only in the Collection menu to toggle the active collection between read only mode and read/write mode. The collection's tab turns green when it's read only.

There is no toolbar button for the Read Only command. If you've hidden the main menu, you can access the command by right-clicking on a collection's tab.

If you are the only person using a particular collection, making it read only does nothing except protect the collection from accidental changes. If you're accessing a collection via your local network while other people may also be using it, an additional factor comes into play: only one person can modify a collection at the same time.

The first person to open a collection automatically opens it in read/write mode by default. Everybody else will be forced to open the collection in read only mode, until the first person either closes the collection, or makes it read only. Then, another person can turn off read only mode for that collection and modify it.

If you turn on read only mode after you've modified a collection, AceText first saves your changes. When you turn off read only mode, AceText checks whether the collection was changed on disk and automatically reloads it if it was. While working with a collection in read only mode, AceText periodically checks whether it was modified by somebody else. The net result is that when collaborating across the local network, you don't have to worry about collections going out of sync or overwriting other people's changes.

If you use file synchronization software that updates your files in the background, it won't be able to update your AceText collection files that you have open in read/write mode in AceText. When synchronizing your

files, make sure you don't have any collections open in AceText, or that the collections you have open are all in read-only mode.

In the Files Preferences you can enable various options to automatically make collections read-only in certain situations, or to open them as read-only by default. Use these options to give other people or other applications more opportunity to modify the collections that you have open in AceText.

Set Password

Select Set Password in the Collection menu to protect the active collection with a password, or to change the password. If you do not enter a password then any previous password is removed.

There is no toolbar button for the Set Password command. If you've hidden the main menu, you can access the command by right-clicking on a collection's tab.

Be careful when password-protecting collections. AceText encrypts collections with the "blowfish" algorithm, designed by world-famous cryptographer Bruce Schneier. This encryption cannot be broken except by guessing the password. If you forget your password then the clips in the collection will be irretrievably lost. To make sure you don't mistype your password, the Set Password dialog asks you to enter your password twice. If you're sure that nobody is looking over your shoulder, then you can untick the Mask Password checkbox. Then you need to enter the password only once. Inspect your password closely to make sure it is correct.

To make it infeasible to guess the password by using a computer program to try every password that you might have used, AceText uses the PBKDF2 algorithm with a very large number of rounds to derive the encryption key from your password. AceText uses a different random salt for each collection to ensure that each collection has a different encryption key even if you set the same password for them. It may take a second or so, depending on your computer's performance, for your new password to be applied to your collection. The algorithm is deliberately designed to be slow and CPU-intensive so that a program running on a powerful PC could perhaps try tens of passwords per second instead of billions per second.

If you want to use the same password for multiple collections, you can tick "also encrypt all open collections that do not have a password" and/or "replace the password of all unlocked collections with this password". AceText can use multiple CPU cores to derive the encryption keys for multiple collections at the same time. Using the same password for multiple collections is safe as long as you don't share the password with others and you don't use the same password with other applications or websites.

If you tick "lock the collections immediately" then setting the password also locks the collection. If you don't then the Set Password command does not have any obvious immediate effect. But when you close and reopen the collection then the collection will be locked. AceText does immediately derive the encryption key even if you don't lock the collection immediately. AceText does not keep your password in memory.

When a collection is locked, its tab is yellow and the collection appears empty. The AceText Editor displays a box where you can enter the password to unlock the collection. Unlocking a collection does not remove the password. If you close and re-open the collection, you'll be prompted for the password again.

If you try to add a secret clip to a collection that does not yet have a password, then AceText automatically pops up the Set Password dialog. If you cancel the dialog then the secret clip is not added to the collection. The options to use the same password for other collections and to immediately lock the collection are not

available in this situation. This way AceText can immediately add the secret clip to the collection after you set the password.

After setting a password or unlocking a collection with its password, you can use the Lock Collection command to lock the collection at any time. In the Files Preferences you can enable various options to automatically lock collections in certain situations. Use these options to make sure you don't accidentally leave sensitive collections unlocked.

You can put a password on the ClipHistory if you want to. If you do so then the ClipHistory will be non-functional each time you start AceText, until you unlock the ClipHistory by entering its password.

You can also put a password on the Recycle Bin. If you do then you will not be able to delete any clips until the recycle bin is unlocked. clips deleted from password-protected collections are saved in the recycle bin. Deleting a secret clip only adds it to the recycle bin if it has a password. If you work with password-protected collections then you should either put a password on the Recycle Bin too or turn on the option to purge the Recycle Bin when closing AceText in the Files Preferences.

Lock Collection

Select Lock Collection in the Collection menu to lock the collection with the password that you previously set with the Set Password command. When the collection is locked its tab will be yellow and the collection will appear to be empty. The AceText Editor will display a box where you can enter the password to unlock the collection.

There is no toolbar button for the Lock Collection command. If you've hidden the main menu, you can access the command by right-clicking on a collection's tab.

In the Files Preferences you can enable various options to automatically lock collections in certain situations. Use these options to make sure you don't accidentally leave sensitive collections unlocked.

Archive ClipHistory

Select Archive ClipHistory in the Collection menu to save the contents of the ClipHistory into an AceText collection file and then start with a blank ClipHistory. You will be asked for the name and location of the file that you want to archive the ClipHistory into. If you tick the "open the saved file" checkbox in the file selection dialog, then the archived ClipHistory will be opened into a collection tab.

There is no toolbar button for the Archive ClipHistory command. If you've hidden the main menu, you can access the command by right-clicking on the ClipHistory tab.

Empty Recycle Bin

Select Empty Recycle Bin in the Collection menu to delete all the clips in the Recycle Bin. Once clips are deleted from the Recycle Bin, there's no way to recover them.

There is no toolbar button for the Empty Recycle Bin command. If you've hidden the main menu, you can access the command by right-clicking on the Recycle Bin tab.

Sort Collection Tabs

Select Sort Collection Tabs in the Collection menu to arrange the tabs in AceText in their default order. From left to right, the default order is the Clipboard tab followed by the ClipHistory tab. If you have any AceText Collection tabs open, those follow the ClipHistory tab sorted in the alphanumerical order of their file names. The final two tabs are the All Collections tab and the Recycle Bin tab.

Delete Collection

Select Delete Collection in the Collection menu to permanently delete the current collection. The collection will be closed and its file deleted. The file will be placed in the Windows Recycle Bin (if stored on a drive with a Windows Recycle Bin). The clips will *not* be placed in the AceText Recycle Bin.

There is no toolbar button for the Delete Collection command. If you've hidden the main menu, you can access the command by right-clicking on a collection's tab.

5. Operation Menu

The Operation menu lists the options that control how AceText automatically captures text into the ClipHistory. They all have buttons on the ClipHistory toolbar. They are described in the collection toolbar reference section of this manual.

The Operation menu also lists commands for sending clips directly to other applications. These commands all have buttons on the main toolbar. They are described in the main toolbar reference section of this manual.

 Automatic Capture

 Auto Append

 Auto Split

 AcePaste Now

 AceType Lookup

 Double-Click Sends Clip

 SequencePaste Up

 SequencePaste Down

6. Clip Menu

The Clip menu in the main menu lists commands that affect the active clip or the clips you've selected in the active collection. Most of these commands have buttons on the collection toolbar. Those are described in the collection toolbar reference section of this manual.

-  New Folder
-  New Subfolder
-  New Clip
-  Create Folder from Disk
-  Create Clip from Text File
-  Generate Random Clip

-  Duplicate Clips
-  Move Clips
-  Move Up
-  Move Down
-  Move to Top
-  Move to Bottom
-  Reverse Clips

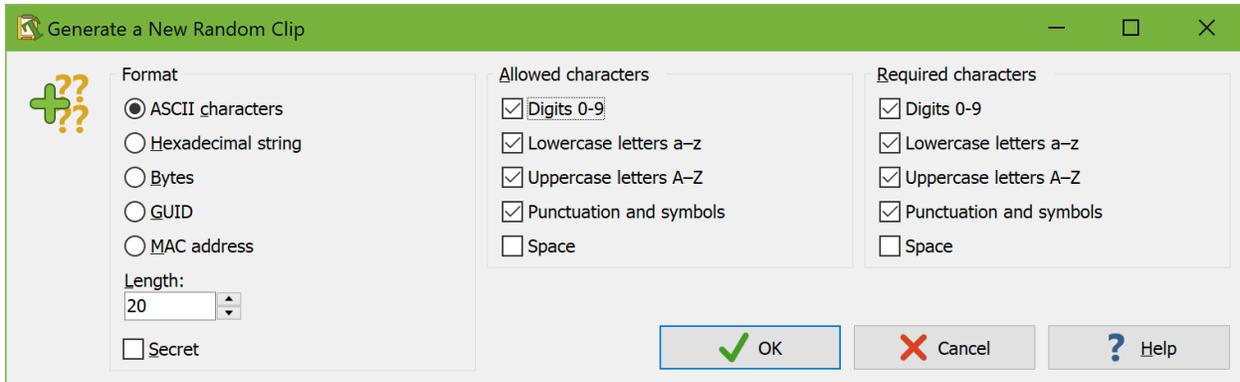
-  Merge Clips
-  Split Clips
-  Extract Label
-  Delete Duplicate Clips

-  Go To Collection
-  Browse to URL

Generate Random Clip

Select the Generate Random Clip in the Clip menu to add a new clip to your collection with semi-random contents. The contents are semi-random rather than fully random because they will follow the rules that you specify in the Generate Random Clip dialog. Within those rules, AceText ensures the clip is as random as possible by using the random number hardware that is built into your CPU.

If the active collection is protected with a password then you can tick the “secret” checkbox to add the clip as a secret clip. The contents of the clip won't be revealed unless you click the Reveal button after the clip is generated.



ASCII Characters

Generates a string of characters, such as `{ ' IF*rX1aG~4\?m9KMg~.`. Set “length” to the number of characters you want.

Tick one or more checkboxes in the “allowed characters” group to choose which characters may occur in the string. Ticking all the boxes allows all characters from positions 32 to 95 in the ASCII table. The string will never contain control characters or non-ASCII characters.

For maximum randomness, untick all the “required characters” options. If you do tick one or more of these, then the generated string will contain at least one character of each type of required character at a random position in the string. This also increases the minimum length to the number of different types of characters required. Requiring a certain type of character automatically allows a random number (zero or more) of additional characters of that type.

Hexadecimal String

Generates a string of hexadecimal characters, such as `BED9BE947CDD5C229D11.` The “length” you specify is the number of hexadecimal characters or nibbles you get. If you need a number of random bytes, set the length to twice the number of bytes you need. The hexadecimal characters are added as a plain text clip.

Bytes

Generates a “binary data” clip with random bytes. The “length” you specify is the number of bytes you get.

Secret clips cannot store binary data. If you tick the “secret” checkbox and select the “bytes” format then you’ll get a hexadecimal string instead. The length will still be interpreted as a number of bytes.

GUID

Generates a globally unique identifier (GUID) or universally unique identifier (UUID) in the form of `{3C8EBAC0-E7A1-4542-AF23-4F070DA53580}`. AceText uses the `UuidCreate()` Win32 API call to generate the GUID.

MAC address

Generates a string of 6 random bytes delimited with colons, such as `5B:EA:53:87:8E:E9`. The bytes are fully random. There are no options to make MAC address look like a local or universal address, for example.



Browse to URL

Every clip and folder in AceText can have an associated URL. This URL can be a web address or the path to a file on your computer. This allows a clip or folder to reference an external resource without making that resource part of the clip's contents.

AceText can open this URL in your default browser if it points to a website. If it is a path to a file then AceText can open the file in the application that is associated with that file's type on your computer. There are several ways to do this. The URL field in the AceText Editor highlights valid URLs. Double-clicking a highlighted URL opens it. You can also right-click the URL field and select Browse to URL. The same Browse to URL command is also available in the Clip menu. Its keyboard shortcut is Ctrl+E.

AceText can also highlight URLs within the body text of a clip. For a clips that use a syntax coloring scheme, it depends on the syntax coloring scheme whether the clip is highlighted. For clips that don't use a syntax coloring scheme, including clips with emphasis, you can choose in the Appearance Preferences whether URLs should be highlighted. When URLs are highlighted in a clip's text, you can open them by double-clicking them.

7. Edit Menu

The Edit menu in the main menu lists basic editing commands that you find in the Edit menu in many applications. Most of these commands have buttons on the collection toolbar. Those are described in the collection toolbar reference section of this manual.

 Undo

 Redo

 Cut Clip to Clipboard

 Copy Clip to Clipboard

 Paste as New Clip

 Copy Link

 Delete Clips

Copy Link

Select Copy Link in the Edit menu or right-click the Label field in the AceText Editor and select Copy Link to create an URL that points to the active clip or folder and put that URL on the clipboard. The copied URL will look like this:

```
acetext:clip/9vhfWqB/Collection+File.atc/Folder/Clip+Label
```

The `acetext:` scheme indicates that this is an URL to be handled by AceText. Within AceText, such URLs always work. When AceText is installed on your computer, other applications that can open URLs in their associated applications can also launch AceText using `acetext:` URLs because AceText registers itself as the protocol handler in Windows for the `acetext:` URL scheme.

An AceText URL that points to a clip or a clip folder has a path that always starts with `clip/`. This part of the URL is fixed and mandatory.

The second part of a path is the identifier code of the clip or clip folder that the URL points to. It consists of up to 11 ASCII letters, digits, hyphens, and/or underscores. The ID uniquely identifies a clip or folder within a single AceText collection file. The ID does not change when a clip is edited. When the clip is duplicated the duplicate gets a new ID. AceText enforces unique IDs within a single collection. It tries to keep IDs unique between collections too. But it cannot ensure uniqueness between collections because you may make a copy of an AceText collection file outside AceText.

The third part is the URL-encoded file name of the collection that contains the clip. Links generated by AceText only specify the file name. But you can include the full path to the collection if you make sure to URL-encode all the slashes in the file path.

The last part is the path to the clip within the collection. If the clip is inside a clip folder, then the path includes the URL-encoded labels of all the parent folders separated by forward slashes. The path ends with the URL-encoded label of the clip.

All the parts after `clip/` are optional. But you obviously need to specify at least one part for the URL to actually point to anything. If you omit the path to the clip or the file name and the path to the clip then you can omit trailing backslashes. But you cannot omit the backslashes that separate an omitted part from a following specified part.

AceText locates the clip pointed to by the URL as follows. If the URL specifies the full path to a collection then AceText opens that collection if it's not open yet and makes that collection the active collection. If the collection cannot be opened then the URL has no effect. If the URL specifies the file name of a collection and you have a collection open by that file name then AceText activates that collection. If no collection by that file name is open then the URL has no effect. If the URL does not specify a collection file at all then URL processing continues using the collection that was already active.

If the URL specifies a path within the collection using folder and clip labels then AceText tries to locate that path within the active collection. If the path resolves to an item in the active collection then that item is activated and URL processing is complete.

If the URL does not specify a path or if the path does not point to an item then AceText uses the identifier code to find the clip. It first searches through the active collection. If an item with that ID is found then it is activated. If not then AceText searches for the ID through all other open collections from left to right. If found AceText activates the item and the collection it is found in. Either way, URL processing is done.

As you can see, the clip label path and the clip identifier in the URL are redundant. AceText needs only one for the link to work. Using both increases the likelihood that links to clip continue to work when collections are edited. AceText gives priority to the readable part of the URL. The opaque ID is used as a fallback.

So the full link, as specified above, is the most reliable if "Collection File.atc" is already open. It will survive the .atc file being moved to a different location on disk or a drive letter change. If you need the link to work even when the collection is not yet open, specify the full path:

```
acetext:clip/9vhfWqB/C%3A%5CMy Documents%5CCollection+File.atc/Folder/Clip+Label
```

If you want to use a link to point to another clip in the same collection then you can omit the file name to allow the URL to survive a change in the collection's file name. The URL will then also continue to work if both the clip that contains the link and the clip that the link points to are moved to a different collection.

```
acetext:clip/9vhfWqB//Folder/Clip+Label
```

If you want your links to be really short, you can specify just the ID. The link will work as long as the collection that contains the target clip is open.

```
acetext:clip/9vhfWqB
```

8. Text Menu

The Text menu in the main menu lists various commands for editing the text of the active clip. The collection toolbar has a button with the same items in a drop-down menu.

 Duplicate Text into New Clip
 Move Text into New Clip

 Cut Text
 Copy Text
 Paste Text into Clip

 Insert Date & Time
 Insert Text File into Clip
 Save Text

 Print Text

Duplicate Text into New Clip

The Duplicate Text into New Clip item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar when the ClipHistory tab or an AceText Collection tab is active.

Select this item, or press Shift+Ctrl+D on the keyboard, to create a new clip, using the selected part of the text of the active clip as the text for the new clip. The new clip becomes the active clip.

See Editing Clips and Text to learn how to use AceText.

Move Text into New Clip

The Move Text into New Clip item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar, except when the Recycle Bin tab is active.

Select this item, or press Shift+Ctrl+M on the keyboard, to create a new clip, using the selected part of the text of the active clip as the text for the new clip. That selected part is then deleted from the original clip. The new clip becomes the active clip.

See Editing Clips and Text to learn how to use AceText.

Cut Text

The Cut Text item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar, except when the Recycle Bin tab is active.

Select the Cut Text item to place the selected part of the text of the active clip onto the Windows clipboard. That selected part is then deleted from the clip. You can also cut text by pressing Ctrl+X on the keyboard while editing the clip's text.

Use the  Cut Clip to Clipboard command instead if you want to cut entire clips.

See Editing Clips and Text to learn how to use AceText.

Copy Text

The Copy Text item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar, except when the Recycle Bin tab is active.

Select the Copy Text item to place the selected part of the text of the active clip onto the Windows clipboard. You can also copy text by pressing Ctrl+C on the keyboard while editing the clip's text. If you want to copy the entire clip, use the Copy to Clipboard button instead.

Use the  Copy Clip to Clipboard command instead if you want to cut entire clips.

See Editing Clips and Text to learn how to use AceText.

Paste Text into Clip

The Paste Text into Clip item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar, except when the Recycle Bin tab is active.

Select the Paste Text into Clip item insert the text on the clipboard into the text of the active clip. You can also paste text into a clip by pressing Ctrl+V on the keyboard while editing the clip's text. If you want to create a new clip, use the Paste as New Clip button instead.

See Editing Clips and Text to learn how to use AceText.

Insert Date & Time

The Insert Date & Time button is available on the collection toolbar when the ClipHistory tab, the All Collections tab, or an AceText Collection tab is active. Click the button to insert the current date and time into the active clip using the date and time format that you most recently used. If you've never used this command then it uses the short date and time format as configured in Windows.

The toolbar button has a drop-down list that keeps a history of date and time formats you've used. Select one to insert the current date and/or time using that format. Select Other Date & Time Format at the top of the menu to show a dialog box to enter a new date and time format using the following specifiers. Format specifiers may be written in uppercase as well as in lowercase letters. Both produce the same result.

Specifier	Description
-----------	-------------

c	Displays the date in short format followed by the time in long format as specified in the regional settings in the Windows Control Panel.
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) in the language of the current Windows locale.
dddd	Displays the day as a full name (Sunday-Saturday) in the language of the current Windows locale.
dddddd	Displays the date using the short date format specified in the regional settings in the Windows Control Panel.
ddddddd	Displays the date using the long date format specified in the regional settings in the Windows Control Panel.
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) in the language of the current Windows locale.
mmmm	Displays the month as a full name (January-December) in the language of the current Windows locale.
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).
z	Displays the millisecond without a leading zero (0-999).
zzz	Displays the millisecond with a leading zero (000-999).
t	Displays the time using the short time format specified in the regional settings in the Windows Control Panel.
tt	Displays the time using the long time format specified in the regional settings in the Windows Control Panel.
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.
/	Displays the date separator character specified in the regional settings in the Windows Control Panel.
:	Displays the time separator character specified in the regional settings in the Windows Control Panel.

'xx' / "xx" Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

Insert Text File into Clip

The Insert Text File into Clip item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar, except when the Recycle Bin tab is active.

Select the Insert Text File into Clip item to read the contents of a file, and insert those into the text of the active clip. If you want to create a new clip from a file, use the  Create Clip from Text File button instead.

AceText can only read plain text files, such as text documents (.txt), HTML files (.html), XML data (.xml), software source code, etc. AceText cannot decode proprietary file formats, such as Microsoft Word documents (.doc), Excel spreadsheets (*.xls) or Acrobat PDF files (*.pdf). To transfer text from such files into AceText, open the file in its corresponding application, and then use copy and paste or drag and drop between that application and AceText. The software will then decode its own proprietary file format, and offer the contents as plain text to AceText.

See Editing Clips and Text to learn how to use AceText.

Save Text

The Save Text item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar, except when the Recycle Bin tab is active.

Select the Save Text item to store the selected part of the text of the active clip into a text file.

If you want to save multiple clips into one or more text files, use the  Export Collection item in the Collection menu instead.

See Editing Clips and Text to learn how to use AceText.

Print Text

The Print Text item is available in the Text menu and via the Insert and Extract Text button on the collection toolbar, except when the Recycle Bin tab is active.

Select the Print Text item to print the text of the active clip. The difference between using the Print Text item and the  Print Collection button on the main toolbar is that the Print Text item only prints the text. It doesn't print the clip's label and other information. Instead, the Print Text item makes additional headers available when you click the Headers button in the print preview. These include the clip's label, date, URL and containing folder.

AceText automatically remembers the print options between printouts. The headers and footers you select are remembered separately for the Print Text and Print Collection commands.

The Print Text command shows the same Print Preview as the  Print Collection command, with one difference. The “What to Print” button is replaced with a “Selection Only” checkbox. Tick the checkbox to print only the selected lines in the clip. Partially selected lines are printed entirely.

9. Convert Menu

The Convert menu in the main menu contains various commands for converting between different kinds of characters. The commands in the four submenus are also available in the drop-down menus of some of the commands on the collection toolbar.

 Case
 Quotes
 Tabs
 Characters

ROT-13

Convert | Case

The Case submenu of the Convert menu has several commands for adjusting between uppercase and lowercase letters. One of these commands, Auto Adjust Case, has a button directly on the collection toolbar, except when the Recycle Bin tab is active. This toolbar button has a drop-down menu that lists the same commands as the Convert | Case submenu.

Auto Adjust Case is a toggle that automatically adjusts the case of future text you type. The other commands do an immediate one-time conversion on the selected text. If you didn't select any text then the conversion is applied to the line that the cursor is on. You can use the case conversion commands on the Label and URL fields of clips and folders as well as on the body text of clips.

See Editing Clips and Text to learn how to use AceText.

Convert | Case | Adjust Case

The Adjust Case command is only available if the clip uses a syntax coloring scheme that applies case conversion rules to specific parts of the text. The HTML scheme included with AceText, for example, specifies that HTML tags should be lowercase.

Click the Adjust Case command to convert the text you selected in the clip's contents according to the syntax coloring scheme's case conversion rules. If there is no selection, the command converts the case of the line that the cursor is on.

Convert | Case | Auto Adjust Case

Auto Adjust Case is a toggle that affects all future text you enter via the keyboard into the contents of all clips. This lasts until you turn off Auto Adjust Case. It only has an actual effect on clips for which you selected a syntax coloring scheme with case conversion rules before you entered the text.

When Auto Adjust Case is on the case conversion rules are applied automatically to any text you enter when you move the cursor away from the line you entered the text on. This delay ensures that Auto Adjust Case

does not incorrectly adjust partially entered words. The Delphi scheme, for example, converts keywords such as “begin” to lowercase. But if you enter “Begin” as part of a longer identifier such as “BeginOperation()” then no conversion should take place. By holding off the conversion until you move the cursor away from the line, AceText ensures you do not end up with “beginOperation()”.

Convert | Case | Uppercase

Convert all lowercase letters to uppercase letters.

Convert | Case | Lowercase

Convert all uppercase letters to lowercase letters.

Convert | Case | Initial Caps

Convert the first letter of each word to uppercase and all other letters to lowercase.

Convert | Case | Invert Case

Turns all uppercase letters into lowercase letters, and all lowercase letters into uppercase.

If you have been typing with Caps Lock on, you do not have to start over. Let this function come to the rescue.

Convert | Quotes

The Quotes submenu of the Convert menu lets you convert between straight quotes, smart quotes, and primes. Straight quotes are the quotes that you can type on your keyboard. Smart quotes are typographic quotes such as the high commas and turned high commas in English texts. They are called “smart quotes” because some applications are “smart” enough to automatically replace straight quotes with typographic quotes. AceText is particularly smart in this area as it supports many different quotation styles as well as primes. Primes are symbols used to indicate units such as feet, inches, minutes, and seconds.

The  Auto Adjust Quotes command can be toggled directly on the collection toolbar. This toolbar button has a drop-down menu that lists all the commands in the Convert | Quotes menu.

 Adjust Quotes

 Auto Adjust Quotes

Straight ⇒ Smart Quotes

Smart ⇒ Straight Quotes

Replace Smart Quotes

Quotes ⇒ Primes

Primes ⇒ Quotes

Quotation Style

Convert | Quotes | Adjust Quotes

A syntax coloring scheme can specify that certain straight quotes should be converted into smart quotes or that certain smart quotes should be converted into straight quotes. If the active clip uses such a syntax coloring scheme then you can use the Adjust Quotes command in the Quotes submenu of the Convert menu to convert the selected text according to the scheme's quote conversion rules. If there is no selection, the command converts the quotes on the line that the cursor is on.

The syntax coloring scheme specifies which straight quotes can be converted to smart quotes and which straight quotes that smart quotes can be converted into. This allows the syntax coloring scheme to ensure that characters that are part of the file's syntax are not converted. The HTML scheme, for example, specifies that straight single and double quotes within the body text should be converted to smart quotes. It does not allow any quote conversion in HTML tags because those use quotes to delimit attribute values. Thus when you use Adjust Quotes on an HTML clip, AceText converts the quotes that the reader will see into smart quotes, while leaving the quotes that are part of the HTML syntax alone.

AceText can convert single quotes, double quotes, backticks, and less-than and greater-than signs into smart quotes. It can convert smart quotes back to those characters. But the syntax coloring scheme can restrict the quote conversion to only some of these characters. The Delphi language, for example, uses single straight quotes to delimit strings. The Delphi syntax coloring scheme allows double straight quotes to be converted to smart quotes in strings. It does not allow single straight quotes to be converted because that would break the string. If you do want to have smart single quotes inside a Delphi string, you can select the single straight quotes you want to convert and use Convert | Quotes | Straight ⇒ Smart Quotes to force the conversion.

The syntax coloring scheme does not specify the style of smart quotes that should be used. You can choose that via Convert | Quotes | Quotation Style. It uses the settings in the same way as Convert | Quotes | Straight ⇒ Smart Quotes and Convert | Quotes | Smart ⇒ Straight Quotes. The settings are described in detail in those topics. If the syntax coloring scheme does not allow certain characters to be converted, then any options using them are implicitly turned off.

In the Quotation Style dialog you can also tick “replace smart quotes before adjusting quotes”. If you do then Adjust Quotes does what Convert | Quotes | Replace Smart Quotes does before converting between straight and smart quotes according to the syntax coloring scheme. This can be useful if you're pasting text that may contain straight quotes or smart quotes in a different style that you want to convert to your chosen smart quotation style. Then you need only use Adjust Quotes to do both conversions.

If you'd like the quote conversion based on the syntax coloring scheme to happen automatically as you enter text, turn on Convert | Quotes | Auto Adjust Quotes.

Convert | Quotes | Auto Adjust Quotes

The Auto Adjust Quotes item in the Quotes submenu of the Convert menu is a toggle that affects all future text you enter via the keyboard into the contents of all clips. This lasts until you turn off Auto Adjust Quotes.

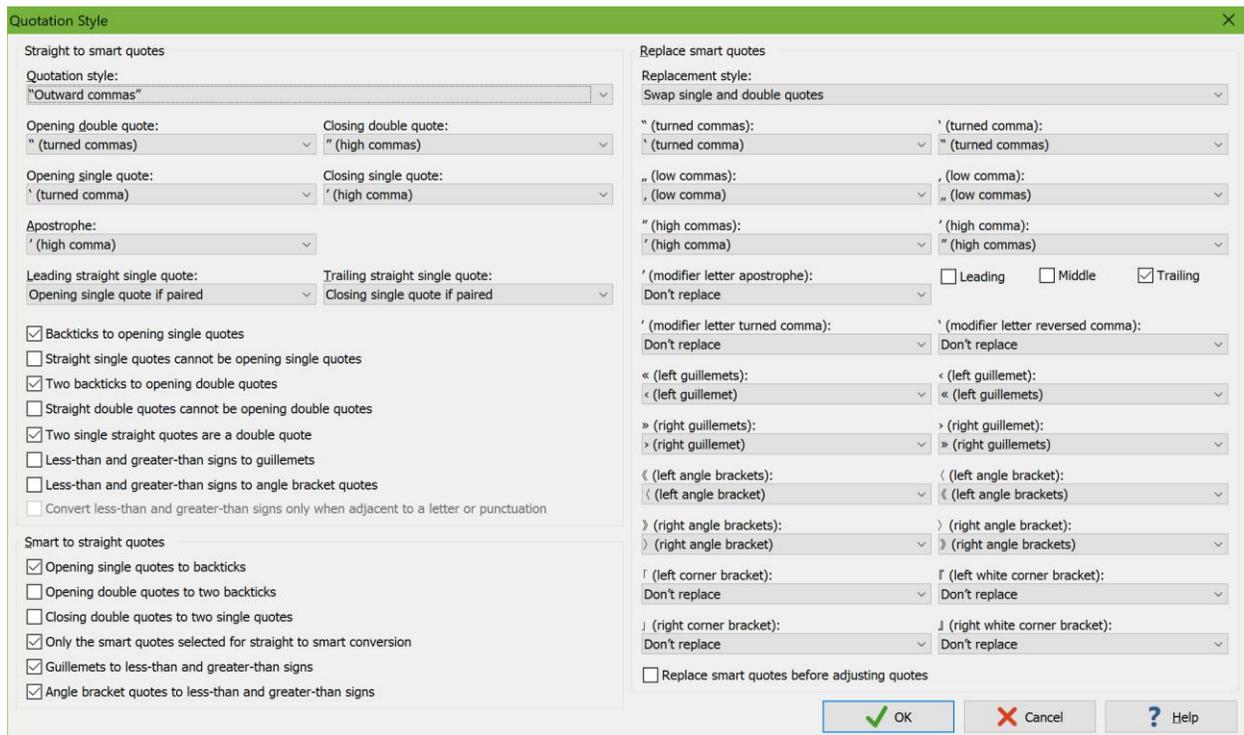
When Auto Adjust Quotes is on, quotes are automatically converted in any text that you enter when you move the cursor away from the line you entered the text on. This delay ensures that AceText can correctly distinguish between single quotes and apostrophes. It also ensures that the syntax coloring scheme can match up opening and closing quotes when the line is complete before any conversion is done.

If the clip uses a syntax coloring scheme then Auto Adjust Quotes uses the quote conversion rules of the syntax coloring scheme like the Adjust Quotes command does. If the clip does not use a syntax coloring scheme then the Auto Adjust Quotes converts straight quotes to smart quotes like the Straight \Rightarrow Smart Quotes command does.

Convert | Quotes | Straight \Rightarrow Smart Quotes

Select Straight \Rightarrow Smart Quotes in the Quotes submenu of the Convert menu to convert straight quotes in the selected text into smart quotes. If there is no selection, the quotes on the line that the cursor is on are converted.

Exactly which straight quotes are converted into exactly which smart quotes depends on the choices you made in Convert | Quotes | Quotation Style. The Straight \Rightarrow Smart Quotes command uses the settings in the “straight to smart quotes” group in the upper left corner of that dialog. It does not use any of the other settings.



Quotation Style

Straight to smart quotes

Quotation style: "Outward commas"

Opening double quote: " (turned commas) Closing double quote: " (high commas)

Opening single quote: ' (turned comma) Closing single quote: ' (high comma)

Apostrophe: ' (high comma)

Leading straight single quote: Opening single quote if paired Trailing straight single quote: Closing single quote if paired

Backticks to opening single quotes
 Straight single quotes cannot be opening single quotes
 Two backticks to opening double quotes
 Straight double quotes cannot be opening double quotes
 Two single straight quotes are a double quote
 Less-than and greater-than signs to guillemets
 Less-than and greater-than signs to angle bracket quotes
 Convert less-than and greater-than signs only when adjacent to a letter or punctuation

Smart to straight quotes

Opening single quotes to backticks
 Opening double quotes to two backticks
 Closing double quotes to two single quotes
 Only the smart quotes selected for straight to smart conversion
 Guillemets to less-than and greater-than signs
 Angle bracket quotes to less-than and greater-than signs

Replace smart quotes

Replacement style: Swap single and double quotes

" (turned commas): ' (turned comma) ' (turned comma): " (turned commas)

„ (low commas): ‚ (low comma) ‚ (low comma): „ (low commas)

" (high commas): ' (high comma) ' (high comma): " (high commas)

' (modifier letter apostrophe): Leading Middle Trailing
 Don't replace

' (modifier letter turned comma): ' (modifier letter reversed comma):
 Don't replace

« (left guillemets): < (left guillemet) < (left guillemet): « (left guillemets)

» (right guillemets): > (right guillemet) > (right guillemet): » (right guillemets)

« (left angle brackets): < (left angle bracket) < (left angle brackets)

» (right angle brackets): > (right angle bracket) > (right angle brackets)

⌈ (left corner bracket): ⌋ (left white corner bracket):
 Don't replace

⌋ (right corner bracket): ⌈ (right white corner bracket):
 Don't replace

Replace smart quotes before adjusting quotes

OK Cancel Help

The “**quotation style**” drop-down list is a speed setting. Selecting a style changes the 5 different quotation characters that you can choose. Pretty much all quotation styles used around the world are available in AceText. You can make a custom style by selecting quotation characters from the drop-down lists for double quotes, single quotes, and apostrophes.

AceText is smart enough to distinguish between apostrophes and single quotes. The sentence “In the ’90s we gave ’em hell.” has two apostrophes and zero single quotes. An apostrophe is not a quote, even if it looks like one, but a character that indicates something was omitted. In American typography, the apostrophe should look like a high comma or a tiny 9 (which is also used for closing single quotes). The apostrophe should not look like a turned comma or a tiny 6 (which is only used for opening single quotes). This is true despite the fact that word processors from major American software companies get this wrong.

If you enter the example sentence “In the '90s we gave 'em hell.” with two straight single quotes then AceText is smart enough to realize that they are not a pair of single quotes but rather two apostrophes. AceText only converts a single straight quote into an opening quote if there is a matching closing single quote. This is also the reason why Convert|Quotes|Auto Adjust Quotes converts quotes only after you’ve moved away from the line. AceText waits to see if any straight quote you enter will end up being paired or alone before deciding how to convert it.

To enable AceText’s automatic differentiation between single quotes and apostrophes, set “**leading straight single quote**” to “opening single quote if paired” and “**trailing straight single quote**” to “closing single quote if paired”. Then unpaired single quotes are converted into apostrophes. If your language has different rules for apostrophes or no apostrophes at all, you can change these to always be an apostrophe or always be an opening or closing single quote.

Exactly what your smart quotes look like also depends on the font you are using to display the file, of course. Some fonts render the turned comma (which looks like a tiny number 6) as a reversed comma (which looks like a mirror image of a tiny 9). But this does not affect the discussion of apostrophes versus opening single quotes.

In English text, the “high comma” character is normally used for both closing single quotes and for apostrophes. The Unicode standard recommends this. It is the default in AceText. One issue with this is that the high comma is a punctuation character. By default, AceText does not consider it to be part of a word. Double-clicking “isn’t” selects “isn” or “t” but not “isn’t”. You could change this by turning on the “other punctuation” option in the text layout configuration. But then any punctuation around a word is selected when you double-click it. The regex `\w` for matching a word character never matches the high comma.

If you prefer your apostrophes to be word characters, you can select “modifier letter apostrophe” as your smart quote. Most fonts display this character identically to the high comma. But it is a different character that is in the “modifier letter” Unicode category. This category contains various symbols that may not be letters but should act as letters. AceText always considers them to be part of a word. You will need to save your files as Unicode since the legacy Windows code pages do not support the modifier letter apostrophe.

Some languages don’t use apostrophes but other characters with a similar function. You can select these in AceText too. The “modifier letter turned comma” marks the phonemic glottal stop in many Polynesian languages, such as the ‘okina in Hawai‘i. The “modifier letter reversed comma” is used in languages such as Armenian.

In some circles it is customary to use backticks (`) as opening single quotes. On a US keyboard, the backtick can be found to the left of the 1 key in the upper left corner of the keyboard. If you use this style, check

“backticks to opening single quotes” to have AceText interpret backticks as opening quotes. Backticks are never interpreted as apostrophes. If you exclusively use backticks to enter opening quotes then you should check “**straight single quotes cannot be opening single quotes**”. Then AceText knows that a straight single quote is either an apostrophe or a closing quote.

There is a similar custom of using two backticks (``) as opening double quotes. Check “**two backticks to opening double quotes**” if you use this style. Also check “straight double quotes cannot be opening double quotes” if you use the style with double backticks exclusively.

People who use two backticks as opening double quotes sometimes use two single straight quotes (") rather than a double straight quote (") as closing double quotes. If you do this, check “**two single straight quotes are a double quote**”. If you don't, AceText interprets two single straight quotes as two single quotes.

You have two ways of adding «guillemets» or «angle brackets quotes» to your text files in AceText. One way is to set “quotation style” to one of the guillemet or angle bracket options. Then any straight quotes that you enter are converted into guillemets or angle bracket quotes. This can be useful when working with file formats like HTML where less-than and greater-than signs are part of the syntax and shouldn't be converted. The other way is to check “**less-than and greater-than signs to guillemets**” or “**less-than and greater-than signs to angle bracket quotes**”. Then you can enter < and > on the keyboard and have them converted into < and > or < and > . If your text may also include actual less-than and greater-than signs then check “**convert less-than and greater-than signs only when adjacent to a letter or punctuation**”. Then < and > are not converted when surrounded by spaces as they would typically be in a mathematical formula.

Do not confuse a guillemet with a guillemot. A guillemet is a quotation character. A guillemot is a species of bird.

Convert | Quotes | Smart ⇒ Straight Quotes

Select Smart ⇒ Straight Quotes in the Quotes submenu of the Convert menu to convert smart quotes in the selected text into straight quotes. If there is no selection, the quotes on the line that the cursor is on are converted.

Exactly which smart quotes are converted into exactly which straight quotes depends on the choices you made in Convert | Quotes | Quotation Style. The Smart ⇒ Straight Quotes command uses the settings in the “smart to straight quotes” group in the lower left corner of that dialog. Some of these settings refer to the five quotation characters that you can select in the “straight to smart quotes” group in the upper left corner. No settings other than those five quotation characters from the upper left group are used by Smart ⇒ Straight Quotes.

Check “**opening single quotes to backticks**” if you want opening single smart quotes to be converted to backticks (`) instead of single straight quotes. This produces the ASCII quotation style of using a backtick as the opening single quote and a straight quote as the closing single quote. This rule is applied to the character you select in the “opening single quote” drop-down list if you did not select the same character in the “closing single quote” drop-down list. The character you select in the “closing single quote” drop-down list is never converted into a backtick. The high comma (⁂) is always converted into a single straight quote if you selected it in the “apostrophe” drop-down list.

Other characters may be converted into backticks if you did not select them in either of the “opening single quote” and “closing single quote” drop-down lists. This always applies to turned commas (‘) and low commas (’). High commas (ˆ) are converted into backticks if you did not select the high comma in the “apostrophe” drop-down list. Single left guillemets (‹) and single left angle brackets (‹) are converted into backticks if you did not turn on the option to convert them into less-than signs.

Check “**opening double quotes to two backticks**” if you want opening double smart quotes to be converted to two backticks (``) instead of one double straight quote. This rule is applied to the character you select in the “opening double quote” drop-down list if you did not select the same character in the “closing double quote” drop-down list. The character you select in the “closing double quote” drop-down list is never converted into a backtick.

Other characters may be converted into two backticks if you did not select them in either of the “opening double quote” and “closing double quote” drop-down lists. This always applies to double turned commas (‘‘) and double low commas (’’). Double left guillemets (‹‹) and double left angle brackets (‹‹) are converted into two backticks if you did not turn on the option to convert them into less-than signs.

Checking “**closing double quotes to two single quotes**” converts closing double smart quotes into two single straight quotes instead of one double straight quote. This option is best used in combination with “opening double quotes to two backticks”. Together they produce the ASCII quotation style of using two backticks as opening quotes and two single straight quotes as closing quotes. Some people find this style more balanced than using two backticks for the opening quote and one double straight quote for the closing quote. With a monospaced font, two backticks or two single straight quotes are twice as wide as one double straight quote.

The character you select in the “closing double quote” is converted into two single straight quotes if you did not select the same character in the “opening double quote” drop-down list. If you select the same character in both drop-down lists then it is always converted into straight double quotes.

Other characters may be converted into two single straight quotes if you did not select them either of the “opening double quote” and “closing double quote” drop-down lists. This always applies to double high commas (ˆˆ). Double right guillemets (››) and double right angle brackets (››) are converted into two single straight quotes if you did not turn on the option to convert them into greater-than signs.

Check “**only the smart quotes selected for straight to smart conversion**” to convert only the characters that you selected in the five drop-down lists in the “straight to smart quotes” group into straight quotes. Otherwise, all of the characters that are available in those drop-down lists are converted into straight quotes. The sole exception are corner brackets. Those are only converted into straight quotes if you selected them in the drop-down lists.

Check “**guillemets to less-than and greater-than signs**” to convert left guillemets into less-than signs and right guillemets into greater-than signs. Double guillemets are converted into two less-than or greater-than signs. If you don’t check this option then guillemets may be converted into straight quotes and/or backticks or be left alone depending on the above options.

Check “**angle brackets to less-than and greater-than signs**” to convert left angle bracket quotes into less-than signs and right angle bracket quotes into greater-than signs. Double angle bracket quotes are converted into two less-than or greater-than signs. If you don’t check this option then angle bracket quotes may be converted into straight quotes and/or backticks or be left alone depending on the above options.

You can convert guillemets and angle bracket quotes into less-than and greater-than signs at the same time.

Convert | Quotes | Replace Smart Quotes

Select Replace Smart Quotes in the Quotes submenu of the Convert menu to convert the selected text from one style of smart quotes into another style of smart quotes. If there is no selection, the quotes on the line that the cursor is on are converted.

Before you can use this command, you need to use Convert|Quotes|Quotation Style to specify which smart quotes should be replaced with which other smart quotes. If you forget, the Replace Smart Quotes command pops up a message box to remind you. Replace Smart Quotes only uses the settings in the right-hand side of the Quotation Style dialog box.

The “replacement style” drop-down list is a speed setting. Selecting something in this drop-down list changes all of the other drop-down lists. The provided replacement styles all convert from one common smart quotation style to another smart quotation style.

You can replace specific quote characters with another quote character by selecting the replacement character from the drop-down list of the character you want to replace. You can select the same character from multiple drop-down lists to replace all those characters with the same character. You can select “don’t replace” to leave certain smart quotes alone.

The high comma (⁂) drop-down list is accompanied by three check boxes. Check “leading” to convert high commas when they occur before a word, “middle” when they occur within a word, and “trailing” when they occur after a word. This allows you to selectively replace high commas depending on whether they are used as quotation characters or as apostrophes. All other characters are replaced wherever they occur.

The option “replace smart quotes before adjusting quotes” is not used by Convert|Quotes|Replace Smart Quotes. That option only affects Convert|Quotes|Adjust Quotes and Convert|Quotes|Auto Adjust Quotes.

Convert | Quotes | Quotes ⇒ Primes

Select Quotes ⇒ Primes in the Quotes submenu of the Convert menu to convert all quotes in the selected text into primes. If there is no selection, the quotes on the line that the cursor is on are converted.

The prime (′) is the correct symbol to use to indicate units such as feet or arcminutes. The double prime (″) represents inches and arcseconds. Triple primes (‴) represent a ligne (1/12 of an inch) or a third (1/60 of a second). Quadruple primes represent a fourth (1/60 of a third). In mathematics, primes are used to indicate derivatives. In music, primes represent octaves. The middle C is c', while c'' is one octave above c' and c''' is two octaves higher.

5'8" (using primes) is the correct way to write 5 feet 8 inches. 5'8" is how you would enter 5 feet 8 inches on a keyboard. You'd have to leave it as that in a plain ASCII text file. 5'8" (using high commas) is nonsense produced by (automatic) straight to smart quote conversion.

Depending on the font you're using, the difference between a prime (′) and a high comma (⁂) can be subtle. The prime is straight while the high comma is curved like a little 9. Even if your font doesn't give high commas much curvature, you should take care to use the correct symbol. If you switch fonts later, the difference can be more glaring. Arial is a font that makes primes and quotes look rather different. If somebody uses Convert|Quotes|Replace Smart Quotes to change the quotation style later, high commas may be replaced but primes will stay as primes.

The Quotes ⇒ Primes command converts straight quotes, high commas (⁂ and ⁂), turned commas (‘ and ’), and all the characters you selected in the five “straight to smart quotes” drop-down lists in Convert|Quotes|Quotation Style into primes. This ensures that the straight quotes you enter on the keyboard are converted into primes, even if those straight quotes were already converted into smart quotes by Convert|Quotes|Auto Adjust Quotes.

Individual single quotes are converted into single primes. Two consecutive single quotes and double quotes are converted into double primes. Three consecutive single quotes, a single and double quote pair, and a double and single quote pair are converted into triple primes.

Quotes ⇒ Primes does not produce quadruple primes. Most fonts included with Windows do not support them. Two consecutive double quotes become two double primes. Four consecutive single quotes become a triple prime followed by a single prime.

Quotes ⇒ Primes converts backticks (`) into reverse primes. This only works if you did not allow Convert|Quotes|Auto Adjust Quotes to convert backticks into smart quotes first. Since Auto Adjust Quotes waits until you move away from the line you were editing, you can use Quotes ⇒ Primes immediately after entering the backticks to convert them into primes. Many fonts included with Windows do not support reverse primes, however.

Convert|Quotes|Primes ⇒ Quotes

Select Primes ⇒ Quotes in the Quotes submenu of the Convert menu to convert all primes in the selected text into quotes. If there is no selection, the primes on the line that the cursor is on are converted.

Single and double primes as well as single and double reverse primes are converted into smart quotes according to the settings in the “straight to smart quotes” group in Convert|Quotes|Quotation Style. A prime may be converted into an opening or a closing quote using the same rules and logic that Convert|Quotes|Straight ⇒ Smart Quotes uses to convert straight quotes into opening or closing quotes.

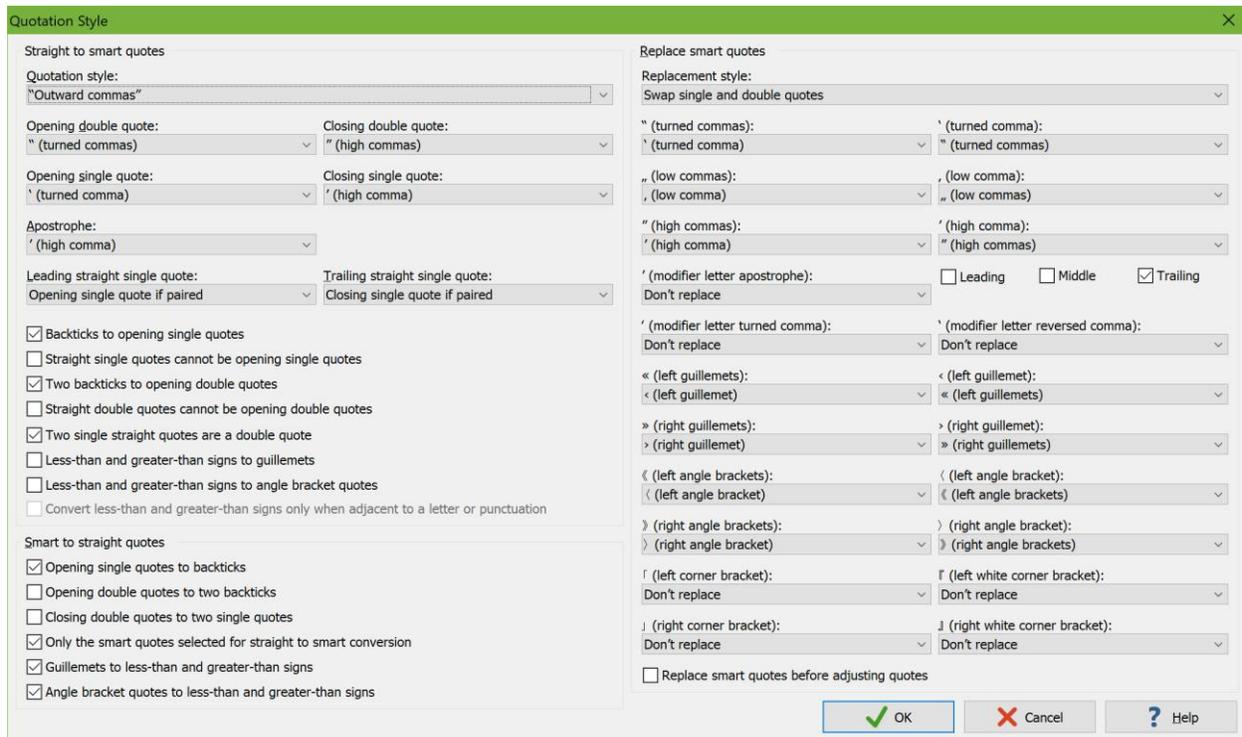
If you set the quotation style to “keep straight quotes” then Primes ⇒ Quotes converts single and double primes into straight single and double quotes. Reverse primes are then converted into backticks and reverse double primes into two backticks.

Triple primes are always converted into three straight single quotes. Quadruple primes are always converted into two straight double quotes. Reverse triple primes are always converted into three backticks.

Convert | Quotes | Quotation Style

Select Quotation Style in the Quotes submenu of the Convert menu to choose which quotation characters are converted by all the other items in the Convert|Quotes submenu. The choices in the “straight to smart quotes” group are explained in the Convert|Quotes|Straight ⇒ Smart Quotes topic. The options in the “smart to straight quotes” group are explained in the Convert|Quotes|Smart ⇒ Straight Quotes topic. The “replace smart quotes” group is explained in the Convert|Quotes|Replace Smart Quotes topic.

Whenever you change settings in the Quotation Style dialog, those settings are used for all future quote conversions in all files until you change the settings again.



Convert | Tabs

The Tabs submenu of the Convert menu has several commands for converting between spaces and tabs. The conversions use the tab size that you specified in the Appearance Preferences. These commands only have their desired effect when using a monospaced text layout. If you select some text first then only the selected lines are converted. If there is no selection then the conversion affects the entire clip.

Convert | Tabs | Indentation Spaces ⇒ Tabs

Convert as many spaces at the start of each line into tab characters as possible, without changing the layout of the clip. Spaces that occur after a non-whitespace character on a line are not converted.

Convert | Tabs | Spaces ⇒ Tabs

Convert as many spaces into tab characters as possible, without changing the layout of the file.

Convert | Tabs | Tabs ⇒ Spaces

Convert all tab characters in the active file into spaces.

Each tab in the file is replaced with the same number of spaces as the number of columns between the tab's position and the next tab stop. If the text is monospaced and thus the width of a space is the same as the width of any other character, then the result is that after the conversion, the layout of the text remains exactly the same. You would only see a difference when visualizing spaces.

You can also find the tab conversion items in the drop-down menu of the »• Visualize Spaces button on the collection toolbar.

Convert | Characters

The Characters submenu of the Convert menu has several commands for converting between actual characters and various escapes and entities representing those characters. The conversion can work on clip or folder labels, the body text of clips, or clip or folder URLs. The edit control that most recently had keyboard focus is the one that the conversion is applied to. If you select some text first then only the selected text is converted. If there is no selection then the conversion affects all the text in the edit control.

Convert | Characters | Characters ⇒ \uFFFF

Replace non-ASCII characters with escape sequences in the form of \uFFFF representing the Unicode code points of the characters.

Convert | Characters | \uFFFF ⇒ Characters

Replace escape sequences in the form of \uFFFF with their actual characters.

Convert | Characters | Characters ⇒

Replace non-ASCII characters with a decimal character references in the form of .

Convert | Characters | Characters ⇒ ÿ.

Replace non-ASCII characters with a hexadecimal character references in the form of ÿ.

Convert | Characters | `` and `` ⇒ Characters

Replace decimal and hexadecimal character references in the form of `` and `` with their actual characters.

This command does not convert numeric character references that represent the characters `<`, `>`, `'`, `"`, and `&` that have special meanings in XML. To convert these, use `Convert | Characters | XML Entities ⇒ Reserved Characters`.

Convert | Characters | Characters ⇒ `&htmlchar;`

Replace characters that have HTML entities with those HTML entities. It changes “après-ski” into “après-ski”, for example.

Characters that do not have HTML entities are not changed. If you want to convert your clip into pure ASCII using HTML entities and numeric character references for non-ASCII characters, first use the `Characters ⇒ &htmlchar;` command to replace those characters that have named HTML entities. Then use the `Characters ⇒ ` or `Characters ⇒ ` command to replace the remaining non-ASCII characters that do not have named HTML entities.

This command does not replace reserved characters such as `<`. Thus you can safely use this command on an entire HTML file without destroying its HTML tags. If you do want to replace reserved characters such as `<` with entities such as `<`, use the `Convert | Characters | Reserved Characters ⇒ XML Entities` menu item.

Convert | Characters | `&htmlchar;` ⇒ Characters

Replace named HTML entities with their actual characters.

This command does not replace numeric character references. Only named HTML entities are replaced. To replace both named HTML entities and numeric character references, use both the `&htmlchar; ⇒ Characters` command and the `Convert | Characters | and ⇒ Characters` command.

This command also does not replace HTML entities that represent reserved characters such as `<`. To replace those, use `Convert | Characters | XML Entities ⇒ Reserved Characters`.

Convert | Characters | Reserved Characters ⇒ XML Entities

Replace the characters `<`, `>`, `'`, `"`, and `&` that have special meanings in XML with their respective XML entities `<`, `>`, `'`, `"`, and `&`.

Convert | Characters | XML Entities ⇒ Reserved Characters

Replace XML entities <;, >;, ';, ";, and &; with the characters <, >, ', ", and & that they represent. Numeric character references that represent these 5 characters are also converted.

Convert | ROT-13

Select ROT-13 in the Convert menu to apply ROT-13 character rotation to the current selection. This makes the text unreadable. Applying ROT-13 again restores the original text.

Before the Internet went commercial, people had the good habit to use ROT-13 to conceal possible offensive postings in newsgroups and then add a warning message in plain text. This made sure that nobody would read the text by accident and be offended. If one was offended after applying ROT-13, he or she would only have himself or herself to blame for ignoring the warning.

10. Format Menu

The Format menu in the main menu lists various commands for reformatting the text of the active clip. The Reformat Text button on the collection toolbar lists the same items in a drop-down menu.

Reflow Paragraphs
Limit Line Length

Comment Block
Uncomment Block
Toggle Comment
Quote Block
Unquote Block

Sort Alphabetically A–Z
Sort Alphabetically Z–A
Sort Alphanumerically A–Z, 0–9
Sort Alphanumerically 9–0, Z–A
Reverse Lines

Trim Whitespace
Trim Whitespace from Paragraphs

Delete Duplicate Lines
Delete Blank Lines
Consolidate Blank Lines

Reflow Paragraphs

The Reflow Paragraphs item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

Reflow paragraphs deletes all single line breaks, and replaces them with a space. If two or more line breaks appear right after each other then those line breaks will remain.

In software that does not support automatic word wrapping, lines are terminated with a single line break, and paragraphs are terminated with two line breaks. Most email and newsgroup software also sends text this way, for compatibility with old email and newsgroup software. The problem is that if you want to edit the text afterward, the line breaks are not automatically adjusted, which ruins the layout of the text. AceText's Reflow Paragraphs command intelligently removes those superfluous line breaks, so the text will properly reflow and rewrap when you edit it in AceText or another application.

Almost all Windows applications support automatic word wrapping, and interpret single line breaks as the end of a paragraph. So does AceText. If you copied text with fixed line lengths, such as an email message, into AceText, use the Reflow Paragraphs command to be able to edit the text the way you would expect. If later you need to send the text back to an application that expects fixed line lengths, use the Limit Line Length command from the Reformat Text button's menu.

See Editing Clips and Text to learn how to use AceText.

Limit Line Length

The Limit Line Length item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

Almost all Windows applications support automatic word wrapping, so there is no need to insert a line break at the end of each line. The key advantage is that the text will automatically rewrap when you edit it. However, some older applications do not support automatic wrapping, or do not support text containing very long lines or paragraphs. Use the Limit Line Length command to make AceText insert line breaks as needed, to accommodate such applications.

AceText will ask you for the maximum number of characters that any line may contain. Longer lines will be broken up by line breaks. AceText will make sure that words are not split up, so some lines may end up shorter than the length you specified.

If you select the option “justify between both margins”, AceText will insert spaces into lines that are shorter than the number of characters you specified. The spaces will be inserted next to spaces that were already present in the original text. The inserted spaces will be spread out evenly across the length of the line. When the result is viewed with a fixed-pitch font such as Courier, the text will appear to be aligned to both the left and right margin. No spaces will be inserted into very short lines, since that would make them difficult to read. Such lines will remain short.

The Reflow Paragraphs item does the opposite, enabling you to comfortably edit text that was wrapped using line breaks, rather than automatic wrapping.

See Editing Clips and Text to learn how to use AceText.

Comment Block

The Comment Block item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

If the current clip is being syntax colored by a syntax coloring scheme that contains a character string for single-line comments then Comment Block automatically uses those characters. If there is no selection then the comment characters are inserted at the first non-whitespace character on the line that the cursor is on. If you’ve made a selection that flows along with the text then the comment characters are inserted at the first non-whitespace character on each the line. Essentially, indentation is preserved after lines are commented out. Blank lines are not commented out.

If you’ve made a rectangular selection then the comment characters are inserted on each line covered by the selection at the leftmost column in the selection, even if the line is blank.

If you prefer the comment characters to be at the start of the line or the left hand edge of the rectangular selection, use the Quote Block command instead.

If the syntax coloring scheme does not define characters for single-line comments, but does define characters for multi-line comments, then `Block|Comment` inserts the characters that start a multi-line comment in the same way that it inserts characters that start a single-line comment. It then additionally inserts the characters that end a multi-line comment at the end of each line or at the right-hand edge of the selection if it is rectangular.

If the current file type does not use a syntax coloring scheme, or the scheme does not define any characters to comment out something, then `Comment Block` asks you for the characters used to start a single-line comment. Those are then used in the same way as single-line comment characters from a single coloring scheme. You can tick a checkbox to make `AceText` remember the comment characters you provided. It does so separately for each collection while you have the collection open.

If only one line was selected or if there was no selection then `Comment Block` moves the text cursor to the next line. This allows you to quickly comment out multiple lines without having to select them.

To remove the comment characters, use `Uncomment Block` or `Toggle Comment`.

You can use `Comment Block` on lines that are already commented out. Those are then doubly commented out. This allows you to comment out a large block in which some lines are commented out and then uncomment the whole block with `Uncomment Block`. The lines that were originally commented out will still be commented out.

Uncomment Block

The `Uncomment Block` item is available via the `Reformat Text` button on the collection toolbar and via the `Format` menu, except when the `Recycle Bin` tab is active.

`Uncomment Block` removes the comment characters added by `Comment Block`. When you've made a selection that flows with the text, `Block|Uncomment` skips over any whitespace at the start of each selected line to look for comment characters to remove. Comment characters after the first non-whitespace character on a line are left in place. When you've made a rectangular selection, `Uncomment Block` only removes the comment characters if they start at the left-hand edge of the selection and are completely covered by the selection. Comment characters that start at another column or that are not or only partially selected are left in place.

If the clip's syntax coloring scheme defines characters that start and end multi-line comments then `Block|Uncomment` removes the characters that start a multi-line comment in the same way that it removes characters that start a single-line comment. On each line where it managed to remove the start of a comment it then also tries to remove the characters that end the comment. For a linear selection, `Uncomment Block` skips over any whitespace at the end of the line to look for comment characters to remove. For a rectangular selection, `Uncomment Block` only removes the characters that end comments if those characters end at the right-hand edge of the selection and are completely covered by the selection.

Though `Uncomment Block` requires you to be a bit precise when selecting the block to be uncommented, this makes it possible to comment out blocks that already have comments in them, and then uncomment those blocks while leaving the original comments.

If only one line was selected or if there was no selection then Uncomment Block moves the text cursor to the next line. It does this even if the current line had no comment characters to remove. This allows you to quickly uncomment multiple lines without having to select them.

Toggle Comment

The Toggle Comment item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active. It combines the functionality of Comment Block and Uncomment Block.

On each line in the selection, Toggle Comment attempts to remove the comment characters exactly like Uncomment Block does. If it does not remove any comment characters from a particular line then it adds comment characters to that line just like Comment Block does.

If only one line was selected or if there was no selection then Block|Toggle Comment moves the text cursor to the next line. This allows you to quickly comment and uncomment multiple lines without having to select them simply by pressing the Ctrl+ / keyboard shortcut.

The advantage of Toggle Comment is that it allows you to use the same keyboard shortcut for commenting and uncommenting lines. But if you need to comment out a larger block in which some lines are already commented out that need to stay commented out, then you need to use Comment Block instead.

Quote Block

The Quote Block item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

You can use the Quote Block command to quote a block of text by inserting certain characters at the start of each line in the selection. In email, for example, it is common to quote excerpts from the email you're responding to by prefixing them with > . When you select the Quote Block command, AceText asks you for the characters used to quote the block. AceText then inserts those characters at the start of each line in the selection. Blank lines in the selection are also quoted. If you've made a rectangular selection then the comment characters are inserted on each line covered by the selection at the leftmost column in the selection. You can tick a checkbox to make AceText remember the comment characters you provided. It does so separately for each collection while you have the collection open.

You can use Quote Block on lines that are already quoted. Those are then doubly quoted.

When  Word Wrap is on, only paragraphs are quoted. Quote Block does not insert quotation characters in front of wrapped lines. This way you don't end up with quotation characters in the middle of lines when the lines are rewrapped when resizing AceText. You can use the Limit Line Length command to first insert line breaks at wrapping positions and then use Quote Block to quote all the lines.

See Editing Clips and Text to learn how to use AceText.

Unquote Block

The Unquote Block item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

In email messages it is common to quote previous messages by prefixing each line of the previous message with a special character, typically >. If you captured some text from a quoted message into AceText, you can instantly remove the > and similar characters and spaces from the start of each line with the Unquote Block command.

When you select the Unquote Block command, AceText asks you for the characters used to quote the block. AceText then removes those characters from the start of each line in the selection. Lines that don't start with the quotation characters are left untouched. If you've made a rectangular selection then AceText removes the quotation characters if they appear right at the left hand edge of the selection. You can tick a checkbox to make AceText remember the comment characters you provided. It does so separately for each collection while you have the collection open.

When a line is quoted multiple times, as you may get in email when multiple replies quote all the previous messages, Unquote Block removes only one quotation character from the start of that line. You can repeat the Unquote Block command if you want to remove further levels of quotation.

After unquoting a block you can use the Reflow Paragraphs command so the text will automatically rewrap when you edit it.

See Editing Clips and Text to learn how to use AceText.

Sort Alphabetically or Alphanumerically

Four commands for sorting lines are available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

If no part of the active clip has been selected then the entire clip is sorted alphabetically, paragraph by paragraph.

To sort the entire clip on a specific column of text, select that column of text before sorting. Make sure that the selection does not span more than one paragraph. All the paragraphs in the file are sorted, using the selected column as the sort key.

If the selection spans more than one paragraph, then the selected paragraphs are sorted. If you want to sort only a selected number of paragraphs on a certain column then make a rectangular selection first. The paragraphs that are partially covered by the rectangular selection are sorted entirely as if the selection was a normal one. However, the sort order is determined only by the text covered by the selection.

Alphabetic Sort works on a character by character basis, even for numbers. "A19" sorts before "A4" because 1 sorts before 4.

Alphanumeric Sort orders numbers logically. When the text contains numbers that consist of multiple digits, the numbers are compared as a whole. So “A4” sorts before “A19” because 4 is less than 19. This sort order is also known as “natural sort”.

See Editing Clips and Text to learn how to use AceText.

Reverse Lines

The Reverse Lines item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

This command reverses the lines in the active clip. It swaps the first line with the last line, the second line with the penultimate line, and so on. If you selected multiple lines in the clip, then only the selected lines are reversed.

See Editing Clips and Text to learn how to use AceText.

Trim Whitespace

The Trim Whitespace item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

Select the Trim Whitespace command or press Ctrl+T on the keyboard to remove any whitespace from the start and the end of the active clip’s text. Spaces, tabs and line breaks are all considered whitespace.

When selecting text in a document or a web page, it takes quite a bit of effort to make sure no superfluous whitespace characters end up in the selection. When using AceText, you don’t have to worry about this. Just capture the text into AceText, and press Ctrl+T to get rid of the additional whitespace, if any.

To remove extraneous whitespace at the start and the end of each line, use Trim Whitespace from Paragraphs instead of or in addition to Trim Whitespace.

See Editing Clips and Text to learn how to use AceText.

Trim Whitespace from Paragraphs

The Trim Whitespace from Paragraphs item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active.

Select the Trim Whitespace from Paragraphs command or press Shift+Ctrl+T on the keyboard to remove any whitespace from the start and the end of each paragraph in the active clip’s text. Spaces and tabs are considered whitespace.

If you captured some text that is indented, use Trim Whitespace from Paragraphs to remove all the indentation. If you want to unindent the text only one or a few steps, preserving remaining indentation, select the text and then press Shift+Tab on the keyboard.

To remove empty paragraphs from the start and the end of the clip, use Trim Whitespace in addition to or instead of Trim Whitespace from Paragraphs.

See Editing Clips and Text to learn how to use AceText.

Delete Duplicate Lines

The Delete Duplicate Lines item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active. Use it to delete duplicate lines from the active clip. You can make a number of choices as to what AceText considers to be a duplicate line.

Scope

If you've selected part of the clip before using the Delete Duplicate Lines command then you can limit the command to delete only lines that are selected. If the first and/or last line in the selection are only partially selected then the selection is expanded to include them entirely. If the selection is rectangular then lines covered by the selection are deleted entirely.

Proximity of Duplicate Lines

Select "anywhere in the scope" to delete all lines that are duplicated anywhere. The first copy of the line remains, while all the others are deleted. If you've set the scope to "selected lines", the lines must be duplicated inside the selection. Lines that are not duplicated inside the selection are not deleted, even if they have lines outside the selection.

Select "adjacent lines only" if you only want to delete a line's duplicates if they're immediately below the line they duplicate, without any other lines between them. If the clip's lines are sorted alphabetically, then the end result of "anywhere in the scope" and "adjacent lines only" is the same. In a sorted clip, all duplicates are sorted together. However, selecting "adjacent lines only" deletes the duplicate lines significantly faster, certainly when the number of lines in the clip is large. If you select "anywhere in the scope", AceText has to compare each line with every other line in the clip.

Comparison Options

By turning on one or more comparison options, you can tell AceText to consider lines as duplicates even when they aren't identical.

The "compare selected columns only" option is only available when you've made a selection that does not span more than one line, or when you've made a rectangular selection. With this option, AceText only compares the selected columns. If the selection spans from column 10 to column 20, for example, then AceText compares columns 10 through 20 of each line. If a line has less than 10 characters it is considered to be blank. This has important consequences (see next section).

“Ignore differences in leading spaces and tabs” treats lines that only differ in the number of spaces and tabs at the start of the line as duplicates. Similarly, “ignore trailing spaces and tabs” ignores differences in spaces and tabs at the end of each line. “Ignore all differences in spaces and tab” is more than a combination of the two previous options. AceText then completely ignores all spaces and tabs, including spaces and tabs in the middle of lines.

“Ignore difference in case” compares lines without regard to the difference between uppercase and lowercase letters.

Lines to Delete

You must select one or two choices in the “lines to delete” section. Every line in the clip belongs to one of the 3 categories. Selecting none of the options would have no effect, and selecting all of them would delete all the lines in the clip.

Turn on “2nd and following occurrences of duplicate lines” and turn off the other two options to delete all duplicate lines, leaving only unique clips in the clip, regardless of whether they were previously unique. Use this to delete unnecessary duplicates from a clip.

Turn on both “2nd...” and “1st occurrence of duplicate lines” to delete all duplicate lines, leaving only lines that were previously unique.

Turn on both “2nd...” and “non-duplicated lines” to leave only one copy of all lines that had duplicates. If you paste the contents of two lists that consist of unique lines (when viewed separately) into a clip in AceText, then you can use this combination to get the lines that occurred in both clips, but not the lines that occurred in only one of the clips.

If you want to keep only lines that occur a certain number of times, use the Delete Duplicate Lines several times. E.g. if you only want lines that occur 3 times or more, use it twice with the “1st occurrence...” and “non-duplicated...” options turned on. Then use it again with the “2nd occurrence...” and “non-duplicated...” options. The first time you delete the lines that occur only once, the second time you delete lines that occur only twice, and the third time you delete the duplicates of lines that occur four times or more.

Blank Lines

Since blank lines are technically all duplicates of each other, AceText offers you an extra choice for blank lines. You can choose to either delete all blank lines, not to delete any blank lines, or to only delete duplicate blank lines. The “duplicate blank lines” option takes into account the “proximity” setting, deleting either all but the first blank lines (“anywhere in the scope”), or only replacing subsequent blank lines with a single blank line (“adjacent lines only”).

If you’ve turned on the “compare selected columns only” option, a line may be considered blank even when it isn’t. If a line is shorter than the leftmost column in the selection, it is considered to be blank, even if it does have text on it.

Lines with only spaces and tabs on them are only considered to be blank if you’ve turned one of the options to ignore differences in spaces and tabs. On a line with only spaces and tabs, all spaces and tabs are considered to be both leading and trailing at the same time.

Delete Blank Lines

The Delete Blank Lines item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active. It deletes all totally blank lines from the file. Only lines that have no characters at all are deleted. Lines that consist solely of whitespace are not deleted.

If you want to delete lines that have whitespace too, first use Trim Whitespace from Paragraphs to remove the whitespace, and then use Delete Blank Lines.

See Editing Clips and Text to learn how to use AceText.

Consolidate Blank Lines

The Consolidate Blank Lines item is available via the Reformat Text button on the collection toolbar and via the Format menu, except when the Recycle Bin tab is active. It replaces all blocks of consecutive totally blank lines with a single blank line. Only lines that have no characters at all are consolidated. Lines that consist solely of whitespace are not consolidated.

If you want to consolidate lines that have whitespace too, first use Trim Whitespace from Paragraphs to remove the whitespace, and then use Consolidate Blank Lines.

11. Options Menu

The Options menu in the main menu lists various options that you can toggle at any time to adjust AceText's behavior to match what you're trying to accomplish. Many of these options have buttons on the main toolbar. Those options are described in the main toolbar reference section of this manual.

-  AceText Editor
-  AceText Tower
- Focus Editor
- Focus Clip Tree
-  Always Visible

-  Live Spell Check
-  Word Wrap
-  Indent Wrapped Lines
-  Right-to-Left
-  Persistent Selections

-  Line Numbers
-  Visualize Spaces
-  Visualize Line Breaks

- Dark Theme
-  Preferences

The Preferences item gives you access to the Preferences screen, which provides a wide range of configuration options. It has its own reference section in this manual.

Indent Wrapped Lines

You can find the Indent Wrapped Lines option in the Options menu.

When working with text like source code where indentation is important, you can turn on Indent Wrapped Lines to maintain indentation when  Word Wrap is enabled.

Line Numbers

You can find the Line Numbers option in the Options menu. You can toggle it to show or hide line numbers in the margin of the contents of clips. When  Word Wrap is enabled, all lines, including wrapped lines, are numbered.

Dark Theme

Select Dark Them in the Options menu to toggle AceText's user interface (toolbars, menus, dialog boxes, etc.) between the default Windows theme (which is a light theme) and AceText's own dark theme.

If you regularly switch between themes you should select color palettes that have companion palettes in the Appearance Preferences. Then switching between themes also switches between light and dark palettes.

12. Help Menu

Via the help menu in the main menu, you can access the help file that ships with AceText. It also provides links to the AceText web site with up to date information.

The User Forum is a great place to discuss AceText with other AceText users. It's also the best way to request technical support from AceText's developer and publisher Just Great Software. The rightmost button on the main toolbar gives you direct access to the forum.

The Share Collections item allows you to download AceText collections shared by other people, and to share your own.

The Create Portable Installation item copies AceText to a USB stick or other removable device, so you can carry AceText with you wherever you go, and run it on any computer without installing it.

Share Collections

Select Share Collections in the Help menu to download AceText collections shared by others, or to upload your own collections. A new window will pop up. AceText will connect to the Internet immediately to download the list of available collections. If your computer can only access the internet via a proxy server, click the Settings button in the lower left corner of the screen to enter your proxy server's settings.

The Download tab shows all available collections, including the ones you uploaded. The list shows the collection's title, the file name used by the author, how many clips it contains, who shared it, when it was first shared and when it was last updated. Click on a collection to get a more detailed description, and the author's name and email address (if provided by the author). Click the Download button to download the selected scheme.

The Share tab shows the collections that you previously uploaded. AceText automatically uses your AceText license to identify you. You can check your license details in the about box. To share a collection, select its file, and enter a title and your name. You can also enter your email address and web site URL if you want. In the Description box, explain what your collection is about. Click the Upload button to upload it. If you select a collection you previously uploaded, AceText will fill out all the fields with what you entered when you uploaded that collection. This way you can easily update a previously shared collection, or use the same details for another one. If you upload a collection with the same file name as a collection you already shared, the newly uploaded collection will replace the old one. If you upload a collection with another file name, it will be added as a new collection, even if you give it the same title or description as another one. Click the Delete button to stop sharing a collection.

AceText downloads and uploads in the background. The Progress tab shows all collections in the queue, and recently completed transfers. If an error occurs, AceText will not pop up a message box. The Progress tab will indicate the error.

Create Portable Installation

The Create Portable Installation item in the Help menu makes it easy to create a portable copy of AceText on a removable disk, USB stick or flash memory card. You can run AceText off the removable device on any computer, without leaving any traces of AceText on that computer.

If the capacity of the device you want to install AceText onto is limited, you can choose not to install certain parts of AceText on the device. The only part that is not optional is the “main application” part, which is AceText itself. You can select if you want to copy over the help file and the spell check dictionaries.

Note that the Create Portable Installation command will not delete any files from the device. If you use the Install onto Removable Drive command a second time, and select fewer parts than the first time, AceText will *not* remove the deselected parts from the device.

AceText will show you the complete list of devices that Windows reports as removable devices. External hard disks often report themselves as being hard disks rather than being removable devices. If you want to install AceText in a portable manner on such a device, turn on the option to treat all drives as removable drives. AceText will then create a file RemovableDrive.sys on the destination drive. This file act as a token to tell AceText not to touch the host computer.

To proceed with the installation, click on the device you want to install onto, and click the Install button. The label showing the amount of disk space needed will indicate the parts being copied to the removable drive. Once the operation is complete, the Create Portable Installation screen will close automatically.

AceText will copy itself to a fixed AceText subfolder off the root on the device. If you like more complex folder structures, you can move the complete AceText folder into a different parent folder on the device.

Creating Portable Installations using The Installer

If you have not yet installed AceText onto your hard disk, you can create a portable installation by running AceText’s installer. On the welcome screen, click the Portable Installation button. Follow the steps. This method works even if you do not have the necessary permissions to install software onto the computer you’re using.

If you run the installer on 32-bit Windows the portable install will be 32-bit. It will work on both 32-bit and 64-bit Windows. If you run the installer on 64-bit Windows, then the portable install will be 64-bit. It will only work on 64-bit Windows. You can pass the /32 command line parameter to the installer to force it to create a 32-bit installation on 64-bit Windows.

13. AceText Tray Icon

When AceText is running, it displays its icon in the system tray. The system tray is the area in the taskbar that also displays the system clock. If AceText is running and you don't see the icon, click the < button next to the icons you see next to the system clock. This reveals all the icons that Windows is hiding. Use the mouse to drag AceText's icon onto the taskbar to make it permanently visible while AceText is running.

Windows 11 hides both AceText's tray icon and its ^ button to reveal the icons by default. If you previously made the ^ button visible then you can click it to reveal all the icons that Windows 11 is hiding and drag AceText's icon onto the taskbar to make it permanently visible. If not, right-click the taskbar and select Taskbar Settings. This opens the Personalization > Taskbar section in the Settings app in Windows 11. Among the settings on the right hand side, expand the "other system tray icons" group. AceText's icon should be in the list when it is running. Toggle the Off button to On to make it visible on the taskbar. You can toggle "hidden icon menu" to On to reveal the ^ button to be able to directly drag tray icons on and off the taskbar in the future.

The icon indicates the mode that AceText is in:

 AceText is running, but is not in a special mode.

 This icon flashes once when you use the  Copy Clip to Clipboard or  Copy Text command.

 This icon flashes once to indicate the ClipHistory has captured the text that another application just copied to the clipboard.

 This icon flashes once to indicate the ClipHistory captured text that another application just copied to the clipboard and merged it with the previously captured clip.

 This icon flashes once to indicate the ClipHistory captured text that another application just copied to the clipboard and split it into multiple clips.

 This icon flashes once when you press the AcePure hotkey to indicate that AceText removed all clipboard formats except plain text from the Windows clipboard..

 AcePaste mode. Double-clicking a clip sends it to the application from which you activated AcePaste. Press the Escape key on the keyboard to cancel AcePaste.

 AceType mode. Type in an AceType abbreviation to send a clip to the application from which you activated AceType. Press the Escape key on the keyboard to cancel AceType.

 AceEdit mode. Press Ctrl+Enter or the AceEdit hotkey to send the text you're editing on the Clipboard tab back to the originating application. Press the Escape key on the keyboard to cancel AceEdit.

 SequencePaste Up mode. Pressing the AcePaste hotkey automatically pastes the active clip, and activate the clip above it in the collection.

 SequencePaste Down mode. Pressing the AcePaste hotkey automatically pastes the active clip, and activate the clip below it in the collection.

Click the AceText tray icon with the left mouse button to make AceText appear. Click it with the right mouse button to show the tray icon menu. The menu offers the following commands:

 Restore AceText

 New Collection

 Open Collection

 AcePaste

 AceType

 AceEdit
 AcePure

 AceText Editor
 AceText Tower

 Shut Down AceText

Restore AceText

You can make AceText visible again with a single left-click on AceText’s icon next to the system clock. If you right-click the icon you’ll get a popup menu with a “Restore AceText” item at the top. This item is bold to remind you that it’s the default action when you left-click the icon. You can click the menu item to restore AceText too.

If you prefer to use the keyboard, you can configure a hotkey to make AceText appear. The Restore AceText item at the top of the right-click menu of AceText’s tray icon reminds you of the present hotkey, if one is enabled.

AcePaste

AcePaste is a special AceText feature that enables you to quickly paste a clip into an application, and continue working with that application after pasting. The default hotkey is Win+Alt+V. When you press this hotkey when working with an application other than AceText, AceText will pop up. If you like to use the mouse, right-click on the AceText tray icon and select AcePaste from the menu.

After activating AcePaste, select the clip you want to paste. Do so either by double-clicking on it, by pressing Ctrl+Enter, or pressing the AcePaste hotkey again. If the clip contains parameters, AceText substitutes them and prompts if needed. The clip is then pasted directly into the application that was active when you initiated AcePaste. If you change your mind, press the Escape key to cancel AcePaste. Either way, AceText resurfaces the application that was active when you initiated AcePaste, so you can continue working straight away.

To rapidly paste a series of clips, activate SequencePaste. When SequencePaste is active, AcePaste does not pop up AceText for you to select the clip to paste. Rather, the active clip is pasted automatically, and the next clip activated. Each time you press the AcePaste hotkey, the next clip is pasted.

By default, AceText tries to paste into the target application by placing the clip on the clipboard and simulating the Ctrl+V key combination. If you want to use AcePaste with an application that pastes differently, or can only accept keyboard input, you can configure that application in the Applications Preferences.

See Look up and Reuse Text to learn how to use AceText.

AceType

AceType is a special AceText feature that enables you to quickly paste a clip into an application, and continue working with that application after pasting. The default hotkey is Win+Alt+A. Press this hotkey while working with an application other than AceText. AceText will pop up. If you like to use the mouse, right-click on the AceText tray icon and select AceType from the menu.

You will notice that after activating AceType, the AceText window looks a bit different. Above the list with clips in the collection, a highlighted AceType field appears. Enter the AceType abbreviation of the clip you want. As you enter the abbreviation, the list of clips in the collection is filtered. It will only show clips that have an abbreviation starting with the same characters as the ones you typed. As soon as you activate AceType, the labels of the clips in the list are prefixed with their abbreviations. Clips that do not have an abbreviation are hidden. If you forgot the exact abbreviation of a clip, you can easily look it up. While typing the abbreviation, you can press Ctrl+Arrow Up or Ctrl+Arrow Down on the keyboard to select one of the clips in the list, without typing in (the remainder of) its abbreviation. If none of the clips have an abbreviation starting with the characters you entered then AceText ignores as many characters at the end of the abbreviation you're looking up as needed to be able to show at least one clip that starts with the remaining characters.

Press the Enter key on the keyboard to confirm the selection. If the clip contains parameters, AceText substitutes them and prompts if needed. The clip is then pasted directly into the application that was active when you initiated AceType. Press the Escape key to cancel AceType. Either way, AceText resurfaces the application that was active when you initiated AcePaste, so you can continue working straight away.

If you press the AceType hotkey while AceText has keyboard focus, you can look up a clip in the same way. But when you confirm the selection with the Enter key, the clip is not sent to another application.

Obviously, AceType is only useful if you have previously assigned memorable abbreviations to your clips. This way, you can easily and even blindly insert standard chunks of texts while typing. When replying to email, for example, instead of typing the same “thank you for your purchase” phrase over and over again, store the phrase into a clip with an abbreviation of “ty”. Then you can instantly add the phrase while typing. Just press Win+Alt+A, T, Y, Enter and continue typing the rest of the email. Besides saving you the time to type in the same phrase, AceType enables you to make the phrase much lengthier or polished. Since you need to type it only once into AceText, you can spend 5 minutes writing a very nice thank you message, and then instantly replicate it whenever you answer an email message. Your correspondents will be impressed.

By default, AceText tries to paste into the target application by placing the clip on the clipboard and simulating the Ctrl+V key combination. If you want to use AceType with an application that pastes differently, or can only accept keyboard input, you can configure that application in the Applications Preferences.

See Look up and Reuse Text to learn how to use AceText.

AceEdit

AceEdit is a special AceText feature that enables you to quickly edit some text in another application with AceText, and continue working with that application when you're done editing. The default hotkey is Win+Alt+X. When you press this hotkey when working with an application other than AceText, AceText

will pop up. If you like to use the mouse, right-click on the AceText tray icon and select AceEdit from the menu.

First, select the text you want to edit in the application you're working with. Then press the AceText hotkey. AceText will pop up with the Clipboard tab active, ready to edit the text you selected.

When you're done editing, press the AceEdit hotkey again, or press Ctrl+Enter, or click with the mouse to activate the target application. AceText then sends the modified text back to the application you were working with. If you change your mind, press the Escape key to cancel AceEdit. Either way, AceText resurfaces the application that was active when you initiated AceEdit, so you can continue working straight away.

AceEdit is particularly useful when working with applications that provide very small boxes to enter lots of text, or that lack a spell checker or handy text formatting features.

By default, AceText tries to extract the text from the application by simulating a Ctrl+C keystroke, and paste it back by simulating the Ctrl+V key combination. If you want to use AceEdit with an application that copies or pastes differently, you can configure that application in the Applications Preferences.

AcePure

AcePure is a special AceText feature that enables you to paste plain text into an application after another application has placed both plain text and other formats onto the clipboard. The default hotkey is Win+Alt+P. When you press this hotkey when working with an application other than AceText, AceText removes all clipboard formats except plain text from the clipboard. If you like to use the mouse, right-click on the AceText tray icon and select AcePure from the menu.

If the option “paste the text on the clipboard directly into the active application” is selected in the Hotkeys Preferences then AceText sends a Ctrl+V key combination after it has reduced the clipboard to plain text. This way, you can use the AcePure hotkey to directly paste plain text into the application you're working with. If your application uses a key combination other than Ctrl+V for pasting then you can configure that application in the Applications Preferences.

A typical scenario where AcePure is useful is when copying text from a web page into a document. Your web browser copies the text you selected in both plain text and HTML formats to the clipboard. When you paste into your word processor, it uses the HTML format to paste so that it can preserve all the formatting of the web page. But this does mean that the pasted text will use the same font as the web page, which may be different than the font of your document. When pasting just a few words or sentences from a web page, you likely want to retain the formatting of your document, not that of the web page where you found the text. By pressing the AcePure hotkey, you tell AceText to remove the HTML from the clipboard. This leaves only the plain text to be pasted by the word processor. The word processor will apply your document's formatting to the pasted text.

AcePure does not convert formatted text like HTML into plain text. AcePure relies on the fact that most applications that are capable of copying formatted text also copy plain text as an alternative. If an application does not put plain text on the keyboard, you would not be able to copy and paste from that application into applications like Microsoft Notepad that only support plain text. So when you use AcePure, you are pasting the plain text conversion that was provided by the application that copied the text.

Shut Down AceText

When you close AceText by clicking the X button in the caption bar of AceText's window, or by pressing Alt+F4, AceText's window disappears. But AceText continues to run in the background. The icon in the system tray remains visible, hotkeys continue to work, and the ClipHistory continues to capture text.

To shut down AceText completely, right-click on AceText's icon next to the system clock, and select Shut Down AceText from the icon's menu. You can also select Shut Down AceText in the Collection menu of the main menu. AceText then stops running. Its window and tray icon disappear. All its functionality stops until you start it again.

14. Searching and Replacing

At the bottom of the AceText Editor and AceText Tower, you can find a toolbar with buttons for searching and replacing. The one on the AceText Editor has two edit boxes. In the one at the left you can enter a search term to be found. In the one at the right you can enter the replacement for a search-and-replace. The AceText Editor provides the following commands:

-  Find First
-  Find Next
-  Find Previous
-  Filter Clips

-  Replace Current Match
-  Replace Current and Find Next
-  Replace Current and Find Previous
-  Replace All

-  Search Options
-  Find Duplicate
-  Highlight All
-  Incremental Search

-  RegexBuddy
-  RegexMagic

-  Calendar

The AceText Tower only has one edit box for the search term. It offers the following commands:

-  Find First Clip
-  Find Next Clip
-  Find Previous Clip
-  Filter Clips
-  Search Options

Find First

Click the Find First button on the Search toolbar of the AceText Editor to find the first occurrence of the search term in the active clip. If you turned on the search option “all clips in this collection” then all the clips in the current collection will be searched, just like the Find First Clip button on the AceText Tower does.

If the “all clips in all collections” search option is on, then AceText starts searching in the first open collection (leftmost tab) to find the first matching clip.

Find First Clip

Click the Find First Clip button on the Search toolbar of the AceText Tower to find the first clip in the active collection containing the search term. All clips in the collection will be searched, regardless of the state of the “all clips in this collection” search option.

If the “all clips in all collections” search option is on, then AceText starts searching in the first open collection (leftmost tab) to find the first matching clip.

If you want to find the first search match in the text of the current clip, use the Find First button on the AceText Editor instead.

Find Next

Click the Find Next button on the Search toolbar of the AceText Editor to find the next occurrence of the search term in the active clip. The next occurrence is the first one to the right or below the text cursor, searching from left to right. If you turned on the “loop automatically” search option, and the search term cannot be found after the text cursor position, AceText will restart the search from the start of the clip.

If you turned on the “all clips in this collection” or “all clips in all collections” search option, then the Find Next button works the same as the Find Next Clip button on the AceText Tower.

Find Next Clip

Click the Find Next Clip button on the Search toolbar of the AceText Tower to find the first clip after the current one in the active collection containing the search term. AceText starts with the clip below the active one in the clip tree, going down until a match is found. All clips after the current one in the collection will be searched, regardless of the state of the “all clips in this collection” search option.

If the “loop automatically” option is on, and there’s no matching clip after the current one, AceText continues searching the first clip in the collection, all the way back down to the current one.

If the “all clips in all collections” search option is on, then AceText searches the next collection if there’s no matching clip after the current one in the current collection. The next collection is the one under the tab immediately to the right of the current collection’s tab. AceText starts at the topmost clip in the next collection, proceeding downwards until a matching clip is found. If not, AceText will continue searching through the collections from left to right until a match is found. If none is found, and “loop automatically is on”, AceText starts searching the first collection (leftmost tab), again proceeding from left to right until a match is found or all collections have been searched.

Find Previous

Click the Find Previous button on the Search toolbar of the AceText Editor to find the previous occurrence of the search term in the active clip. The previous occurrence is the first one to the left or above the text

cursor, searching from right to left. If you turned on the “loop automatically” search option, and the search term cannot be found before the text cursor position, AceText will restart the search from the end of the clip.

If you turned on the “all clips in this collection” or “all clips in all collections” search option, then the Find Previous button works the same as the Find Previous Clip button on the AceText Tower.

Find Previous Clip

Click the Find Previous Clip button on the Search toolbar of the AceText Tower to find the first clip before the current one in the active collection containing the search term. AceText starts with the clip above the active one in the clip tree, going up until a match is found. All clips before the current one in the collection will be searched, regardless of the state of the “all clips in this collection” search option.

If the “loop automatically” option is on, and there’s no matching clip before the current one, AceText continues searching the last clip in the collection, all the way back up to the current one.

If the “all clips in all collections” search option is on, then AceText searches the previous collection if there’s no matching clip before the current one in the current collection. The previous collection is the one under the tab immediately to the left of the current collection’s tab. AceText starts at the bottommost clip in the previous collection, proceeding upwards until a matching clip is found. If not, AceText will continue searching through the collections from right to left until a match is found. If none is found, and “loop automatically is on”, AceText starts searching the last collection (rightmost tab), again proceeding from right to left until a match is found or all collections have been searched.

Filter Clips

Click the Filter Clips button on the Search toolbar of the AceText Editor or the AceText Tower, or press Ctrl+F3 on the keyboard, to display only clips containing the search term you’ve entered. The clips that don’t contain the search term are not deleted. They’re simply made invisible. Click the Filter Clips button again to make all clips visible again.

If you edit the search term while filtering clips then AceText automatically re-applies the filter using the new search term.

Replace Current Match

Click the Replace Current Match button on the Search toolbar of the AceText Editor to replace the currently highlighted search match with the replacement text.

Replace Current and Find Next

The Replace Current and Find Next button on the Search toolbar of the AceText Editor is a shortcut to clicking Replace Current Match and then Find Next.

Replace Current and Find Previous

The Replace Current and Find Previous button on the Search toolbar of the AceText Editor is a shortcut to clicking Replace Current Match and then Find Previous.

Replace All

Click the Replace All button on the Search toolbar of the AceText Editor to search for and replace all search matches in the scope set in the search options. If you turned on the “all clips in all collections” option then the scope is all clips in all collections. If you turned on “all clips in this collection” then the scope is all clips in the active collection. If you didn’t turn on either option, the scope is just the current clip.

If you turned on the “prompt before replace all” search option then the Replace All command shows an “Are you sure?” dialog box before it does anything. This dialog box lets you see and toggle the “All Clips in This Collection” and “All Clips in All Collections” options. It also lets you turn off the prompt and choose which clip parts to make replacements in. If you toggle any options on the “Are you sure?” prompt then your new options will be used by all search commands until you change them again.

AceText does not have an option to prompt for each replacement to be made. If you would like to make replacements selectively, begin with the  Find First command to select the first search match. If you want to replace the selected search match then click  Replace Current and Find Next. If you do not want to replace the selected search match then click  Find Next. Either way the next search match will be selected. Continue clicking  Replace Current and Find Next and  Find Next to replace or not replace each search match until no further matches are found.

The benefit of replacing matches this way is that you're not stuck in the loop of a Replace All command. You can make other edits between making replacements. To continue replacing matches simply place the cursor near the position where you left off and use  Find Next or  Find Previous to go back to the match that you skipped.

Search Options

Click the Search Options button on the Search toolbar of the AceText Editor or the AceText Tower, or press Ctrl+F3 on the keyboard, to toggle various options affecting the search and replace commands in AceText.

Regular Expression

Turn on this option if you want to search for a regular expression rather than for a simple word or phrase.

Free-Spacing

When searching for a regular expression, you can turn on this option to tell AceText to ignore whitespace in your regular expression.

Dot Matches Line Breaks

When searching for a regular expression, you can turn on this option to make the dot match all characters, including line breaks. By default, the dot matches all characters except line break characters.

Case Sensitive

Turn on this option to make AceText treat the uppercase and lowercase variant of the same letter as different characters. When the “case sensitive” option is on, the search term “dog” will only match “dog”. It won’t match “DOG” or “Dog” or “DoG”. When the option is off, searching for any of these 4 variants will find all 4.

Adapt Case

When the “case sensitive” option is off, you can turn on the “adapt case” option to make AceText adapt the case of the replacement text to that of the search term. E.g. when searching for “dog” and replacing with “cat”, AceText will replace “Dog” with “Cat” and “DOG” with “CAT” when the “adapt case” option is on. If the option is off, all matches of “dog” regardless of their capitalization will be replaced with “cat”, as you entered it. AceText recognizes and adapts to all uppercase (SEARCH TERM), all lowercase (search term), first capital (Search term) and first capital for each word (Search Term). If the search match uses another kind of capitalization, the replacement text is not adapted, and used as you entered it.

Whole Words Only

By default, AceText will find all occurrences of the search term, even if as a part of another word. E.g. searching for “cat” will match the first three letters in “category”. If you turn on the “whole words only” option, “cat” will only match the word “cat”. “Category” won’t be matched.

All Clips in This Collection

By default, the Find First, Find Previous, Find Next, Replace Current and Find Previous, Replace Current and Find Next and Replace All commands on the AceText Editor only work on the active clip. If you turn on the “all clips in this collection” option, they’ll search through all clips in the collection.

The Find First Clip, Find Previous Clip, Find Next Clip and Filter Clips commands that you find on the AceText Tower always search through all clips in the collection, regardless of this option.

All Clips in All Collections

By default, all search commands only work on the clips in the active collection. Turn on “all clips in all collections” to search through all the clips in all the collections that you have opened in AceText.

Prompt Before Replace All

If you turn this on then the  Replace All command shows an “Are you sure?” dialog box before it does anything. If you turn on the options to replace all clips in all collections then you may also want to turn on the option to prompt before replacing to make sure you don’t accidentally replace across all open collections if you forget to turn off that option when you no longer need it.

The dialog box that asks you to confirm the Replace All command also lets you toggle the “All Clips in This Collection” and “All Clips in All Collections” options so you can disable them just-in-time. It also lets you turn off the prompt and choose which clip parts to make replacements in. If you toggle any options on the “Are you sure?” prompt then your new options will be used by all search commands until you change them again.

Loop Automatically

Turn on this option to make the search restart from the beginning (or from the end when searching backwards) when the search term cannot be found.

Folder and Clip Parts

You can select which parts of the clips and the folders in your collections should be affected by the search commands. If you untick a part then AceText won’t find any matches in that part. The part won’t be affected by a search-and-replace.

AceText can search through folder labels, clip labels, AceType abbreviations, the contents of text clips, the contents of binary clips, and the URLs of clips and folders. These are basically all the parts of clips and folders into which you can freely enter text or binary data.

Persist Search Options

If you turn on “persist search options” then AceText remembers your search options when you shut down and restart AceText. Be careful with this. You could be in for an unpleasant surprise when clicking the Replace All button when AceText remembers but you forgot that “all clips in all collections” was turned on last time you used AceText. Undoing that would require you to click the  Undo button in each open collection.

If you turn off “persist search options” then AceText always starts with the default search options. Those have all the folder and clip parts turned on and all other options turned off.

Persist Search History

When you right-click the Search or Replace box you can select a recently used search term or replacement string. If you turn on “persist search history” then AceText preserves those two history lists when you shut down and restart AceText. If you turn this off then the history lists are blank when you restart AceText. You should turn it off if your search terms may contain sensitive information.

AceText always preserves the “persist search history” option itself when you restart AceText even if you turn off “persist search option”. So you can independently choose whether the search options and the search history should be preserved.

Find Duplicate

Click the Find Duplicate button on the Search toolbar of the AceText Editor to find a clip that is a duplicate of the active clip. A clip is considered to be a duplicate of the active clip if all the parts that you selected in the search options are the same between the active clip and the other clip. If a part differ only in case between the two clips, then turning on the “case sensitive” search option treats the clips as different, while turning it off treats them as duplicates. The “all clips in all collections” search option determines whether Find Duplicate looks for duplicates in other collections too.

The other search options are not used by the Find Duplicate command. It always searches through all clips in the active collection. It always loops automatically if there is no duplicate of the active clip below the active clip.

Highlight All

Click the Highlight All button on the Search toolbar of the AceText Editor to highlight all search matches in the active clip.

To change the color or underlining style used for highlighting search matches, customize the color palette and change the “Editor: Highlighted search match” color.

Incremental Search

Click the Incremental button on the Search toolbar of the AceText Editor to toggle incremental search mode on or off. When off, entering a search term has no immediate effect. When on, editing the search term makes AceText look for the next search match immediately, starting from the current position of the text cursor.

While you enter the search term in instant incremental search mode, AceText remembers previous search matches as you type. If you type “Test”, AceText searches for “T” starting from the position of the text cursor. As you type, it searches for “Te” beginning at the start of the match for “T”, then for “Tes” at the start of the match for “Te”, and finally for “Test”. If there is no search match, nothing happens.

When you press Backspace, AceText goes back to the previously highlighted match for “Tes”. If you continue pressing Backspace, it goes back to the match for “Te”, “T”, and finally goes back to where the search started. So while entering the search term, you can easily change your mind about the search term you’re looking for by backing up with the Backspace key, and typing in the new search term. Backing up does not cause AceText to forget match positions. If you type in “Joe” after erasing “Test”, AceText searches for “J” until “Joe” beginning from the original starting position. If you then back up again and retype “Test”, the match previously found for “Test” is highlighted.

What does cause AceText to forget previous incremental search matches is taking any action in the main editor, including something as trivial as moving the text cursor or switching to another clip. AceText then moves the starting position for the next incremental search to the new position of the text cursor.

Regular Expressions

A regular expression (regex or regexp for short) is a special text string for describing a search pattern. You can think of regular expressions as wildcards on steroids. You are probably familiar with wildcard notations such as *.txt to find all text files in a file manager. The regex equivalent is `.*\.`. Regular expressions were not invented by Just Great Software. They're a common technology useful in a variety of situations.

AceText search-and-replace functions provide full support for regular expressions. Instead of searching for fixed search terms, you can search for text fitting a certain pattern, without specifying an exhaustive list or even knowing in advance exactly what you want to find. E.g. you could use the regular expression `\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\b` to search for an email address. Any email address, to be exact.

Regular expressions also allow you to automate tedious editing tasks with a quick search-and-replace. To turn all URLs into HTML anchor tags, for example, search for `http://\S+` and replace with `\0`.

AceText uses the same regular expression engine as PowerGREP and EditPad. It's a custom engine that blends the best features of the Perl, Java, and .NET regular expression flavors. AceText's regex flavor is fully supported by RegxBuddy and RegxMagic. These are two tools that make it easy to create regular expressions. If you have them installed you can launch them using their buttons on the Search toolbar of the AceText Editor.

RegxBuddy

Click the RegxBuddy button on the Search toolbar of the AceText Editor or press Ctrl+Shift+R in AceText to launch RegxBuddy. The regular expression you were using in AceText is automatically transferred to RegxBuddy. Click the Send button on the main toolbar in RegxBuddy to send the modified regex back to AceText.

RegxBuddy makes it easy to prepare regular expressions for use with AceText and a variety of other applications and programming languages. RegxBuddy can also analyze existing regular expressions and convert regular expressions from the hundreds of regex flavors that it supports to AceText's flavor, and vice versa.

RegxBuddy is a separate product available from Just Great Software, the developer of AceText. If you don't have it installed, then the RegxBuddy button in AceText shows a message introducing RegxBuddy.

RegxMagic

Click the RegxMagic button on the Search toolbar of the AceText Editor or press Ctrl+Shift+R in AceText to launch RegxMagic. Click the Send button on the Regex panel in RegxMagic to send the generated regular expression to AceText.

RegexMagic allows you to create regular expressions by using high-level RegexMagic patterns to describe the text that you want to match. RegexMagic then generates the regular expression for you. With RegexMagic, you don't need to deal with the terse and cryptic regex syntax at all.

RegexMagic is a separate product available from Just Great Software, the developer of AceText. If you don't have it installed, then the RegexMagic button in AceText shows a message introducing RegexMagic.

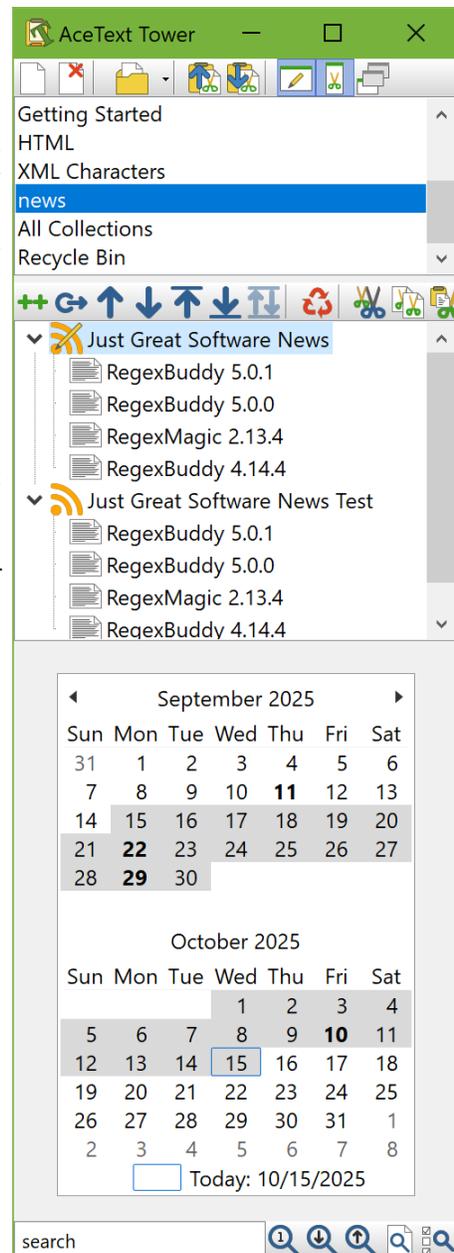
Calendar

Click the Calendar button on the Search toolbar in the lower right corner of the AceText Editor to show a calendar to filter clips by date. The calendar appears below the clip tree. By default, only one month is visible on the calendar. You can make it show multiple months by making the clip tree wider or by making the calendar taller by dragging the splitter between the calendar and the clip tree.

If one or more clips in the current collection bear a certain date, that date is indicated in bold on the calendar. Click on a date to show the clips having that date. All other clips are hidden. To show clips for a range of dates, click on the first date, and then hold down the Shift key while you click on the last date in the range.

Click on the small left and right arrow icons around the header of the first month in the calendar to make the calendar go one month back or forward in time.

Click the calendar button again to hide the calendar. All clips immediately become visible again.



15. Parameters

When you use AcePaste, AceType or the double-click method to paste a clip directly into an application, AceText scans the clip for parameters if you have ticked the Parameters checkbox for that clip on the AceText Editor. A parameter is a word or alphanumeric sequence between two percentage signs such as %PARAMETER123%. These parameters are like placeholders or variables to be substituted.

When looking for the proper text to substitute the parameter with, AceText first checks whether the parameter is a predefined parameter such as %CLIPBOARD%. These parameters are substituted with their predefined value. If the parameter isn't predefined, AceText scans the active collection for a clip that has its label set to the name of the parameter (including the two % signs). If such a clip is found, the parameter is substituted with the contents of that clip. If such a clip does not exist in the active collection, then you are prompted to enter a value for the parameter.

Predefined Parameters	Substituted Text
%%	A single literal % sign
%CURDATE%	The current date in the short date format configured in Windows
%CURTIME%	The current time in the short time format configured in Windows
%CURDATETIME%	The current date and time in the short date and time formats configured in Windows
%CURYMD%	The current date and time in YYYYMMDD format
%CURYEAR%	The current year (1601 to 9999)
%CURMONTH%	The current month (1 to 12)
%CURMONTHZ%	The current month (01 to 12)
%CURWEEK%	The current week of the year (1 to 53)
%CURWEEKZ%	The current week of the year (01 to 53)
%CURDAY%	The current day of the month (1 to 31)
%CURDAYZ%	The current day of the month (01 to 31)
%CURHOUR12%	The current hour (1 to 12)
%CURHOUR12Z%	The current hour (01 to 12)
%CURHOUR24%	The current hour (0 to 23)
%CURHOUR24Z%	The current hour (00 to 23)
%CURMIN%	The current minute (0 to 59)
%CURMINZ%	The current minute (00 to 59)
%CURSEC%	The current second (0 to 59)
%CURSECZ%	The current second (00 to 59)
%CURAMP%	AM or PM indicator for the current time
%CLIPDATE%	The date on the clip in the short date format configured in Windows
%CLIPTIME%	The time on the clip in the short time format configured in Windows
%CLIPDATETIME%	The date and time on the clip in the short date and time formats configured in

	Windows
%CLIPYMD%	The date on the clip in YYYYMMDD format
%CLIPYEAR%	The year of the date on the clip (1601 to 9999)
%CLIPMONTH%	The month of the date on the clip (1 to 12)
%CLIPMONTH%Z	The month of the date on the clip (01 to 12)
%CLIPWEEK%	The week of the year of the date on the clip (1 to 53)
%CLIPWEEKZ%	The week of the year of the date on the clip (01 to 53)
%CLIPDAY%	The day of the month of the date on the clip (1 to 31)
%CLIPDAYZ%	The day of the month of the date on the clip (01 to 31)
%CLIPHOUR12%	The hour of the time on the clip (1 to 12)
%CLIPHOUR12Z%	The hour of the time on the clip (01 to 12)
%CLIPHOUR24%	The hour of the time on the clip (0 to 23)
%CLIPHOUR24Z%	The hour of the time on the clip (00 to 23)
%CLIPMIN%	The minute of the time on the clip (0 to 59)
%CLIPMINZ%	The minute of the time on the clip (00 to 59)
%CLIPSEC%	The second of the time on the clip (0 to 59)
%CLIPSECZ%	The second of the time on the clip (00 to 59)
%CLIPAMP%	AM or PM indicator for the time on the clip
%CLIPURL%	The URL of the clip
%CLIPBOARD%	The text on the Windows clipboard
%CLIPHISTORY1%	The text of the topmost clip in the ClipHistory
%CLIPHISTORY2%	The text of the 2nd clip in the ClipHistory
%CLIPHISTORY999%	(etc.)

When the Parameters checkbox is ticked, AceText highlights all predefined parameters using the “Editor: Built-in parameter” color. All other parameters are highlighted with the “Editor: Custom parameter” color. You can find these colors at the bottom of the list of individual colors when you customize the color palette.

The %CURWEEK% and %CLIPWEEK% parameters follow the ISO 8601 standard. Monday is the first day of the week. If January 1st falls on a Friday, Saturday, or Sunday, then it falls in week 52 or 53 of the previous year. In this case, if you use %CURYEAR% or %CLIPYEAR% after their respective %CURWEEK% or %CLIPWEEK% parameter then %CURYEAR% or %CLIPYEAR% will be substituted with the number of the previous year, which is the year that the week started in. So if the clip’s date is January 1st, 2022, which was on a Saturday, then “%CLIPWEEK%/ %CLIPYEAR%” is substituted with “52/2021”. But “%CLIPMONTH%/ %CLIPYEAR%” is substituted with “1/2022”. “%CLIPYEAR%/ %CLIPWEEK%” would be substituted with “2022/52” because the parameter for the year has to come before the parameter for the week for it to be able to override the year number.

How Clips Are Found and Substituted

If the parameter isn't predefined then AceText scans the active collection for a clip that has its label set to the name of the parameter. Only the collection containing the clip you're pasting is searched. Clips in other collections that you may have open are ignored. There are a number of rules involved with this substitution.

First, the label of the clip must consist solely of the parameter, including its two percentage signs. Parameters are case insensitive, so %PARAMETER% can be substituted with a clip that has %parameter% as its label. If the collection has multiple clips with the parameter as their label, AceText substitutes only the first one. AceText doesn't even look to see if there is another.

If a clip with label %PARAMETER% has a before and after part then another clip can use the parameter %PARAMETERBEFORE% to reference the "before" part of the clip labeled %PARAMETER% and %PARAMETERAFTER% to reference the "after" part.

If you also ticked the Parameters checkbox for the clip with the label matching the parameter then AceText substitutes all parameters in that clip as well. It goes through all three steps (predefined, clip, prompt). If AceText needs to prompt, it simply adds the substitution clip's parameters to the list of parameters to be substituted for the clip you're trying to paste.

AceText does not substitute a parameter within a clip if the same clip has that parameter as its label. This prevents AceText getting stuck in an infinite loop substituting a parameter with the same clip containing the same parameter. You can take advantage of this to disable a specific parameter. If a clip has %DONTREPLACE% as its label and also has %DONTREPLACE% as its body text then AceText finds this clip whenever another clip containing %DONTREPLACE% needs to be pasted. It will substitute %DONTREPLACE% with %DONTREPLACE%, effectively doing nothing. %DONTREPLACE% will still be highlighted as a parameter in clips with the Parameters checkbox ticked. But it will be pasted unchanged. You won't be prompted for it.

How AceText Prompts for Parameters

When one or more parameters aren't predefined and aren't used as the label of a clip then AceText pops up a form listing all the parameters. All of them will default to the value you entered when you last pasted a clip from the current collection using these parameters. You can also access a history list for previously used values. The history lists are maintained separately for each collection, but jointly for all the clips in the collection.

Depending on the parameter's name, AceText expects a different kind of value. For parameters ending in %DATETIME%, it expects a date and time. For %DATE% and %TIME% it expects a date or time only. For %FILENAME% and %FILECONTENT% you'll need to type in a path to a file, or select a file by browsing your computer's drives. AceText then substitutes the parameter either with the full path to the file, or the contents of the file. For all other parameters, AceText allows you to type in any text.

16. Keyboard Shortcuts

Most of the commands available on the main toolbar and the collection toolbar have keyboard shortcuts associated with them. The editor box where you can edit the text held by a clip recognizes many additional text editing keyboard shortcuts not listed here.

The system-wide hotkeys that you can configure in the Hotkeys Preferences are also not listed here.

Main toolbar

Ctrl+N	 New AceText Collection
Ctrl+W	 Close AceText Collection
Ctrl+O	 Open AceText Collection
Ctrl+S	 Save AceText Collection
Ctrl+P	 Print AceText Collection
Shift+F5	 SequencePaste Up
Ctrl+F5	 SequencePaste Down
F8	 AceText Editor
Ctrl+F8	 AceText Tower
F12	 Always Visible
F11	 Word Wrap
Shift+F11	 Line Numbers

Collection toolbar

F7	 Go To Collection
Ctrl+F7	 Automatic Capture
Alt+F7	 Auto Append
Shift+F7	 Auto Split
Shift+F4	 New Folder
Ctrl+F4	 New Subfolder
F4	 New Clip
Shift+Ctrl+O	 Create Clip from Text File
Ctrl+D	 Duplicate Clips
Ctrl+M	 Move Clips
Shift+Ctrl+Up	 Move Up
Shift+Ctrl+Down	 Move Down
Shift+Ctrl+PgUp	 Move to Top
Shift+Ctrl+PgDown	 Move to Bottom
Ctrl+G	 Merge Clips
Shift+Ctrl+G	 Split Clips
Del	 Delete Clips

Ctrl+Z  Undo
 Ctrl+Y  Redo

Shift+Ctrl+X  Cut Clip to Clipboard
 Shift+Ctrl+C  Copy Clip to Clipboard
 Shift+Ctrl+V  Paste as New Clip

Collection tabs

Shift+Ctrl+R Read Only
 Shift+Ctrl+L Lock Collection

Search toolbar

Ctrl+F Focus Search box
 Ctrl+F  Find Next (if Search box has keyboard focus)

Ctrl+F3  Find First
 Ctrl+F6  Find First Clip
 F3  Find Next
 F6  Find Next Clip
 Shift+F3  Find Previous
 Shift+F6  Find Previous Clip
 F2  Filter Clips

Alt+F3  Replace Current and Find Next
 Shift+Alt+F3  Replace Current and Find Previous
 Ctrl+Alt+F3  Replace All

Shift+Ctrl+F  Search Options
 Alt+F2  Find Duplicate
 Ctrl+F2  Highlight All
 Ctrl+I  Incremental Search

Shift+Ctrl+R  RegexBuddy

Text editing and conversion

Shift+Ctrl+D  Duplicate Text into New Clip
 Shift+Ctrl+M  Move Text into New Clip
 Ctrl+X  Cut Text
 Ctrl+C  Copy Text
 Ctrl+V  Paste Text into Clip
 Shift+Ctrl+S  Save Text

Shift+Ctrl+U Upper Case
 Shift+Ctrl+L Lower Case

Ctrl+Q Straight ⇒ Smart Quotes
Shift+Ctrl+Q Smart ⇒ Straight Quotes

Ctrl+/ Toggle Comment
Ctrl+T Trim Whitespace
Shift+Ctrl+T Trim Whitespace from Paragraphs

Keyboard navigation

Ctrl+Tab Activate the next tab
Shift+Ctrl+Tab Activate the previous tab

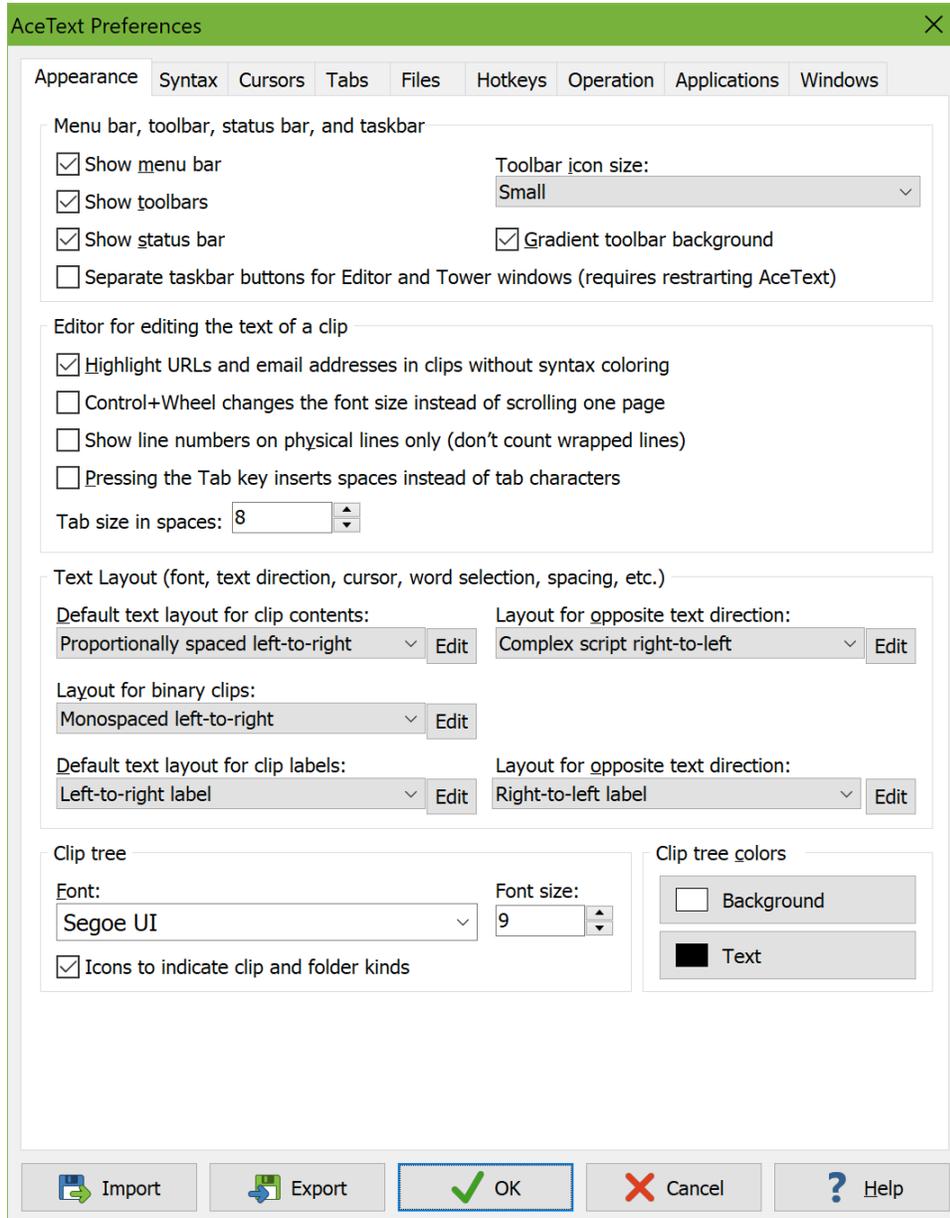
Alt+1 through Alt+0 Activate tabs 1 through 10, counting from left to right

F9 Focus Editor
Ctrl+F9 Focus Clip Tree

Ctrl+Arrow Up Select previous item
Ctrl+Arrow Down Select next item

17. Appearance Preferences

Click on the Preferences button on the main toolbar to access the preferences screen. On the Appearance page, you can change the looks and colors of AceText to better suit your taste and level of eyesight.



Menu Bar, Toolbar and Status Bar

You can choose whether you want to use the menu bar, the toolbars, or both. The menu bar can be easily accessed with Alt+Letter keyboard shortcuts. The toolbars provide direct access to the commands with a single mouse click. A few commands from the Collection menu don't have toolbar buttons. You can access those with the mouse by right-clicking on a collection's tab.

You can set the toolbar icon size to small, medium, or large. The medium icons are 25% larger than the small ones. The large icons are 50% larger than the small ones. You can also choose whether the toolbars should use a fancy gradient background, or a flat background. This is purely a matter of taste.

The status bar is the horizontal bar at the bottom of the AceText Editor. It shows helpful hints while you move the mouse pointer over different areas of AceText. It also shows the number of clips in the active collection, and the length of the active clip. You can choose whether it should be visible.

You can the  AceText Editor and  AceText Tower buttons on the main toolbar to split AceText's user interface into two windows. AceText can give each window a separate button on the Windows taskbar. Or it can use a single taskbar button for both windows. Separate taskbar buttons are useful if your PC has multiple monitors and you've configured Windows to display a separate taskbar on each monitor. If you place the AceText Tower and AceText Editor windows on separate monitors then the separate taskbar buttons can follow their own window to the taskbar on the same monitor. But if you always have the two windows on the same monitor then it may be annoying to have two taskbar buttons for one application, particularly if you are using an older version of Windows that does not automatically combine taskbar buttons that belong to the same application. Because of these considerations, the option is on by default on a PC running Windows 10 or 11 with multiple monitors. It is off by default on a PC running an older version of Windows and on a PC with only a single monitor.

You can toggle the option regardless of your version of Windows or the number of monitors. If you do you need to  Shut Down AceText and restart it for the new option to take effect. The option changes how AceText creates its windows, which it does when you start the application.

The option can also have an impact on how desktop enhancement applications that let you switch between or move around applications treat AceText. They may or may not treat the AceText Editor and AceText Tower as separate windows depending on whether they have separate taskbar buttons. If you find that AceText doesn't play well with a desktop enhancement application you're using then try toggling the option for separate taskbar buttons and restart AceText to see if that solves the issue.

Editor for Editing The Text of a Clip

Turn on the option “highlight URLs and email addresses in clips without syntax coloring” to make URLs and email addresses clickable in clips for which you did not select a syntax coloring scheme. The URLs and email addresses will be highlighted with the “Syntax: Internet link” color from the color palette that you can select on the Syntax page in the Preferences.

The checkbox labeled “Control+Wheel changes the font size instead of scrolling one page” lets you choose what happens when you rotate the mouse wheel while holding down the Control key with the mouse hovering over an edit control. If unchecked, the editor scrolls its contents up or down by as many lines as fit in the editor's height. This lets you scroll rapidly through long clips. If checked, the editor increases or decreases its font size, effectively zooming its contents.

Turn on “show line numbers on physical lines only” to only show a line number on a line if the previous line ended with a line break. This way line numbers do not change when lines are broken up by  Word Wrap. Turn it off to show line numbers for all lines. Use the  Line Numbers item in the Options menu to toggle whether line numbers are shown at all.

Turn on “pressing Tab inserts spaces instead of tab characters” if you want the Tab key on the keyboard to insert the equivalent number of spaces rather than an actual tab character. Use the “tab size in spaces” setting

to specify the width of a tab. This option affects both the display width of tab characters in your clips as well as the number of spaces inserted by the Tab key, if you enabled that option. Since the width of a tab is a multiple of the width of a space, tabbed text will only line up in AceText when using a monospaced text layout.

Text Layout

In AceText, a “text layout” is a combination of settings that control how text is displayed and how the text cursor navigates through that text. The settings include the font, text direction, text cursor behavior, which characters are word characters, and how the text should be spaced.

You can select five different text layouts. These are used in different situations:

- Default text layout for clip contents: Default layout for the body text all clips of all kinds except “binary data” clips. You can choose a left-to-right or right-to-left layout.
- Layout for opposite text direction (clip contents): Layout that all clips except “binary data” clips switch to when you use the Right-to-Left command in the Options menu. This layout must use the text direction opposite to the text direction of the default layout for clip contents.
- Layout for binary clips: Text layout used for “binary data” clips. This should be a monospaced left-to-right text layout. The Options | Right-to-Left command does not affect binary clips.
- Default text layout for clip labels: Text layout used by the single-line edit control for the clip label. You can choose a left-to-right or right-to-left layout. If you choose a left-to-right text layout then it is also used by URL field and its font will be used by every control on the AceText Editor that is not an editor control.
- Layout for opposite text direction (clip label): Text layout that the single-line edit control for the clip label switches to when you use the Right-to-Left command in the Options menu. This layout must use the text direction opposite to the text direction of the default layout for clip labels. If this layout is a left-to-right text layout then it is also used by URL field and its font will be used by every control on the AceText Editor that is not an editor control.

You can select predefined text layouts from the drop-down lists. Or, you can click the Edit button to show the Text Layout Configuration window to customize the text layouts. This is how you customize the fonts used by the AceText Editor.

Clip Tree

The clip tree font is used for the clip tree listing all the clips in the collection. Select a font face from the drop-down list and specify its size in points.

Click the Background and Text buttons to select the colors for the clip tree. Click “Default” in the color picker if you want to use the default window colors specified in the Windows Control Panel.

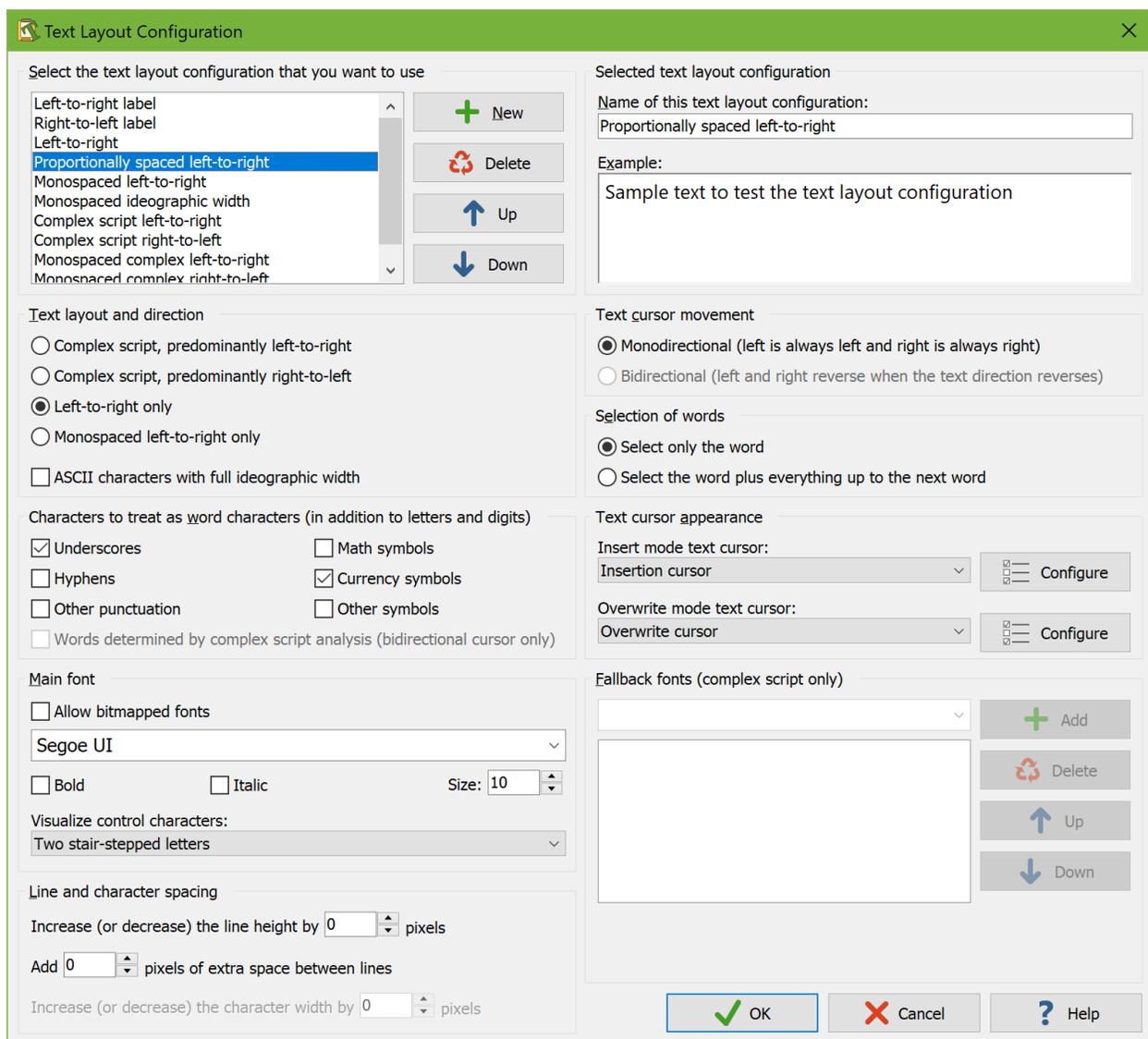
Text Layout Configuration

In AceText, a “text layout” is a combination of settings that control how text is displayed and how the text cursor navigates through that text. The settings include the font, text direction, text cursor behavior, which characters are word characters, and how the text should be spaced.

You can specify three different text layouts on the Appearance tab in the Preferences. These are used in different situations:

- Default text layout: Default layout for all clips of all kinds except “binary data” clips. You can choose a left-to-right or right-to-left layout.
- Layout for opposite text direction: Layout that all clips except “binary data” clips switch to when you use the Right-to-Left command in the Options menu. This layout must use the text direction opposite to the text direction of the default layout.
- Layout for binary clips: Text layout used for “binary data” clips. This should be a monospaced left-to-right text layout. The Options | Right-to-Left command does not affect binary clips.

You can select predefined text layouts from the drop-down lists. Or, you can click the Edit button to show the Text Layout Configuration window to customize the text layouts.



Select The Text Layout Configuration That You Want to Use

The Text Layout Configuration screen shows the details of the text layout configuration that you select in the list in the top left corner. Any changes you make on the screen are automatically applied to the selected layout and persist as you choose different layouts in the list. The changes become permanent when you click OK. The layout that is selected in the list when you click OK becomes the new default layout.

Click the New and Delete buttons to add or remove layouts. You must have at least one text layout configuration. If you have more than one, you can use the Up and Down buttons to change their order. The order does not affect anything other than the order in which the text layouts configurations appear in selection lists.

AceText comes with a number of preconfigured text layouts. If you find the options on this screen bewildering, simply choose the preconfigured layout that matches your needs, and ignore all the other settings. You can fully edit and delete all the preconfigured text layouts if you don't like them.

- Left-to-right: Normal settings with best performance for editing text in European languages and ideographic languages (Chinese, Japanese, Korean). The default font is monospaced. The layout does respect individual character width if the font is not purely monospaced or if you select another font.
- Proportionally spaced left-to-right: Like left-to-right, but the default font is proportionally spaced.
- Monospaced left-to-right: Like left-to-right, but the text is forced to be monospaced. Columns are guaranteed to line up perfectly even if the font is not purely monospaced. This is the best choice for working with source code and text files with tabular data.
- Monospaced ideographic width: Like monospaced left-to-right, but ASCII characters are given the same width as ideographs. This is the best choice if you want columns of mixed ASCII and ideographic text to line up perfectly.
- Complex script left-to-right: Supports text in any language, including complex scripts (e.g. Indic scripts) and right-to-left scripts (Hebrew, Arabic). Choose this for editing text that is written from left-to-right, perhaps mixed with an occasional word or phrase written from right-to-left.
- Complex script right-to-left: For writing text in scripts such as Hebrew or Arabic that are written from right-to-left, perhaps mixed with an occasional word or phrase written from left-to-right.
- Monospaced complex left-to-right: Like “complex script left-to-right”, but using monospaced fonts for as many scripts as possible. Text is not forced to be monospaced, so columns may not line up perfectly.
- Monospaced complex right-to-left: Like “complex script right-to-left”, but using monospaced fonts for as many scripts as possible. Text is not forced to be monospaced, so columns may not line up perfectly.

Selected Text Layout Configuration

The section in the upper right corner provides a box to type in the name of the text layout configuration. This name is only used to help you identify it in selection lists when you have prepared more than one text layout configuration.

In the Example box you can type in some text to see how the selected text layout configuration causes the editor to behave.

Text Layout and Direction

- Complex script, predominantly left-to-right: Text is written from left to right and can be mixed with text written from right to left. Choose this for complex scripts such as the Indic scripts, or for text in any language that mixes in the occasional word or phrase in a right-to-left or complex script.
- Complex script, predominantly right-to-left: Text is written from right to left and can be mixed with text written from left to right. Choose this for writing text in scripts written from right to left such as Hebrew or Arabic.
- Left-to-right only: Text is always written from left to right. Complex scripts and right-to-left scripts are not supported. Choose this for best performance for editing text in European languages and ideographic languages (Chinese, Japanese, Korean) that is written from left to right without exception.
- Monospaced left-to-right only: Text is always written from left to right and is forced to be monospaced. Complex scripts and right-to-left scripts are not supported. Each character is given the same horizontal width even if the font specifies different widths for different characters. This guarantees columns to be lined up perfectly. To keep the text readable, you should choose a monospaced font.
- ASCII characters with full ideographic width: You can choose this option in combination with any of the four preceding options. In most fonts, ASCII characters (English letters, digits, and punctuation) are about half the width of ideographs. This option substitutes full-width characters for the ASCII characters so they are the same width as ideographs. If you turn this on in combination with “monospaced left-to-right only” then columns that mix English letters and digits with ideographs will line up perfectly.

Text Cursor Movement

- Monodirectional: The left arrow key on the keyboard always moves the cursor to the left on the screen and the right arrow key always moves the cursor to the right on the screen, regardless of the predominant or actual text direction.
- Bidirectional: This option is only available if you have chosen one of the complex script options in the “text layout and direction” list. The direction that the left and right arrow keys move the cursor into depends on the predominant text direction selected in the “text layout and direction” list and on the actual text direction of the word that the cursor is pointing to when you press the left or right arrow key on the keyboard.
 - Predominantly left-to-right: The left key moves to the preceding character in logical order, and the right key moves to the following character in logical order.
 - Actual left-to-right: The left key moves left, and the right key moves right.
 - Actual right-to-left: The actual direction is reversed from the predominant direction. The left key moves right, and the right key moves left.
 - Predominantly right-to-left: The left key moves to the following character in logical order, and the right key moves to the preceding character in logical order.
 - Actual left-to-right: The actual direction is reversed from the predominant direction. The left key moves right, and the right key moves left.
 - Actual right-to-left: The left key moves left, and the right key moves right.

Selection of Words

- Select only the word: Pressing Ctrl+Shift+Right moves the cursor to the end of the word that the cursor is on. The selection stops at the end of the word. This is the default behavior for all Just Great Software applications. It makes it easy to select a specific word or string of words without any extraneous spaces or characters. To include the space after the last word, press Ctrl+Shift+Right once more, and then Ctrl+Shift+Left.
- Select the word plus everything to the next word: Pressing Ctrl+Shift+Right moves the cursor to the start of the word after the one that the cursor is on. The selection includes the word that the cursor was on and the non-word characters between that word and the next word that the cursor is moved to. This is how text editors usually behave on the Windows platform.

Character Sequences to Treat as words

Characters that are considered to be letters or digits by the Unicode standard are selected when you double-click them. Ctrl+Left and Ctrl+Right move the cursor to the start of the preceding or following sequence of letters and digits. Ideographs are considered to be letters.

You can use the six checkboxes to add underscores, hyphens, punctuation other than underscores or hyphens, math symbols, currency symbols, or symbols other than math or currency symbols to the characters that are selected when you double-click them. If you tick all of them then double-clicking selects any run of non-whitespace characters.

If you selected the “bidirectional” text cursor movement option then you can turn on “words determined by complex script analysis” to allow Ctrl+Left and Ctrl+Right to place the cursor between two letters for languages such as Thai that don’t write spaces between words.

Text Cursor Appearance

Select a predefined cursor or click the Configure button to show the text cursor configuration screen. There you can configure the looks of the blinking text cursor (and even make it stop blinking).

A text layout uses two cursors. One cursor is used for insert mode, where typing in text pushes ahead the text after the cursor. The other cursor is used for overwrite mode, where typing in text replaces the characters after the cursor. Pressing the Insert key on the keyboard toggles between insert and overwrite mode.

Main Font

Select the font that you want to use from the drop-down list. Turn on “allow bitmapped fonts” to include bitmapped fonts in the list. Otherwise, only TrueType and OpenType fonts are included. Using a TrueType or OpenType font is recommended. Bitmapped fonts may not be displayed perfectly (e.g. italics may be clipped) and only support a few specific sizes.

If you access the text layout configuration screen from the print preview, then turning on “allow bitmapped fonts” will include printer fonts rather than screen fonts in the list, in addition to the TrueType and

OpenType fonts that work everywhere. A “printer font” is a font built into your printer’s hardware. If you select a printer font, set “text layout and direction” to “left to right only” for best results.

Normally, plain text files should not contain control characters other than tabs and line breaks. AceText has separate commands to »• Visualize Spaces and ¶ Visualize Line Breaks. Most fonts are unable to display control characters. Therefore AceText provides several options for visualizing control characters. The first 3 are independent of the font. The last 3 require support from your main font.

- Two stair-stepped letters: Similar to how AceText visualizes specific line break styles, such as ^B for backspace.
- Control+letter: Indicates ASCII control character with the Ctrl+letter combination that they are traditionally entered with on character terminals, such as ^H for backspace. Indicates ISO-8859 control characters as Esc+letter.
- Stair-stepped hexadecimal numbers: Hexadecimal index of the control character in the ASCII or ISO-8859 character tables, such as ⁰⁸ for backspace.
- IBM PC glyphs: The original IBM PC displayed various graphical symbols when an application wrote an ASCII control character to the screen. If your font supports the Unicode characters that represent these symbols then AceText can display these symbols. The backspace control character then appears as an inverse bullet ■, for example. ISO-8859 control characters are visualized with two stair-stepped letters. The IBM PC predates ISO-8859.
- Unicode control pictures: The Unicode standard has code points specific to visualizing the ASCII control characters. Few fonts support these. Those that do typically render them as 2 or 3 stair-stepped letters, such as ^B for backspace. As you can see, these letters can be tiny with some fonts. Unicode does not have such code points for the ISO-8859 control characters. So AceText uses its own stair-stepped letter visualization for those.
- Actual control characters: This option tells AceText to blindly rely on the font to be able to display control characters. With most fonts this makes the control characters completely invisible. This can cause text spacing issues in AceText. So only use this option if you know your font can render all ASCII and ISO-8859 control characters.

Fallback Fonts

Not all fonts are capable of displaying text in all scripts or languages. If you have selected one of the complex script options in the “text layout and direction” list, you can specify one or more “fallback” fonts. If the main font does not support a particular script, AceText will try to use one of the fallback fonts. It starts with the topmost font at the list and continues to attempt fonts lower in the list until it finds a font that supports the script you are typing with. If none of the fonts supports the script, then the text will appear as squares.

To figure out which scripts a particular font supports, first type or paste some text using those scripts into the Example box. Make sure one of the complex script options is selected. Then remove all fallback fonts. Now you can change the main font and see which characters that font can display. When you’ve come up with a list of fonts that, if used together, can display all of your characters, select your preferred font as the main font. Then add all the others as fallback fonts.

Line and Character Spacing

By default all the spacing options are set to zero. This tells AceText to use the default spacing for the font you have selected.

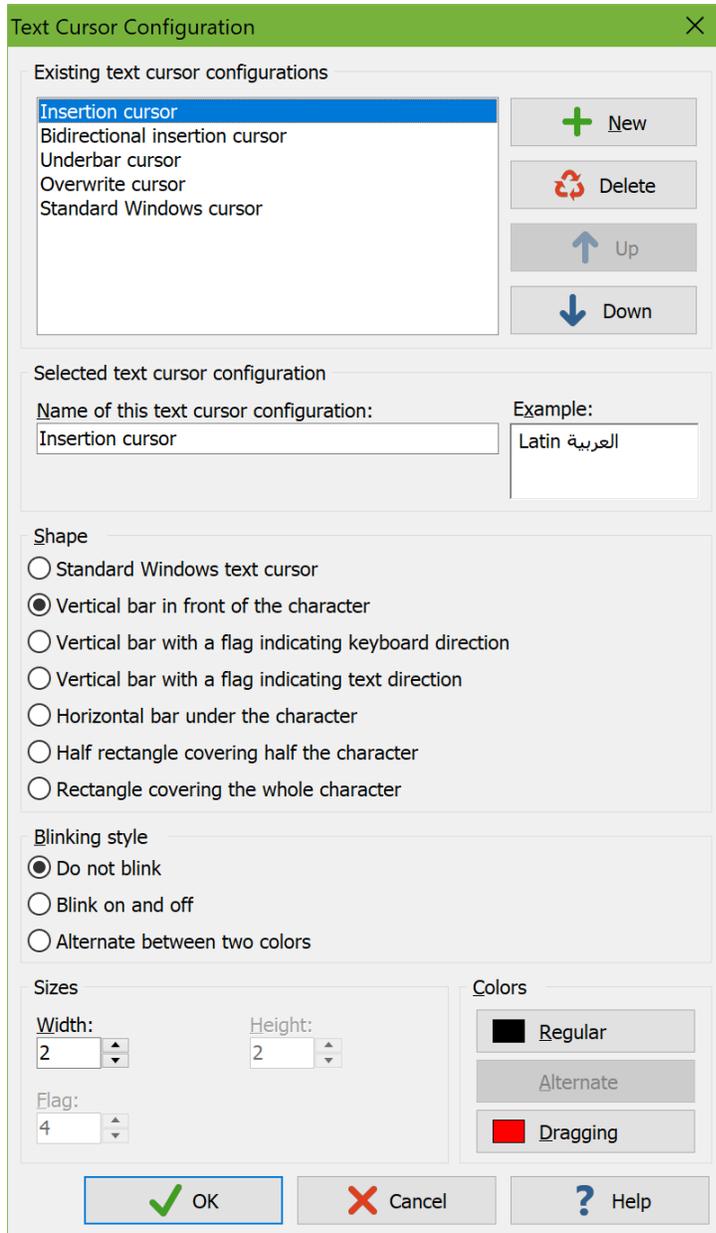
If you find that lines are spaced apart too widely, specify a negative value for “increase (or decrease) the line height”. Set to “add 0 pixels of extra space between lines”.

If you find that lines are spaced too closely together, specify a positive value for “increase (or decrease) the line height” and/or “add ... pixels of extra space between lines”. The difference between the two is that when you select a line of text, increasing the line height increases the height of the selection highlighting, while adding extra space between lines does not. If you select multiple lines of text, extra space between lines shows up as gaps between the selected lines. Adding extra space between lines may make it easier to distinguish between lines.

The “increase (or decrease) the character width by ... pixels” setting is only used when you select “monospaced left-to-right” only in the “text layout and direction” list. You can specify a positive value to increase the character or column width, or a negative value to decrease it. This can be useful if your chosen font is not perfectly monospaced and because of that characters appear spaced too widely or too closely.

Text Cursor Configuration

You can access the text cursor configuration screen from the text layout configuration screen by clicking one of the Configure buttons in the “text cursor appearance” section.



Existing Text Cursor Configurations

The Text Cursor Configuration screen shows the details of the text cursor configuration that you select in the list at the top. Any changes you make on the screen are automatically applied to the selected cursor and persist as you choose different cursors in the list. The changes become permanent when you click OK. The cursor that is selected in the list when you click OK becomes the new default cursor.

Click the New and Delete buttons to add or remove cursors. You must have at least one text cursor configuration. If you have more than one, you can use the Up and Down buttons to change their order. The order does not affect anything other than the order in which the text cursor configurations appear in selection lists.

AceText comes with a number of preconfigured text cursors. You can fully edit or delete all the preconfigured text cursors if you don't like them.

- Insertion cursor: Blinking vertical bar similar to the standard Windows cursor, except that it is thicker and fully black, even on a gray background.
- Bidirectional insertion cursor: Like the insertion cursor, but with a little flag that indicates whether the keyboard layout is left-to-right (e.g. you're typing in English) or right-to-left (e.g. you're typing in Hebrew). The flag is larger than what you get with the standard Windows cursor and is shown even if you don't have any right-to-left layouts installed.
- Underbar cursor: Blinking horizontal bar that lies under the character. This mimics the text cursor that was common in DOS applications.
- Overwrite cursor: Blinking rectangle that covers the bottom half of the character. In AceText this is the default cursor for overwrite mode. In this mode, which is toggled with the Insert key on the keyboard, typing text overwrites the following characters instead of pushing them ahead.
- Standard Windows cursor: The standard Windows cursor is a very thin blinking vertical bar that is XOR-ed on the screen, making it very hard to see on anything except a pure black or pure white background. If you have a right-to-left keyboard layout installed, the cursor gets a tiny flag indicating keyboard direction. You should only use this cursor if you rely on accessibility software such as a screen reader or magnification tool that fails to track any of AceText's other cursor shapes.

Selected Text Cursor Configuration

Type in the name of the text cursor configuration. This name is only used to help you identify it in selection lists when you have prepared more than one text cursor configuration.

In the Example box you can type in some text to see what the cursor looks like. The box has a word in Latin and Arabic so you can see the difference in cursor appearance, if any, based on the text direction of the word that the cursor is on.

Shape

- Standard Windows Text cursor: The standard Windows cursor is a very thin blinking vertical bar that is XOR-ed on the screen, making it very hard to see on anything except a pure black or pure white background. If you have a right-to-left keyboard layout installed, the cursor gets a tiny flag indicating keyboard direction. You should only use this cursor if you rely on accessibility software such as a screen reader or magnification tool that fails to track any of AceText's other cursor shapes. The standard Windows cursor provides no configuration options.
- Vertical bar in front of the character: On the Windows platform, the normal cursor shape is a vertical bar that is positioned in front of the character that it points to. That is to the left of the character for left-to-right text, and to the right of the character for right-to-left text.
- Vertical bar with a flag indicating keyboard direction: A vertical bar positioned in front of the character that it points to, with a little flag (triangle) at the top that indicates the direction of the active keyboard layout. When the cursor points to a character in left-to-right text, it is placed to the left of that character. When the cursor point to a character in right-to-left text, it is placed to the right of that character. The direction of the cursor's flag is independent of the text under the cursor. The cursor's flag points to the right when the active keyboard layout is for a left-to-right language. The cursor's flag points to the left when the active keyboard layout is for a right-to-left language.

- Vertical bar with a flag indicating text direction: A vertical bar positioned in front of the character that it points to, with a little flag (triangle) at the top that points to that character. When the cursor points to a character in left-to-right text, it is placed to the left of that character with its flag pointing to the right towards that character. When the cursor point to a character in right-to-left text, it is placed to the right of that character with its flag pointing to the left towards that character.
- Horizontal bar under the character: In DOS applications, the cursor was a horizontal line under the character that the cursor points to.
- Half rectangle covering half the character: The cursor covers the bottom half of the character that it points to. This is a traditional cursor shape to indicate typing will overwrite the character rather than push it ahead.
- Rectangle covering the whole character: The cursor makes the character invisible. This can also be used to indicate overwrite mode.

Blinking Style

- Do not blink: The cursor is permanently visible in a single color. Choose this option if the blinking distracts you or if it confuses accessibility software such as screen readers or magnification tools.
- Blink on and off: The usual blinking style for text cursors on the Windows platform. The cursor is permanently visible while you type (quickly). When you stop typing for about half a second, the cursor blinks by becoming temporarily invisible. Blinking makes it easier to locate the cursor with your eyes in a large block of text.
- Alternate between two colors: Makes the cursor blink when you stop typing like “on and off”. But instead of making the cursor invisible, it is displayed with an alternate color. This option gives the cursor maximum visibility: the blinking animation attracts the eye while keeping the cursor permanently visible.

Sizes

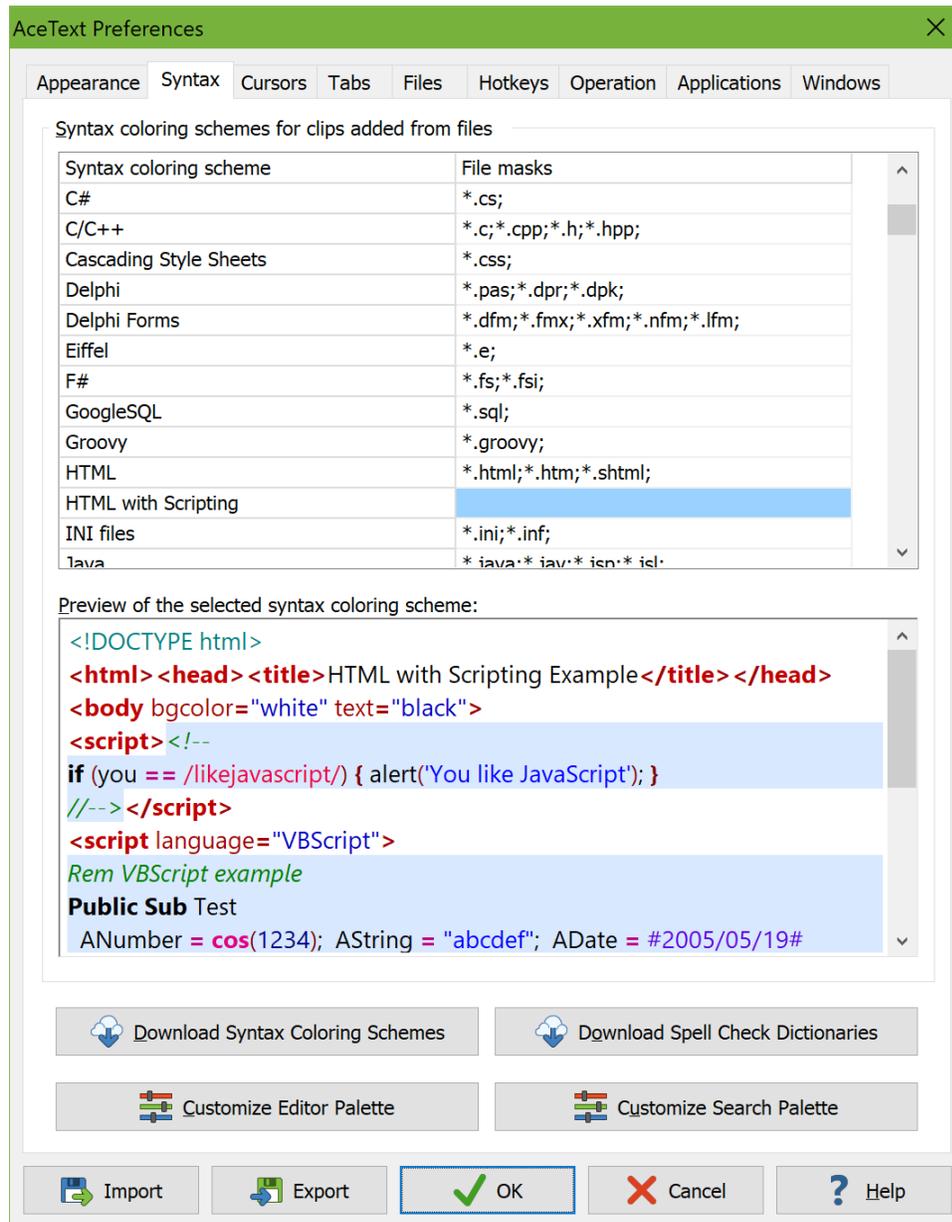
- Width: Width in pixels for the vertical bar shape.
- Height: Height in pixels for the horizontal bar shape.
- Flag: Length in pixels of the edges of the flag that indicates text direction.

Colors

- Regular: Used for all shapes and blinking styles except the standard Windows cursor.
- Alternate: Alternate color used by the “alternate between two colors” blinking style.
- Dragging: Color of a second “ghost” cursor that appears while dragging and dropping text with the mouse. It indicates the position the text is moved or copied to when you release the mouse button.

18. Syntax Preferences

Click on the Preferences button on the main toolbar to access the preferences screen. On the Syntax page, you can associate syntax coloring schemes with file masks. The  Create Clip from Text File uses this to automatically associate the clips that it adds with the appropriate syntax coloring scheme for each file.



The grid shows all syntax coloring schemes available to AceText in the first column. The second column shows the file mask associated with each scheme. By default, AceText uses the file mask stored in the scheme itself. You can directly edit all the file masks in the second column. Doing so does not modify the syntax coloring schemes. It only changes the list of file masks in AceText.

Several of the syntax coloring schemes included with AceText have the same file masks. To make AceText use a particular scheme for a particular file type, make sure to remove any duplicate file masks. For example, the schemes for the various SQL dialects all include *.sql in their file mask. You should remove *.sql from all the SQL schemes except the one that you prefer. Schemes that don't have file masks can still be assigned to clips. They just won't be assigned automatically by the Create Clip from Text File command.

While you edit the file mask of a particular syntax coloring scheme you see a preview of that syntax coloring scheme. Each syntax coloring scheme contains one example. This preview can help you differentiate between multiple syntax coloring schemes for the same file formats. You can edit the example to see how the scheme applies its colors to other text. Your changes are lost as soon as you edit the file masks of another scheme.

File Masks

In a file mask, the asterisk (*) represents any number (including none) of any character, similar to `.*` in a regular expression. The question mark (?) represents one single character, similar to `?` in a regular expression. The file mask `*.txt` tells AceText to use the scheme for any file with a .txt extension.

File masks also support a simple character class notation, which matches one character from a list or a range of characters. The file mask `*.ext[1-9]` matches any file with an extension from .ext1 to .ext9. File masks are not limited to the *.ext format. You can make them match specific file names too. `Oracle*.sql`, for example, matches any file that has a name beginning with "Oracle" and ending with a .sql extension.

You can delimit multiple file masks with any mixture of semicolons and commas. Only one file mask needs to match for the scheme to be selected for a file. The provided scheme for C and C++ has `*.c;*.cpp;*.h;*.hpp`; as its default file mask to select all C and C++ header and source files.

A file mask is only considered to match the file name if it matches the whole file name. The mask `*.txt` matches `joe.txt`, but not `joe.txt.doc` since the latter does not end in txt. You could use the mask `*.txt*` to match both.

Download Syntax Coloring Schemes

Click the Download Syntax Coloring Schemes button to download syntax coloring schemes contributed by other users of AceText and other Just Great Software products such as EditPad that use the syntax coloring schemes.

If you would like to create your own syntax coloring schemes, download and install the syntax coloring scheme editor.

Download Spell Check Dictionaries

AceText's Live Spell Check only works if you have previously downloaded and installed Just Great Software spell checker dictionaries. Click the Download Spell Checker Dictionaries button to download some or all of the spell check dictionaries free of charge.

Customize Palettes

Click the Customize Editor Palette button to customize the color palette used by the AceText Editor. This affects the editor for the body text of the clip as well as the edit boxes for the label, AceType abbreviation, and URL. This includes the actual colors used by the syntax coloring schemes and by clips with emphasis.

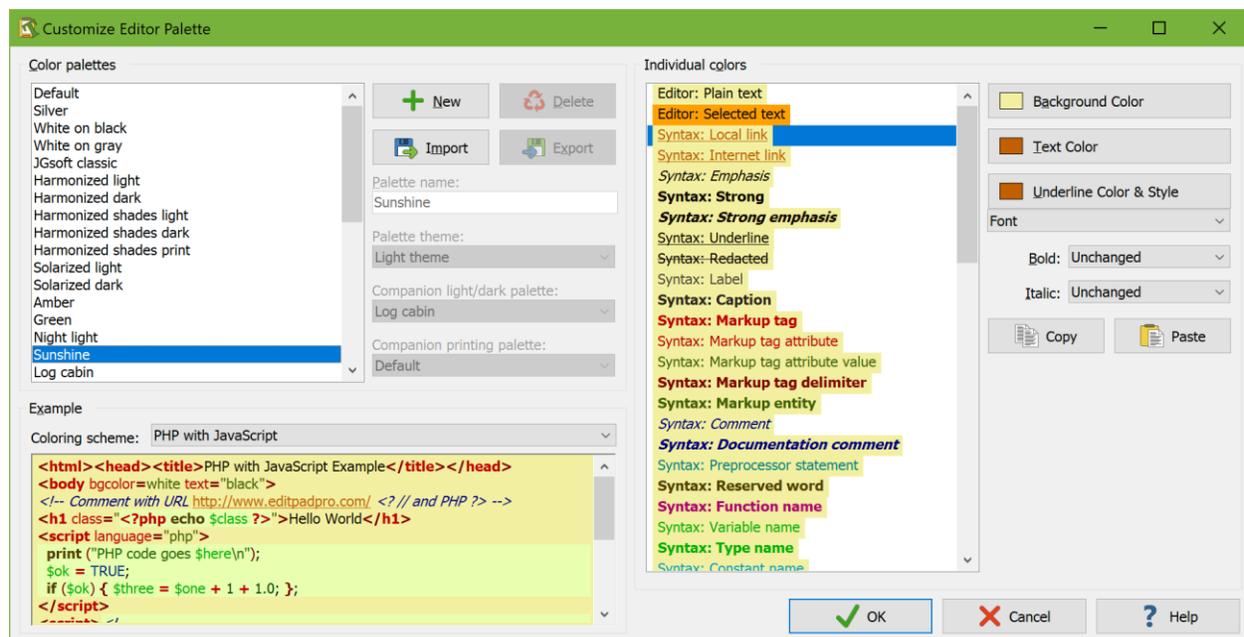
Click the Customize Search Palette button to customize the color palette used by the Search and Replace boxes. This includes the syntax colors for regular expressions and replacement strings.

Color Palettes

Click the Customize Editor Palette button in the Appearance Preferences to customize the color palette used by the AceText Editor. This affects the editor for the body text of the clip as well as the edit boxes for the label, AceType abbreviation, and URL. This includes the actual colors used by the syntax coloring schemes and by clips with emphasis.

Click the Customize Search Palette button in the Appearance Preferences to customize the color palette used by the Search and Replace boxes. This includes the syntax colors for regular expressions and replacement strings.

Both buttons show the same color customization dialog. But the list of individual colors will be different. There are also fewer predefined search palettes. Custom palettes are separate.



Color Palettes

AceText comes with a long list of predefined color palettes. All of these exist as Editor palettes with syntax colors. Most of these also exist as Search palettes with regex colors.

- Default: The default AceText color palette. Its plain text color uses the default window text and background colors in the active Windows theme. In Windows 7 and prior you could change these colors in the Appearance section in the Windows Control Panel. Usually they're black on pure white. This palette has distinct colors for almost all of the items. Choose this configuration if you like very detailed syntax coloring.
- Silver: Same colors as the default palette, but with black text on a silver background. This is easier on the eyes if your monitor has a very high brightness.
- White on black: Pure white text on a pure black background, with colorful syntax highlighting. This is a high contrast palette that works well as a dark palette on monitors with washed-out blacks (backlight bleeding on older LCD monitors).
- White on gray: White text on a dark gray background, with colorful syntax highlighting. The reduced contrast is easier on the eyes on monitors with very deep blacks.
- JGsoft classic: The color scheme used by EditPad Pro 4 and 5. It uses the default window text and background from the Windows Control Panel. Uses a limited set of different colors for syntax coloring.
- Harmonized light: Uses shades of dark gray for plain text and shades of light gray for background and highlighting. Uses 12 colors that are equidistant on the color wheel and of equal brightness for syntax coloring. This gives your files a uniform low-contrast look. Bracket and search match highlighting is done with underlining using some of those 12 colors.
- Harmonized dark: Uses the opposite shades of gray as “harmonized light” and exactly the same 12 colors for syntax coloring.
- Harmonized shades light: Same as “harmonized light”, but adds a lighter, less saturated shade of the 12 colors for things like bracket and search match highlighting.
- Harmonized shades dark: Same as “harmonized dark”, but adds a darker, less saturated shade of the 12 colors for things like bracket and search match highlighting.
- Harmonized shades print: Same as “harmonized shades light”, but using pure black on pure white for plain text.
- Solarized light: Using the 16 colors of the Solarized palette created by Ethan Schoonover.
- Solarized dark: Same as “solarized light” but using the opposite content and background tones.
- Amber: Amber text on a dark gray background like old terminal monitors. Uses a mixture of green and other colors for syntax coloring.
- Green: Green text on a dark gray background like old terminal monitors. Uses a mixture of amber and other colors for syntax coloring.
- Night light: Very dark palette with amber text. Uses some yellow and green for syntax coloring. Does not use any blue colors.
- Sunshine: Colorful palette with a pastel yellow background.
- Log cabin: Colorful palette with a dark brown background.
- Blue sky: Colorful palette with a sky blue background.
- Twilight: Colorful palette with a navy blue background.
- Monochrome light: Dark gray text on a light gray background. Uses different shades of gray for highlighting. Uses bold, italic, underlining, and a few different shades of gray for syntax coloring.
- Monochrome amber: Amber text on a dark gray background like old terminal monitors. Uses different shades of gray for highlighting. Uses bold, italic, underlining, and a few different shades of amber for syntax coloring.
- Monochrome green: Green text on a dark gray background like old terminal monitors. Uses different shades of gray for highlighting. Uses bold, italic, underlining, and a few different shades of green for syntax coloring.
- Monochrome dark: As “monochrome light” with the inverse shades of gray.
- Monochrome print: As “monochrome light” but with pure black text on a pure white background.

- Red-green color blind light: Intended for people who cannot perceive the difference between red and green. All colors have equal amounts of red and green. Dark gray text on a light gray background. Uses shades of yellow and blue for highlighting and syntax coloring.
- Red-green color blind dark: As “red-green color blind light” but with light gray text on a dark gray background.
- Red-green color blind print: As “red-green color blind light” but with pure black text on a pure white background.
- Yellow-blue color blind light: Intended for people who cannot perceive the difference between yellow and blue. All colors have equal amounts of green and blue. Dark gray text on a light gray background. Uses shades of red and cyan for highlighting and syntax coloring.
- Yellow-blue color blind dark: As “yellow-blue color blind light” but with light gray text on a dark gray background.
- Yellow-blue color blind print: As “yellow-blue color blind light” but with pure black text on a pure white background.
- Visual Studio light: Emulates the colors used by Microsoft’s Visual Studio light theme.
- Visual Studio dark: Emulates the colors used by Microsoft’s Visual Studio dark theme.
- Embarcadero: Emulates the colors used by recent versions of Delphi and C++Builder.
- Embarcadero dark: Emulates the colors used by the dark theme in recent versions of Delphi and C++Builder.
- Borland classic: Emulates the colors used by old Borland development tools for DOS, with yellow text on a blue background.
- PowerShell ISE light: Emulates the light colors used by the Windows PowerShell ISE.
- PowerShell ISE dark: Emulates the dark colors used by the Windows PowerShell ISE.

The built-in palettes cannot be customized. If you try to edit any of the individual colors then AceText automatically makes a copy of the palette that you can then edit. You can also duplicate the selected palette by clicking the New button. Click the Delete button to remove a custom palette.

You can share palettes with others by clicking the Export button to save a palette as an .ini file. The Import button allows you to import an .ini file saved by the Export button. It will not import any other .ini files.

Each palette should have a name so you can select it in the list of palettes in the file type configuration.

If you create light and dark versions of the same palette, then you should set the “palette theme” option accordingly. After you have set “palette theme” you can select a custom palette of the opposite theme in the “companion light/dark palette”. If you do this then the View|Dark Theme menu item swaps your file types between these two palettes when you toggle the theme.

If you use  Print Collection or  Print Text then AceText prints your file with the colors specified in your palette. If your palette’s background color is not white then AceText will fill the entire page with ink. In that case you may want to create an extra version of your palette that has the plain text colors set to pure black text on a pure white background. You can then select that palette as the “companion printing palette” to your other palettes. You do not need to designate a palette as a printing palette. The “companion printing palette” drop-down list automatically lists all custom palettes that have the plain text colors set to pure black text on a pure white background. If you edit a file using a color palette for which you selected a companion printing palette then the print preview defaults to that printing palette instead of the palette you are editing with.

Changing Individual Colors

If you customize one of the harmonized, solarized, or monochrome palettes then the color picker restricts you to the specific set of harmonized, solarized, or monochrome colors that are available to the palette you're customizing. To change a color, simply click the Color button and then click on the color you want.

For all other palettes, the color picker shows a grid of 5 rows of 8 common colors. Any other colors used by your palette are shown below the 40 common colors. The other colors are shown in the order of the first individual color that uses that color. The default palette, for example, uses different shades of red for markup tags and markup attributes. The red of markup tags is listed first because “markup tags” is above “markup attributes” in the list of individual colors. Each color is shown only once. All the color buttons show the same color picker. So the Background Color button also shows colors that are only used for text, for example.

You can pick a more specific color by clicking the More button to show the color hexagon. The colors on the outer edge of the hexagon are always fully saturated. The colors nearer to the center of the hexagon are progressively desaturated towards white, gray, or black. Click on the vertical grayscale slider to change the luminance of the colors in the center of the hexagon. If you want to pick a pure grayscale value, click on the horizontal line of small gray hexagons. If you want pure white or black, click on the larger black or white hexagon to the left or right of the gray hexagons. If you want to use a specific RGB value, enter three numbers between 0 and 255 in the edit boxes that are shaded red, green, and blue.

The color hexagon, along with the vertical grayscale slider, makes it easy to pick related colors. The concepts that follow may be easier to understand if you see the large color hexagon as a circle with discrete steps, and the small hexagons that comprise it as positions at a certain radius and angle. Colors with the same distance to the center (same radius) have the same amount of saturation. The colors at the edge are always fully saturated and the color at the center is always fully desaturated. You can change the hue without changing the saturation by selecting another color at the same distance (same radius, different angle). You can change the saturation without changing the hue by selecting a color closer to or further away from the center (different radius, same angle). For colors that are not fully saturated, the grayscale slider controls the luminance. To make a color lighter or darker, first click on the grayscale slider to change the luminance. Then click on the same spot in the color hexagon to select the same color with the different luminance.

Syntax elements that use colors of different hues but similar saturation and luminance allow those elements to be distinguished without drawing attention to some over the others. Syntax elements for important structural parts of the file can be given colors with more luminance while syntax elements for less important parts or more common parts (large blocks of text) can be given colors with less luminance to make it easier for the eye to gravitate towards the important elements. This can be further enhanced by using bold text for the most important syntax elements. If the base (plain text) color of the scheme is white, gray, or black, then setting the luminance slider at that level of white, gray, or black allows you to use more saturated colors for more important elements and less saturated colors for less important elements, as an alternative technique to using luminance. Color schemes that use very saturated colors appear bolder, while color schemes with (somewhat) desaturated colors (using luminance instead of saturation for highlighting) appear more relaxing.

If you prefer to use the standard Windows color selection dialog box to pick your colors, right-click the Background Color or Text Color button. This dialog allows you to specify hue and saturation using a two-dimensional grid and luminance using a vertical slider. You can also enter HSL or RGB values with the keyboard. If you do this while customizing a harmonized, solarized, or monochrome palette then you can break the restricted nature of the palette.

If you want to use exactly the same color for two items, first select the item that already has the color you want. Click the Color button that has the color you want. One of the color squares in the color picker should have a beveled edge. Note its location in the grid of squares.

After observing the color, click on the item in the list that should have the same color. The color picker automatically closes when you do this. Click the Color button that should have the same color. Click on the same square in the grid that you observed.

You can use the Copy and Paste buttons to make two individual colors identical. Clicking Paste changes all the colors and styles of the selected individual color to the ones that you copied.

Default Colors and Unchanged Styles

Colors can be set to “default” and text styles can be set to “unchanged”. For the “editor: plain text” color the default colors are the window text and background colors configured in the Windows Control Panel (as part of the Windows theme or via the advanced appearance settings). The default plain text style is the one you selected in Options | Text Layout or Options | Font.

For all other individual colors, the default color and style depends on how the colors are layered. Multiple colors can apply to the same text. These colors may be layered onto the same piece of text, in decreasing order of priority:

1. Selection
2. Matching bracket
3. Search match
4. Search range
5. Misspelled word
6. Syntax color
7. Active line highlight
8. Syntax coloring subscheme highlight

All colors (background, text, underline, strikeout) and all styles (bold, italic, underline, strikeout) are determined separately. AceText takes each color and each style from the individual color with the highest priority that applies to that text that has that color or style set to something other than “default” or “unchanged”. If all individual colors that apply have the text or background color set to “default” or a style set to “unchanged” then the color or style of the “editor: plain text” color are used for that text. If all individual colors that apply have the underline or strikeout color set to “default” then the text color is used for underline or strikeout.

A few examples make this easier to understand. The “white on black” palette has the “editor: selected text” color set to black text on a purple background. So with this palette, any text you select is black on purple because the selection highlight always has the highest priority. But the “harmonized dark” palette sets the selection background to a shade of gray but the selection foreground to “default”. With this palette, the selection highlight only appears as a different background color. Selected text keeps the colors it gets from syntax coloring. Of course, this only works well if the background color for selected text contrasts well with all the text colors that may be used by syntax coloring.

If the selection contains a highlighted search match then that search match gets a double underline with both these palettes. That underline remains when you select the search match as both palettes have the underline style for “editor: selected text” set to “unchanged”. The underlining remains even though the yellow

background that the “white on black” palette specifies for highlighted search matches disappears when the search match is selected.

List of Individual Colors

The colors prefixed with “editor” are used to draw various parts of AceText’s editor control. Both the editor and search palettes have these colors. The clip editor uses these colors even when the clip does not have syntax coloring. The search box uses them even when the search term is plain text.

- Editor, Plain text: The default text colors. The editor background is filled with this color. Text that is not syntax colored is drawn in this text color.
- Editor, Selected text: Selection highlight.
- Editor, Page break: Color of the horizontal line indicating a page break. Page breaks may occur in your clips if you paste text that includes a form feed control character.
- Editor, Line breaks: Color used to draw line break symbols when you’ve turned on  Visualize Line Breaks.
- Editor, Whitespace: Color used to draw whitespace. The background color is always used if you set it to anything other than “default”. The text color is used to draw the space and tab symbols when you turn on  Visualize Spaces.
- Editor, Control characters: Color used in text mode to draw control characters other than tabs and line breaks. Such control characters normally should not appear in text clips.
- Editor, Margin and line numbers: Color used for the left margin when you turn on  Line Numbers.
- Editor, Extra space between lines: If the text layout adds extra space between lines then this color is used for that extra space. This can simulate the appearance of lined paper.
- Editor, Matching brackets: Color used to highlight matching brackets that do not contain any unmatched brackets. This color is also used to highlight the current byte in binary clips.
- Editor, Incorrectly nested brackets: Color used to highlight matching brackets that contain unmatched brackets.
- Editor, Unmatched brackets: Color used to highlight brackets that do not have a matching opening or closing bracket.
- Editor, Misspelled word: Misspelled colors are indicated using these colors and font style when using  Live Spell Check
- Editor, Highlighted search match: Used by  Highlight All.

The colors prefixed with “syntax” are the named colors used by syntax coloring schemes. Only the Editor palette has them. The descriptions given here are the suggested uses of these colors. The syntax coloring schemes provided with AceText strictly follow these suggested uses, though most schemes do not use all of the colors. Syntax coloring schemes created by other AceText users may use these colors to highlight other things.

- Syntax, Local link: Clickable link to a file on the user’s computer or local network. Only use this color for scheme elements with an action to open the file.
- Syntax, Internet link: Clickable link to a web page, file or email address. Only use this color for scheme elements with an action to open a file or URL, or start an email.
- Syntax, Emphasis: Text in a document that will appear in italics when the document is printed or published.
- Syntax, Strong: Text in a document that will appear bold when the document is printed or published.
- Syntax, Strong emphasis: Text in a document that will appear bold and in italics when the document is printed or published.

- Syntax, Underline: Text in a document that will appear underlined with the document is printed or published.
- Syntax, Redacted: Text that will not appear at all when the document is printed or published.
- Syntax, Label: An indicator that does not need emphasis, such as a fixed label before the actual data.
- Syntax, Caption: An indicator that can use some emphasis, such as a subsection header or a title.
- Syntax, Markup tag: Opening or closing tag in markup languages.
- Syntax, Markup tag attribute: Attribute name in markup languages.
- Syntax, Markup tag attribute value: Attribute value in markup languages.
- Syntax, Markup tag delimiter: Punctuation that starts or ends a tag in markup languages.
- Syntax, Markup entity: An entity name or a numeric character reference in markup languages.
- Syntax, Comment: Human-readable text used for information only.
- Syntax, Documentation comment: Human-readable text of particular importance, used for information only.
- Syntax, Preprocessor statement: Any kind of meta-information, such as compiler and preprocessor directives.
- Syntax, Reserved word: Words or character combinations with a specific meaning and specific use, such as keywords in a programming language.
- Syntax, Function name: The name of a (built-in) function call in a programming language, or a reference in a document.
- Syntax, Variable name: The name of a (built-in) variable in a programming language, or a placeholder for a changeable value or macro in a document.
- Syntax, Type name: The name of a (built-in) type or class in a programming language.
- Syntax, Constant name: The name of a (built-in) constant in a programming language, or a fixed placeholder in a document.
- Syntax, Constant value: A literal value that does not fit one of the following literal types.
- Syntax, Character: A single human-readable character (letter). Can also be used for escaped characters within character strings.
- Syntax, Character string: Human-readable text to be processed by software.
- Syntax, Text pattern: A text pattern such as wild cards or a regular expression.
- Syntax, Integer number: A whole number. Could be decimal, hexadecimal, octal, or binary.
- Syntax, Floating point number: A number with a fractional part and/or an exponent.
- Syntax, Date and time: Any date or time.
- Syntax, Address: Text defining the location of a file or server, such as an IP address or URI. Use this for scheme elements that do not have an action to open or navigate to the address.
- Syntax, Operator: Any kind of mathematical operator or other symbol with a specific meaning or effect.
- Syntax, Bracket: Round, square, or curly brackets used in expressions, such as parentheses and square brackets in C-style languages.
- Syntax, Structural brackets: Round, square, or curly brackets used to group statements or items, such as curly braces in C-style languages.
- Syntax, Section header: Heading that starts a new section.
- Syntax, Success message: Message in tool output, a log file, or a report indicating successful completion.
- Syntax, Hint message: Informative hint message in tool output, a log file, or a report.
- Syntax, Warning message: Non-fatal warning message in tool output, a log file, or a report.
- Syntax, Error message: Fatal error message in tool output, a log file, or a report.
- Syntax, Markup highlight: Highlight color for subschemes that highlight markup in files that do not predominantly consist of markup.
- Syntax, Code highlight: Highlight color for subschemes that highlight procedural code in files that do not predominantly consist of procedural code.

- Syntax, Fountain highlight 1 to 10: Can be used in specialized schemes to make up to 10 different kinds of items stand out, such as to highlight different kinds of entries in log files.

The colors prefixed with “regex” are used by the search and replace boxes to apply syntax coloring to regular expressions and replacement text. Only the Search palette has them.

- Regex, Syntax error: Any invalid combination of characters in regular expressions.
- Regex, Regular expression token: Characters with a special meaning in regular expressions not covered by any of the other colors.
- Regex, Escaped literal: A special character that was escaped to be treated as a literal character.
- Regex, Comment: A comment in a regular expression.
- Regex, Meta token: Characters that control the behavior of the regular expression but that do not match anything themselves.
- Regex, Character class brackets: The brackets around a character class in a regular expression.
- Regex, Character class literal text: Characters that are matched literally inside a character class in a regular expression.
- Regex, Character class token: Character with a special meaning inside a character class in a regular expression.
- Regex, Character class range: Characters that form a range inside a character class in a regular expression.
- Regex, Group (nesting level 1 to 5): Any pair of parentheses in a regular expression. Nested parentheses are highlighted in different colors to make it easier to see which opening parenthesis is paired with which closing parenthesis.

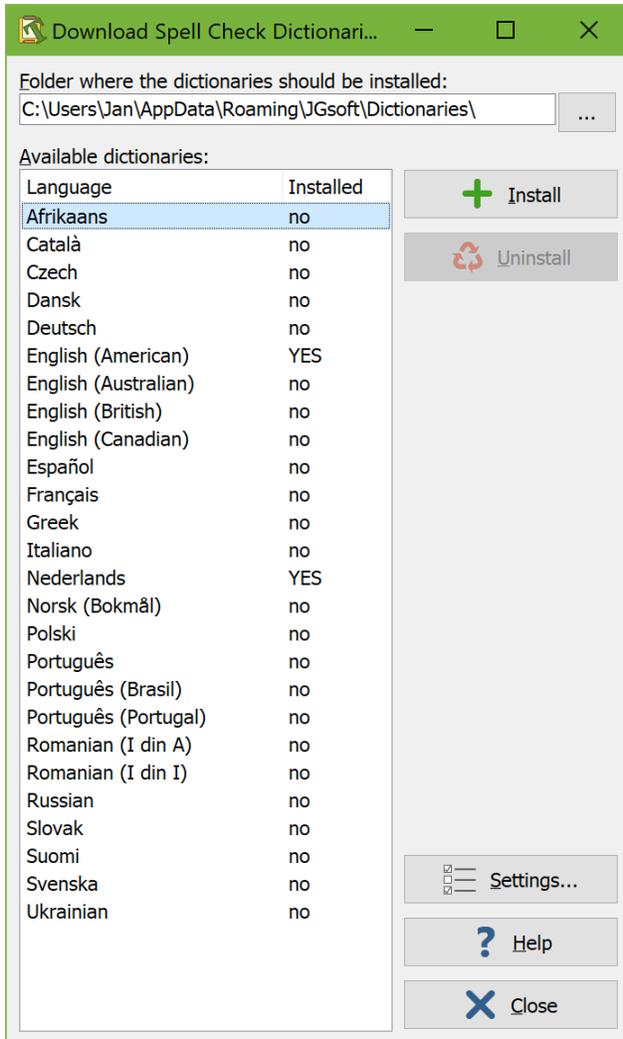
Example

When customizing an Editor palette, you can select one of the available syntax coloring schemes to see an example. Each coloring scheme has its own example text that shows the most important color elements of the scheme. You can double-click any text in the example to select the individual color that was applied by the syntax coloring scheme. The example does not necessarily show all color elements. You can type in or paste in your own example to test the colors. Your example won't be saved. If you select a different coloring scheme in the drop-down list then the example is reset to what is stored in the scheme.

When customizing a Search palette, the example shows a regular expression. The regex doesn't make any sense. It's just intended to show all the available regex colors. You edit the regex or paste in another one if you want a different example.

Download Spell Check Dictionaries

Before you can use  Live Spell Check, you need to download and install spell check dictionaries for one or more languages. You can easily do so right within AceText. On the Appearance tab in the Preferences, click the button “Download Spell Checker Dictionaries”. AceText will then connect to the Internet to fetch a list of available languages.



If you want, you can choose the folder into which the dictionaries should be installed. If you specify a folder that does not exist, AceText will create it. All dictionaries must be installed into the same folder. If you already installed some dictionaries, and then install another dictionary into a different folder, AceText will move the previously installed dictionaries to the new folder.

To install a dictionary, click on the language you want in the list, and click the Install button. AceText will then automatically download the dictionary, and install it into the folder you specified. A progress meter will appear while downloading.

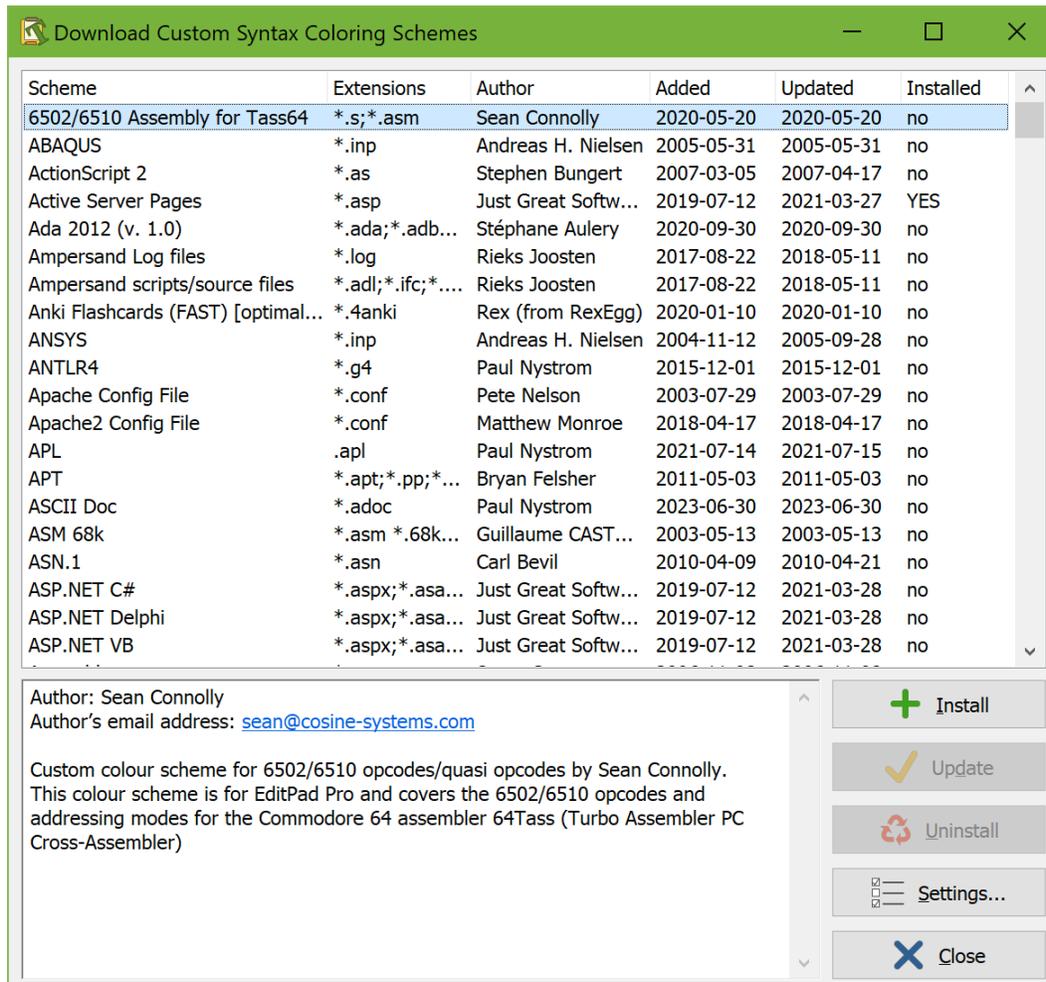
You can have as many dictionaries installed as you want. There is no need to uninstall dictionaries. However, if you want, you can select a language and click the Uninstall button to delete a spell check dictionary.

If you are behind a proxy server, and AceText is unable to detect your proxy settings, you can change them by clicking the Settings button.

Due to firewall and other network settings, it is possible that AceText cannot connect to the Internet directly. In that case, you can manually download and install the dictionaries from <https://www.acetext.com/spell.html>.

Download Custom Syntax Coloring Schemes

When you click on the button to download custom syntax coloring schemes on the Colors and Syntax page in the file type configuration, AceText connects to the Internet. It downloads a list of custom syntax coloring schemes that have been created and generously shared by other AceText users. This list is then displayed in the window shown below.



The list presents you the following information about the available schemes:

- **Scheme:** The name of the scheme. It is possible that more than one scheme with the same name is available. If so, you can install all of them or only one, as you prefer.
- **Extensions:** The extensions typically used for the files that the scheme provides syntax coloring for. They are only shown for your information. You can apply any scheme to any file type, as you see fit.
- **Author:** The author of the scheme. If you have comments or suggestions about a particular coloring scheme, you should contact this person.
- **Added:** The date the scheme was first added to the list of available schemes.
- **Updated:** The date the scheme was last updated.
- **Installed:** Indicates whether this scheme is installed on your computer or not.

If you click the **Install** button, AceText immediately downloads the scheme and saves it into the %APPDATA%\JGsoft\AceText 4 folder.

Click the **Update** button to download a scheme that you already installed again.

If you click the **Uninstall** button, AceText deletes the scheme from your computer when you close the window for downloading custom syntax coloring schemes.

If you are behind a proxy server, and AceText is unable to detect your proxy settings, you can change them by clicking the **Settings** button.

Syntax Coloring Scheme Editor

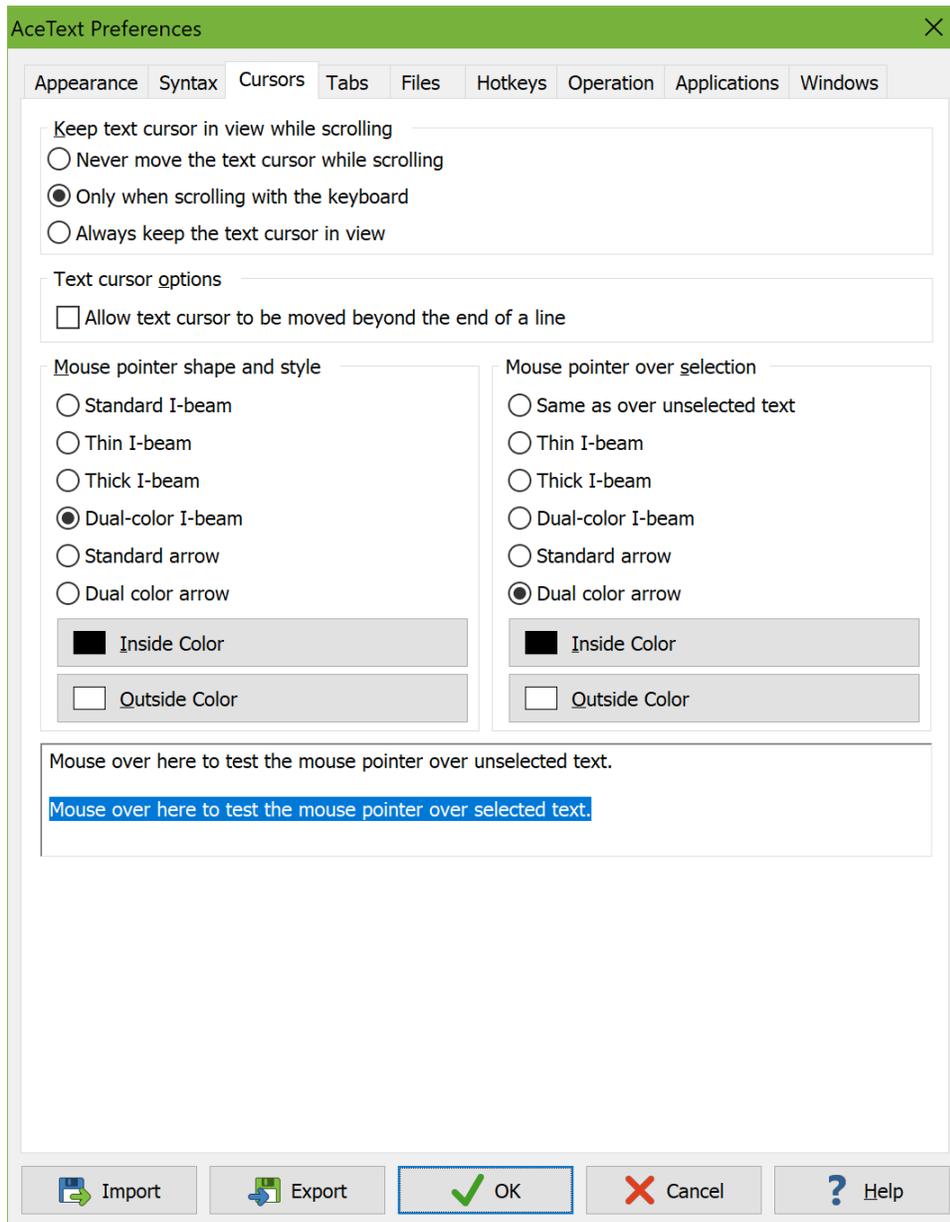
With the Syntax Coloring Scheme Editor, you can create your own syntax coloring schemes for use with AceText. You can edit the schemes included with AceText as well as those you've downloaded. The syntax coloring schemes files have a .jgscs extension. You can't edit them with any program except the Syntax Coloring Scheme Editor. You can download the scheme editor at <https://www.editpadpro.com/cscs.html>. The editor is a free download for all licensed AceText users. Full documentation is included. AceText 4 uses the same syntax coloring schemes as EditPad 8.

You only need to edit syntax coloring schemes if you want to change which text they highlight. For example, you could edit the syntax coloring for C/C++ to create a scheme for another C-style programming language. You don't need to edit the syntax coloring schemes if you only want to change the red/green/blue colors they use or which clips they are applied to. You can do that on the Syntax page in the Preferences. There you can customize the palette to change the colors that the syntax coloring schemes use. You can also change the file masks that are used to automatically assign syntax coloring schemes to clips created from files.

The syntax coloring schemes included with AceText are stored in the folder where you have AceText installed, typically C:\Program Files\Just Great Software\AceText. Windows's default security settings do not allow normal users to modify files under C:\Program Files. Therefore, schemes that you download and schemes that you edit are saved under your Windows user profile, typically C:\Users\YourName\AppData\Roaming\JGsoft\AceText 4. If a scheme with the same file name exists in both the user profile folder and the program files folder, only the scheme in the user profile folder will be available in AceText. Schemes loaded from your Windows user profile are marked with (user) in the drop-down list with syntax coloring schemes in the AceText Editor.

19. Cursors Preferences

On the Cursors tab in the Preferences screen, you can configure AceText’s mouse pointer and some aspects of the text cursor. The other text cursor settings are part of the Text Layout Configuration that you can select in the Appearance tab in the Preferences.



Keep Text Cursor in View While Scrolling

“Keep text cursor in view while scrolling” determines what happens to the text cursor when you scroll the text. Normally, the text cursor will keep its position relative to the text when you scroll the text. This may cause the text cursor to be scrolled off-screen. Depending on your text editing habits, this may be desirable or

undesirable. If you allow the text cursor to be scrolled off-screen, it will keep its position relative to the text. After scrolling, you can immediately continue typing at the same position. When you do so, the text will be scrolled automatically to make the cursor visible again. If you choose to keep the text cursor visible, it will be moved to the topmost or bottommost visible line when scrolling would otherwise have made it invisible. This way, you can always see exactly where the text cursor is pointing to.

By default, AceText will keep the text cursor visible when you scroll with the keyboard (by pressing Ctrl+Arrow Up/Down or Ctrl+Page Up/Down), but not when you scroll with the scroll bars or by rotating the mouse wheel. This is how most text editors for programmers work. AceText gives you the choice. Most general-purpose text editors do not allow keyboard scrolling at all.

Text Cursor Options

If you turn on “allow text cursor to be moved beyond the end of a line”, you can position the text cursor after the last character of the line. If you press the right arrow key when the cursor is at the end of the line, it will move one position to the right. If you click with the mouse beyond the end of the line, the cursor will be placed where you clicked. If you start typing when the cursor is beyond the end of the line, AceText will automatically fill up the line with spaces up to the position where you started typing.

When you do not allow the cursor to be moved beyond the end of a line, pressing the right arrow key when the cursor is at the end of a line will move it to the start of the next line. When you click the mouse beyond the end of the line, the cursor will be placed after the last character on the line.

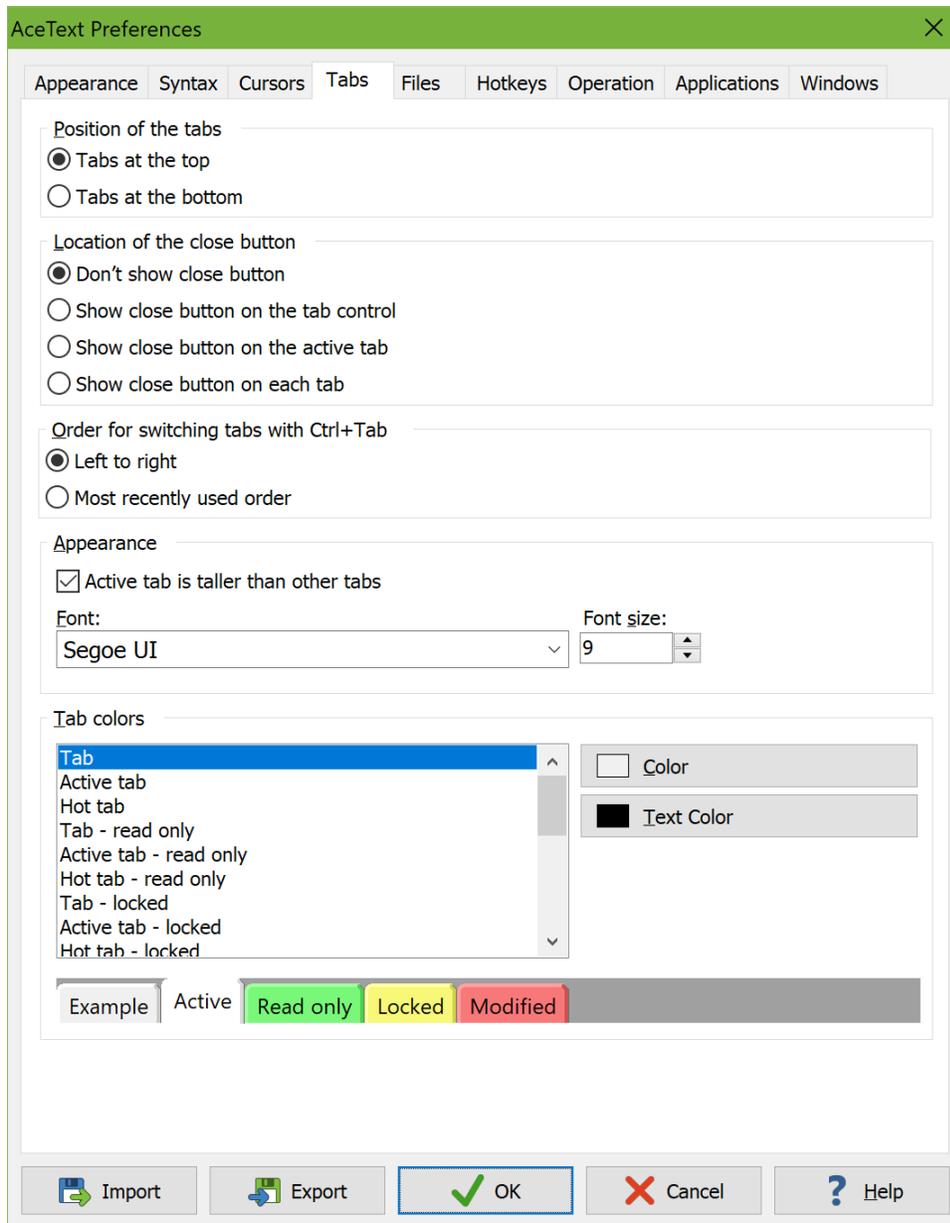
Mouse Pointer Shape and Style

AceText also allows you to select the mouse pointer shape and style. “Standard I-beam” is the standard mouse pointer for text editing controls, typically an I-shaped beam. “Thin I-beam”, “thick I-beam” and “dual-color I-beam” are custom AceText cursors for which you can pick your own colors. By choosing colors that contrast well with the background color you’ve chosen for the editor, you can make the mouse pointer highly visible. “Standard arrow” is the regular Windows mouse pointer. “Dual color arrow” is another custom AceText cursor in the shape of an arrow.

AceText’s custom pointers may not work in all situations. E.g. some remote desktop software cannot handle them. The mouse pointer may either be incorrectly displayed, or simply be invisible. In that case, simply select one of the standard pointers, which will always work.

20. Tabs Preferences

On the Tabs tab in the Preferences screen, you can configure the looks and behavior of AceText's tab pages.



You can choose to have tabs appear at the top, between the main toolbar and the collection toolbar. Or, you can have the tabs appear at the bottom, above the status bar.

AceText can show an X button to make it easy to close tabs. You can have one close button at the right hand edge of the tab control, after the last tab. Clicking this closes the active tab. Alternatively, you can have a close button directly on the tab. You can have it on the active tab only, or on each tab. A close button on each tab makes it easy to close tabs with a left click, but you have to be careful not to accidentally close tabs when switching between them.

If you choose not to show the close button, you can still close the active tab with the  Close Collection on the main toolbar. You can close any collection tab by clicking on it with the mouse wheel. Not showing the close button allows more tabs to fit on the screen.

The Clipboard, ClipHistory, All Collections, and Recycle Bin tabs that handle special functions in AceText cannot be closed. The X button is grayed out for these tabs and clicking it has no effect.

Pressing Ctrl+Tab on the keyboard switches between tabs. If you turn on “use most recent order when switching tabs with Ctrl+Tab”, then pressing Tab repeatedly while holding down Ctrl moves back through tabs you’ve previously activated, in reverse order. When you release the Ctrl key after pressing Tab one or more times, the tab that is activated is moved to the last position in the list of recently activated tabs. So if you press Ctrl+Tab and release both keys and do this repeatedly, you’ll be switching back and forth between two collections. Essentially, Ctrl+Tab switches between collections in AceText like Alt+Tab switches between application in Windows.

If you turn off “use most recent order when switching tabs with Ctrl+Tab”, pressing Ctrl+Tab moves through the tabs from left to right, and pressing Shift+Ctrl+Tab moves through tabs from right to left.

Appearance

The tab of the active collection has a different color than the other tabs. If this distinction is not clear enough, you can either change the tab colors as explained below or turn on “active tab is taller than other tabs”. When this option is selected, the active tab will have a larger height and a piece of background will be visible above all the other tabs. Turning off this option gives the tab control a much more compact look.

You can also change the font and font size used for the captions shown on the tabs. A bigger font is more readable but allows fewer tabs to fit on the screen.

Tab Colors

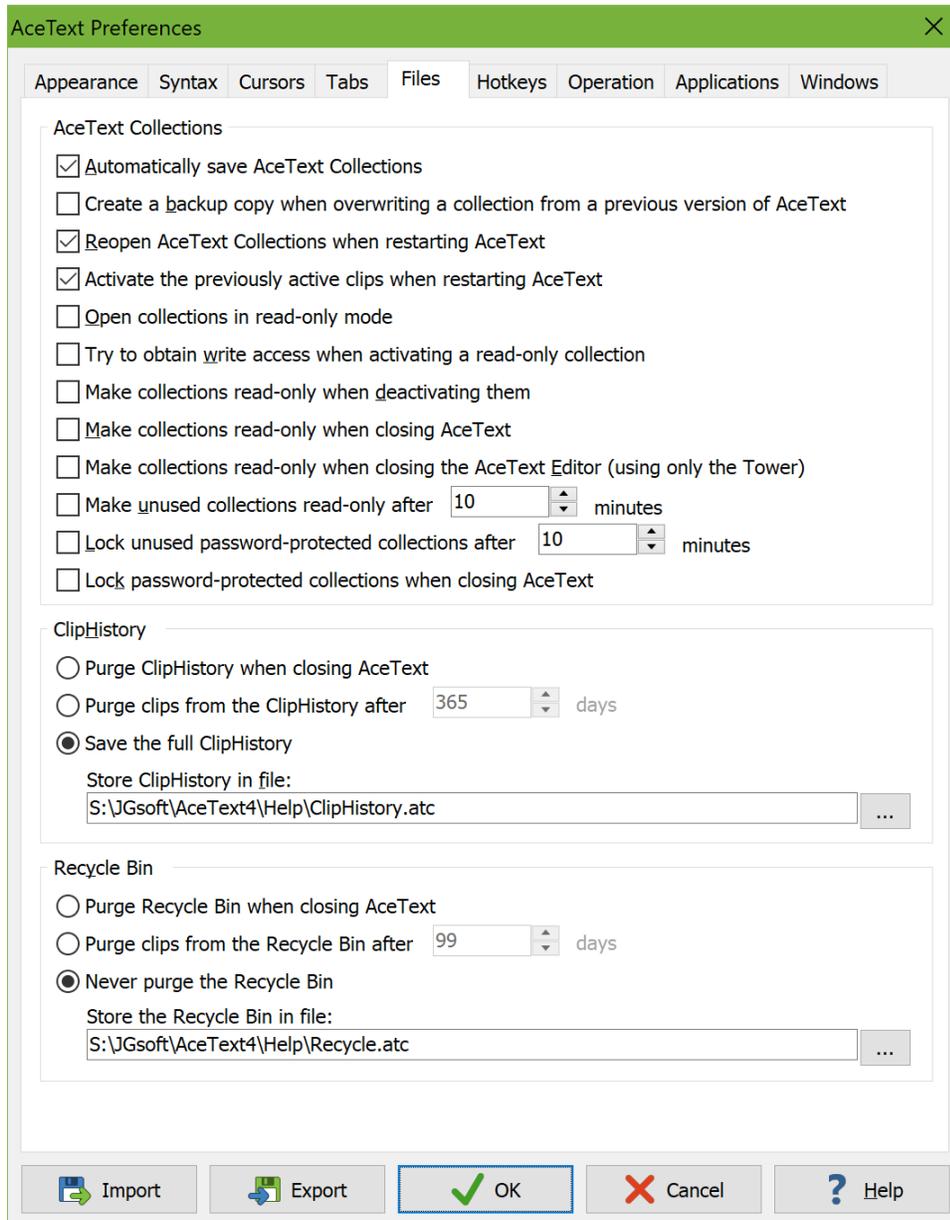
You can configure the colors of the tabs. Select an item in the list and click the Color and Text Color buttons to change the background and text colors of the selected tab style.

- Tab: Regular tab.
- Active Tab: The tab of the collection you are currently viewing.
- Hot Tab: The tab underneath the mouse pointer.
- Tab - read only: Tab of a read only collection.
- Active Tab - read only: The tab of the read only collection you are currently viewing.
- Hot Tab - read only: The tab of a read only collection underneath the mouse pointer.
- Tab - locked: Tab of a locked collection.
- Active Tab - locked: The tab of the locked collection you are currently viewing.
- Hot Tab - locked: The tab of a locked collection underneath the mouse pointer.
- Tab - modified: Tab of a modified collection (only used when automatic save is disabled).
- Active Tab - modified: The tab of the collection you are currently editing.
- Hot Tab - modified: The tab of a modified collection underneath the mouse pointer.
- Tab button: Color of the X button directly on a tab.
- Pressed tab button: Color of the X directly on a tab when you’ve clicked on it with the mouse.
- Hot tab button: Color of the X button directly on a tab underneath the mouse pointer.

- Tab control background: The empty space above and to the right of the tabs.
- Tab control button: Color of the X and other buttons at the right end of the tab row.
- Pressed tab control button: Color of the X or other button when you've clicked on it with the mouse.
- Hot tab control button: Color of the X or other button underneath the mouse pointer.
- Disabled tab control button: Color of disabled buttons at the right end of the tab row. The scrolling buttons will become disabled when you can't scroll further.

21. Files Preferences

Click on the Preferences button on the main toolbar to access the preferences screen. On the Files page, you can change how AceText saves your collections and the ClipHistory and Recycle Bin.



AceText Collections

By default, the option to automatically save AceText collections is on. AceText will then regularly save your collections, and make sure everything is saved when you close AceText, without asking. This way, you never lose any clips, and you never have to worry about saving.

If you turn off this option, AceText will ask you if you want to save modified collections when you close AceText. You will also need to click the Save button regularly when working for an extended period of time with AceText, to make sure you do not lose any clips in case of a power failure or system crash. If you are not sure whether to turn this option on or off, turn it on.

When you turn on “Reopen AceText Collections when restarting AceText” all collections that you had open when you last shut down AceText are automatically reopened the next time you start it. If you tend to work with the same collections every day, turn this option on. If you do, then you can also turn on “Activate the previously active clips when restarting AceText” so that each automatically reopened collection will be showing the same clip that it was showing when you last shut down AceText. Otherwise AceText shows the first item in each collection. This option does not affect collections that were not open when you last shut down AceText. If you open another collection then AceText initially shows the first item in the collection.

Turn on “Open collections in read-only mode” if you work with collection files that are stored on your local network and that are also used by other users, and most of the time you don’t modify the collections. Since only one user can have write access to a collection at any time, opening collections as read-only by default gives others the opportunity to gain write access if they need it. You can use the Read Only item in the Collection menu to modify a collection that was opened as read-only.

If you’ve turned on any of the options to automatically make collections read-only, then you may also want to turn on the option “Try to obtain write access when activating a read-only collection.” When this option is on, AceText will try to obtain write access to a collection when you click on its tab. If write access can be obtained, you’ll be able to modify the collection. If not, the collection will stay read-only. AceText will try to obtain write access again next time you switch to that collection. Turning on this option also makes read-only collections available as destinations for the  Duplicate Clips and  Move Clips commands.

The option “Make collections read-only when deactivating them” is the counterpart to the option above it. Whenever you activate a collection by clicking on its tab, all other collections are automatically made read-only. Any modifications you made to those collections are saved first. Turning on this option along with “open collections in read-only mode” gives others maximum opportunity to modify the collections.

When you close AceText by clicking the X button on its window caption, AceText doesn’t shut down completely. It keeps running in the background so that features like the ClipHistory and AcePaste can continue to work. AceText also keeps a lock on all collection files that aren’t read only, preventing other people and applications from modifying them. Since you can’t modify collections while AceText is closed anyway, turning on “Make collections read-only when closing AceText” gives others the opportunity to modify them when you’re not.

Your ability to edit collections is limited when using the AceText Tower without the AceText Editor. When using AceText this way, you’re probably only pasting text from existing collections rather than editing collections. Turning on “Make collections read-only when closing the AceText Editor (using only the Tower)” you can give others the opportunity to modify collections when you’re not editing them.

Turn on “Make unused collections read-only after X minutes” to automatically make collections read-only after a certain period of inactivity. The usage timer is reset whenever you modify the collection or select a clip or folder in the collection. This option can make the active collection read-only if you leave AceText unattended. It can make other collections read-only while you’re working with AceText. The main reason to turn on this option is to make sure you don’t block other people from modifying a shared collection that you have open if you leave AceText running and your computer doesn’t soon go to sleep when you leave it unattended. If your computer does go to sleep then AceText makes all collections read-only if you turned on

at least one of the 3 options to open collections read-only, to make collections read-only when deactivating them, or to make all collections read-only when closing AceText.

Turn on “Lock unused password-protected collections after X minutes” to automatically lock collections for which you previously set a password after a certain period of inactivity. The usage timer is reset whenever you modify the collection or select a clip or folder in the collection. Unused collections are automatically locked regardless of whether AceText is open or closed or whether the collection’s tab is the active tab or not.

Turn on “Lock password-protected collections when closing AceText” if you want to automatically lock all open password-protected collections when you close AceText. AceText keeps running in the background when you close it, showing its icon next to the system clock. The ClipHistory is not automatically locked, so AceText can keep capturing clips in the background. Regardless of this option, all collections are locked when you shut down AceText completely, and when your computer goes into standby or hibernation mode.

ClipHistory

By default, AceText automatically captures all text that is copied onto the clipboard into the ClipHistory. You can configure this in the Operation Preferences.

In the Files Preferences, you can determine if and where the ClipHistory is saved. If you select to purge the ClipHistory then AceText never saves it. Thus the ClipHistory tab will be empty the next time you start AceText. If you choose this option then you need to manually move any clips that you want to preserve from the ClipHistory to other collections. AceText does not warn that the clips in the ClipHistory will be lost when you close AceText.

The other two options save the ClipHistory into a file each time you close AceText and at regular intervals while AceText is running. Click the (...) button to select the location where you want AceText to save the ClipHistory. By default, it is saved in your “My Documents” folder.

By default, AceText saves the full ClipHistory. You can choose to purge clips that are older than a certain number of days. If you do so then AceText removes those clips when it loads the ClipHistory when you restart AceText. Purged clips are not moved to the recycle bin. If you keep your computer and AceText running 24/7 then the purge never happens. In that case, you need to rely on the setting for the maximum number of clips in the “automatic capture” section on the Operation page to keep the size of the ClipHistory manageable. That limit is enforced each time automatic capture needs space to add a clip to the ClipHistory. It is enforced even if you select to never purge the ClipHistory.

Recycle Bin

When you delete a clip, it is not really deleted but merely moved into the recycle bin. You can restore the clip from the recycle bin, until it is purged from the recycle bin.

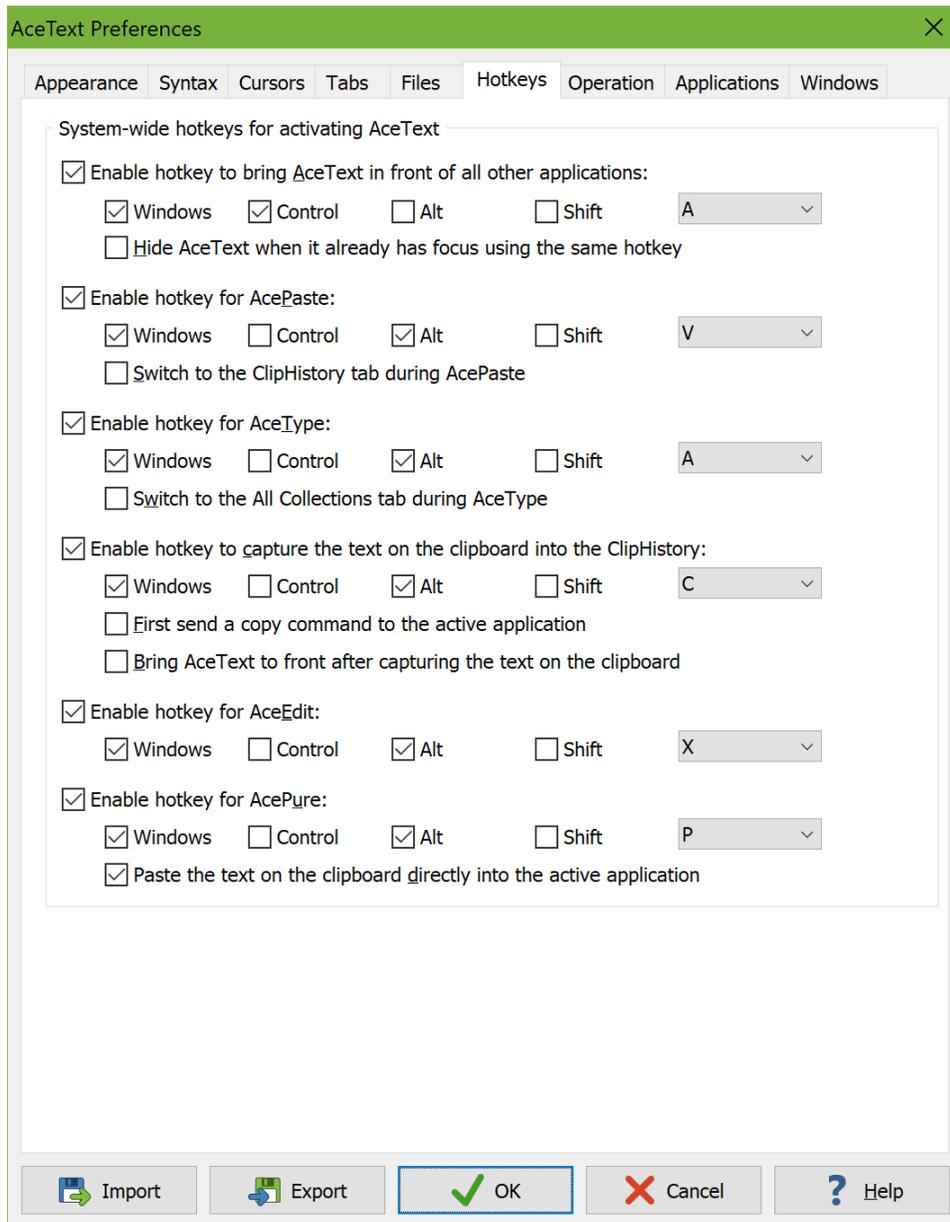
You can choose to automatically purge the recycle bin when you close AceText. With this option, you cannot recover deleted clips after you closed AceText. The benefit is that the recycle bin is not cluttered by old clips, making it more easy to restore a recently deleted clip.

The other two options save the recycle bin into a file each time you close AceText and at regular intervals while AceText is running. Click the (...) button to select the location where you want AceText to save the recycle bin. By default, it is saved in your “My Documents” folder.

If you select to purge clips after a certain number of days then AceText removes clips that were deleted more than that number of days ago when you restart AceText. The default setting is to purge clips after 7 days. You can change this to between 1 and 99 days. If you choose to never purge the recycle bin then you will need to clean it out manually every now and then, to prevent an unwieldy number of old clips from taking hold in the recycle bin. That is not a problem for AceText, but would make it difficult for you to find a recently deleted clip you want to restore.

22. Hotkeys Preferences

Click on the Preferences button on the main toolbar to access the preferences screen. On the HotKeys page, you can configure keyboard shortcuts to quickly activate AceText while you are working with another application.



The shortcuts or hotkeys only work while AceText is active in the background. Check if the AceText icon is visible next to the system clock. If not, you need to start AceText first. In the Operation Preferences you can make AceText start up automatically when you start your computer.

Since the hotkeys are system-wide, you need to make sure to choose key combinations that are not also used by other software that you use. Otherwise, AceText's hotkey will take precedence. Then you can no longer

use that key combination in the other software. Since applications normally don't use the Windows key, including the Windows key in AceText's hotkeys prevents conflicts with any other applications you may be using.

Windows itself reserves many key combinations with the Windows key. AceText cannot use those key combinations. If you try, the Preferences dialog shows the hotkeys that could not be used. You'll have to change or disable them before you can OK the dialog. Windows 8 and later reserve most Windows+Letter combinations. Windows 10 also reserves many Windows+Control+Letter and Windows+Alt+Letter combinations. The list seems to grow with each Windows 10 update. Some of the hotkeys used by Windows can be disabled, as explained below. If you disable them in Windows, then AceText can use them.

Make sure you don't try to use the same key combination twice. Doing so will also make AceText complain that it could not register that key combination. The second attempt to register the key combination will always fail.

Because AceText shouldn't override key combinations used by other applications, and because AceText can't override key combinations reserved by Windows, the default hotkeys are all Windows+Alt+Letter combinations. The letters are the traditional letters for cut (X), copy (C) and paste (V), along with "A" for AceText and "P" for "pure text". These letters should be easy to remember.

To enable one of the hotkeys, mark its checkbox in the hotkey preferences. To change the shortcut, mark the checkboxes of the modifier keys you want to be part of the hotkey, and select a letter or function key from the drop-down list. Combinations including any two of the Windows, Ctrl, and Alt keys are good choices. Few applications use the Windows key or Ctrl+Alt keyboard shortcuts, so these combinations are unlikely to cause conflicts.

Bring AceText to front

With this key combination, Win+Ctrl+A by default, you can quickly switch from the application you are using to AceText, without removing your hands from the keyboard. Without this key combination, you need to click on the AceText taskbar button or system tray icon (next to the system clock) to activate AceText. To close AceText, press Alt+F4, which is the standard Windows shortcut for closing a window.

When enabling this hotkey, you can also turn on the option to hide AceText with the same hotkey if you press it when AceText already has keyboard focus (i.e. it's already in front). This way you can quickly pop up and dismiss AceText at any time.

AcePaste

 AcePaste is a special AceText feature that enables you to quickly paste a clip into an application, and continue working with that application after pasting. The default hotkey is Win+Alt+V. When you press this hotkey, AceText pops up. Select the clip you want to paste, either by double-clicking on it, pressing Ctrl+Enter, or pressing the AcePaste hotkey again. The clip is then pasted directly into the application that was active when you pressed the AcePaste hotkey. If you change your mind, press the Escape key to cancel AcePaste. Either way, AceText brings to front the application that was active when you pressed the AcePaste hotkey, so you can continue working straight away.

By default, AceText tries to paste into the target application by placing the clip on the clipboard and simulating the Ctrl+V key combination. If you want to use AcePaste with an application that pastes differently, or can only accept keyboard input, you can configure that application in the Applications Preferences.

AceType

 AceType works just like AcePaste, except that you select the clip to be pasted by typing in its AceType abbreviation, and pressing Enter on the keyboard. If you took the time to assign abbreviations to clips, this is the fastest way to paste. Once you get the habit of using AceType, you will be able to paste blindly. Win+Alt+A is the default hotkey.

See Look up and Reuse Text to learn how to efficiently reuse text stored in AceText.

ClipHistory capture

If you turned off  Automatic Capture then you can enable a hotkey to manually capture text into the ClipHistory. When you press this hotkey, Win+Alt+C by default, AceText adds the text held by the Windows clipboard to the top of the ClipHistory. To transfer text from an application into AceText, first copy the text in the application, and then press the ClipHistory capture hotkey. You can also enable the option to bring AceText to front after capturing. That way, you can immediately move or duplicate the captured clip into a collection.

See Capture, Enter and Store Text to learn about other ways to get text from an application into AceText.

AceEdit

 AceEdit is a special AceText feature that enables you to quickly edit some text in another application with AceText, and continue working with that application when you're done editing. The default hotkey is Win+Alt+X. When you press this hotkey when working with an application other than AceText, AceText pops up. It shows the Clipboard tab, ready to edit the text that you selected in the application you're working with.

When you're done editing, press the AceEdit hotkey again, or press Ctrl+Enter, or click with the mouse to activate the target application. AceText then sends the modified text back to the application you were working with. If you change your mind, press the Escape key to cancel AceEdit. Either way, AceText brings to front the application that was active when you initiated AceEdit, so you can continue working straight away.

By default, AceText tries to extract the text from the application by simulating a Ctrl+C keystroke, and paste it back by simulating the Ctrl+V key combination. If you want to use AceEdit with an application that copies or pastes differently, you can configure that application in the Applications Preferences.

AcePure

 AcePure removes all clipboard formats other than plain text from the clipboard, leaving only “pure” text on the clipboard. Some applications call this “unformatted” text. The default hotkey is Windows+Alt+P. If you find an application is pasting formatted text when you want to paste only plain text, press the AcePure hotkey before pasting.

If you turn on “paste the text on the clipboard directly into the active application” then AceText also simulates a Ctrl+V key combination when you press the AcePure hotkey, immediately after it has reduced the clipboard to plain text. This allows you to paste plain text just by pressing the AcePure key combination. You can make it simulate a different key combination for pasting or have it simulate keystrokes to type the text on the clipboard into the target application by configuring the application you’re working with in the Applications Preferences.

Disable Specific Windows+Letter Keys in Windows Explorer

Most, but not all, of the Windows+Letter key combinations that are reserved for the operating system are used by Windows Explorer. AceText can add an entry to the Windows registry to tell Windows Explorer not to use certain Windows+Letter combinations. This then allows AceText and other applications to use those Windows+Letter combinations for their own system-wide hotkeys. Disabling a Windows+Letter combination in Explorer also frees up all Windows+Ctrl/Shift/Alt+Letter combinations with that letter.

If AceText detects that it cannot register one or more Windows+Letter key combinations when you click the OK button on the Preferences dialog then it shows a button “Disable Windows+Letter Hotkeys in Windows Explorer”. If you click this button then AceText presents you with this explanation:

Most, but not all, of the Windows+Letter key combinations that are reserved for the operating system are used by Windows Explorer. AceText can tell Windows Explorer not to use certain Windows+Letter combinations by adding an entry to the Windows registry. This then allows AceText and other applications to use those Windows+Letter combinations for their own system-wide hotkeys. Disabling a Windows+Letter combination in Explorer also frees up all Windows+Ctrl/Shift/Alt+Letter combinations with that letter. Windows+Letter key combinations used by other parts of Windows cannot be disabled.

If you have installed other applications that use system-wide hotkeys, then you’ll need to configure those applications and AceText to make sure there is no overlap. Otherwise the first application that registers the key combination as a system-wide hotkey will get it.

If you OK this dialog then AceText will add a registry entry to disable the following letters as system-wide hotkeys in Windows Explorer. You can enter up to 22 letters. They must be entered as uppercase. Lowercase letters have special meanings. The letters suggested by AceText when first showing this dialog are those you just tried to select on the Hotkeys page in the Preferences plus those that are listed in the existing registry entry, if any.

For the changes to take effect in Windows Explorer, you need to sign out from Windows and then sign in again.

If you OK this dialog then AceText adds the letters you entered in the dialog box as a string value named DisabledHotkeys under the key

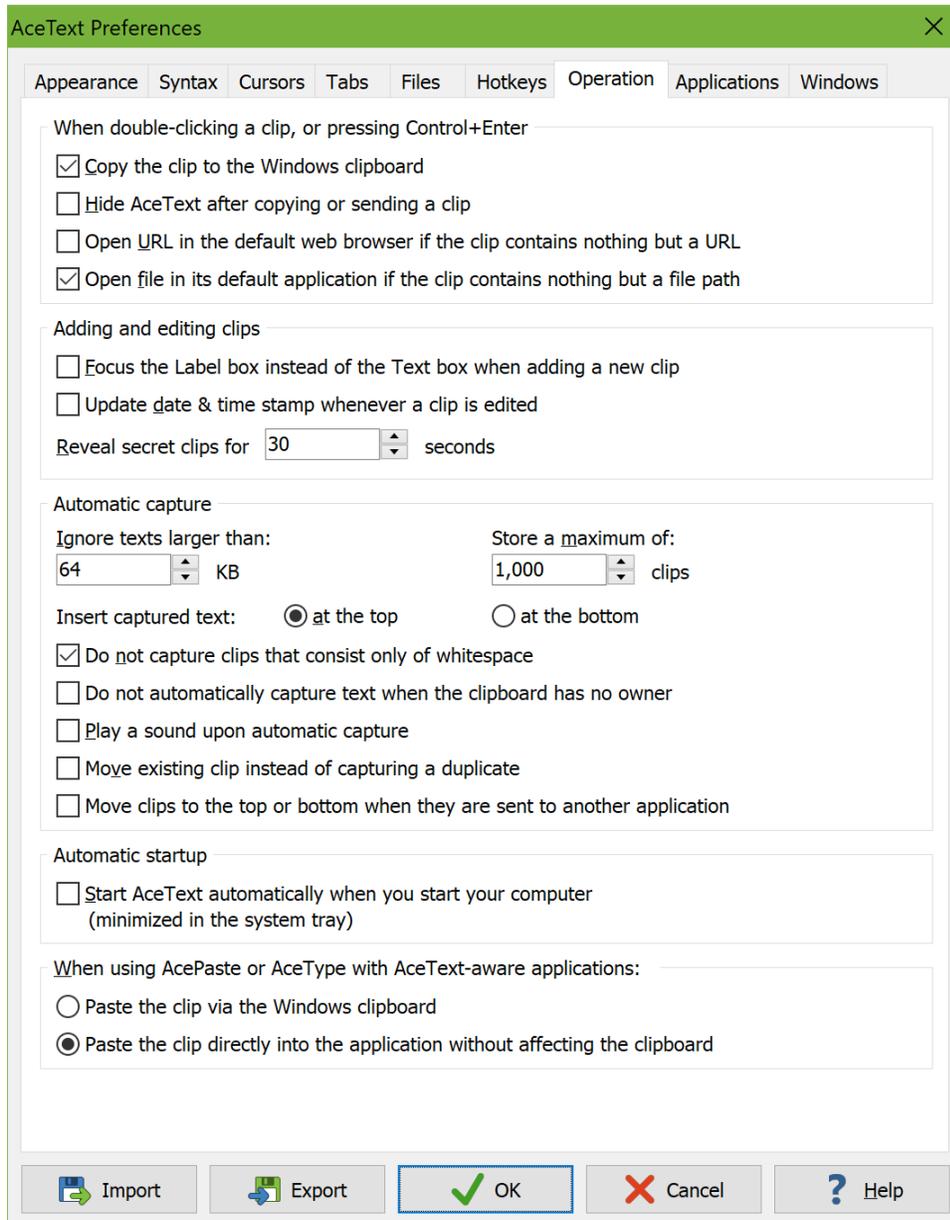
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced in the Windows registry. AceText does this immediately. But Windows Explorer doesn't check this value until you restart it. You can do so by signing out from Windows and signing in again.

After signing back in, check your hotkeys in the Preferences dialog in AceText again. If some still can't be registered then they weren't in use by Windows Explorer. Windows+X and Windows+P, for example, are in use by Windows itself. AceText can't use these key combinations (unless you add another modifier such as Alt).

If you want to use simple Windows+Letter key combinations for all six of AceText's hotkeys you could enter AVTCSQ as the letters to disable in Windows explorer. This lets you use Windows+A to bring AceText to front, Windows+V for AcePaste, Windows+T for AceType, Windows+C for capturing to the ClipHistory, Windows+S for AceEdit, and Windows+Q for AcePure.

23. Operation Preferences

Click on the Preferences button on the main toolbar to access the preferences screen. On the Operation page, you can configure how AceText interacts with other applications.



Double-clicking and Ctrl+Enter

Double-clicking and pressing Ctrl+Enter on the keyboard always have the same effect in AceText. You can perform the same action whether you prefer to use the mouse, or the keyboard.

If you turn on “copy the clip to the Windows clipboard” then AceText places the double-clicked clip onto the Windows clipboard. You can then paste it yourself wherever you want. This makes double-clicking a shortcut for the Copy to Clipboard button.

If you turn on “hide AceText after copying or sending a clip” then AceText hides itself if double-clicking a clip copied it to the clipboard (because you turned on the above option) or set a clip to another application (because you turned on  Double-Click Sends Clip on the main toolbar or in the Operation menu).

Turning on both the preference to hide AceText and the option to make double-clicking send a clip effectively puts AceText into AcePaste mode without having to press the AcePaste hotkey. But there still is one difference. When you press the AcePaste hotkey and then double-click a clip, it will be sent to the application that was active when you pressed the AcePaste hotkey, even if you switched to another application before double-clicking the clip. If you double-click a clip without pressing the AcePaste hotkey first, it is sent to the application that was last active (except for AceText itself, of course).

If the text of a clip consists solely of a URL, you can configure AceText to open that URL in the default web browser when you double-click on the clip in the clip tree. The URL will open in the browser without being copied to the clipboard and without being sent to the previously active application.

The last option does not affect double-clicking on the URL itself in the AceText editor. Double-clicking directly on a URL always opens it in the default web browser. The option affects double-clicking the clip’s label in the clip tree.

Adding and Editing Clips

Turn on “focus Label box instead of Text box when adding a clip” if you like to give all your clips a label when adding them. Turn it off if you prefer to type in the clip’s text straight away. If you don’t type in a label, AceText will use the first 40 character of the clip as the label.

Turn on “update date and time stamp whenever a clip is edited” to use the date and time field of each clip as an automatic “last modified” indicator. Whenever you make a change to a clip, its date and time will be set to the current date and time. Regardless of this setting, when you add a new clip, AceText will assign it the current date and time. You can change the date and time on a clip as you want. However, if the “update date and time stamp” option is on, if you make another change after setting the clip’s date or time, AceText will again automatically stamp the clip with the current date and time.

When you select a secret clip you need to click a Reveal button before you can see the contents of the clip. AceText automatically hides the clip’s contents again if you don’t edit them for a certain number of seconds.

Automatic capture

When you turn on  Automatic Capture on the ClipHistory toolbar the ClipHistory automatically captures all text that other applications copy to the clipboard.

To keep the size of the ClipHistory manageable AceText does not automatically capture very large blocks of text. By default, text longer than 64 KB (about 15 pages of text) is not automatically captured. You can increase this limit up to 32,767 KB which is 32 MB. This limit only affects the automatic capture. If you press

the ClipHistory capture hotkey or otherwise transfer text into AceText then AceText accepts any size of text. AceText can easily handle clips that are many megabytes in size.

To prevent the ClipHistory from amassing heaps of obsolete clips, AceText limits the number of clips stored in the ClipHistory. The default is 1,000 clips. You can adjust the limit between 10 and 100,000 depending on your preference not to keep around copied text any longer than really needed or your preference to be able to retrieve old clips well into the future without having to move them into another collection. The limit is enforced each time AceText automatically captures a clip and adds it to the ClipHistory. If after that number of clips is greater than the limit then AceText deletes the bottommost clips in the ClipHistory to make the number of clips in the ClipHistory equal to the limit. These clips are deleted permanently. They are not moved to the Recycle Bin. Increasing the limit does not recover clips that were deleted due to the previous limit. The limit is not enforced at any other time. So the ClipHistory can go over the limit when you manually add clips to it. This prevents surprises while you're organizing your clips. At the next automatic capture the limit is enforced again.

By default, AceText inserts newly captured text at the top of the ClipHistory as if the ClipHistory were a stack of items that grows upwards. You can choose to add newly captured text to the bottom of the ClipHistory instead. Then it will look like a list of items that grows downwards.

If you turn on “do not capture clips that consist only of whitespace” then the ClipHistory does not automatically capture clips that consist only of spaces, tabs, and line breaks. This affects only automatic capture. You can still paste whitespace into AceText regardless of this setting.

Normally, the Windows clipboard is “owned” by a window that is part of the application that most recently copied text to the clipboard. AceText can identify which application the window is part of. You can configure clipboard owners on the Applications and Windows tabs in AceText to selectively prevent or allow the ClipHistory to automatically capture the text they copy. But some applications deliberately try to hide their ownership of the clipboard by using a temporary window handle to copy text to the clipboard and then immediately destroying that window handle. The desktop application for the BitWarden password manager, for example, behaves this way. The window owner will already be gone when AceText gets to look at the clipboard and decide whether it should automatically capture the text into the ClipHistory. AceText thus has to resort to the settings for (any other application) on the Applications tab. But you may want to allow AceText to capture text copied by applications you haven't configured while still ignoring text copied by BitWarden. To do so, turn on “do not automatically capture text when the clipboard has no owner” on the Operation tab while allowing the ClipHistory for (any other application) on the Applications tab. If you do not allow the ClipHistory for (any other application) then the option “do not automatically capture text when the clipboard has no owner” has no effect.

AceText flashes its icon next to the system clock whenever it automatically adds something to the ClipHistory. If you want an auditory confirmation, turn on “play a sound upon automatic capture”. The default sound is the sound of a pair of scissors closing. You can change the sound by creating a file named ClipHistory.wav in the folder %APPDATA%\JGsoft\AceText 4 under your Windows user profile. AceText creates this folder the first time you use it. You can enter %APPDATA%\JGsoft\AceText 4 in the address bar in Windows Explorer to find the actual location of this folder on your computer. If you are using a portable installation, replace the file ClipHistory.wav in AceText's installation folder instead. AceText has no expectations about this file other than that it is named ClipHistory.wav and that it is a valid Windows PCM wave audio file that can be played by the Windows API.

When the same text is repeatedly copied to the clipboard, the ClipHistory captures only one instance of that text. When text is copied to the clipboard by another application that was captured by the ClipHistory before, but is not the most recently captured text, then the option “move existing clip instead of capturing a

duplicate” determines what happens. If you turned it on then the existing clip that previously captured the same text is moved to the top or bottom of the ClipHistory. This avoids having duplicate clips in the ClipHistory. If you turned it off then the existing clip stays where it is and a new clip with the same text is added to the top or bottom of the ClipHistory. This gives you a full history of what was copied, including non-consecutive duplicates. Turning on the option does not remove duplicates that were already in the ClipHistory. You can remove those with the  Delete Duplicate Clips button on the collection toolbar.

The ClipHistory never captures text that was copied by AceText itself. But you can make it work as a history for reused clips too by turning on “move clips to the top or bottom when they are sent to another application”. Whenever you use  AcePaste,  AceType, or  Double-Click Sends Clip to send a clip from the ClipHistory to another application, this option moves the clip to the top or bottom of the ClipHistory, depending on whether you set “insert captured text” to “at the top” or “at the bottom”. This option does not move clips that you manually copy and paste to another application with .

Automatic startup

AceText’s handy hotkeys only work while AceText is running. Windows does not allow inactive applications to register hotkeys. To make sure the hotkeys are always available, turn on the automatic startup option. This will make the AceText icon visible next to the system clock, and enable the hotkeys. AceText itself will not appear until you press a hotkey, click the tray icon, or start AceText via a shortcut icon.

If you also enabled the option in the Files Preferences to automatically reload your AceText collections, AceText’s automatic startup will add a few seconds to your computer’s startup time. The benefit is that your AceText collections are instantly available when you need them.

AceText-aware applications

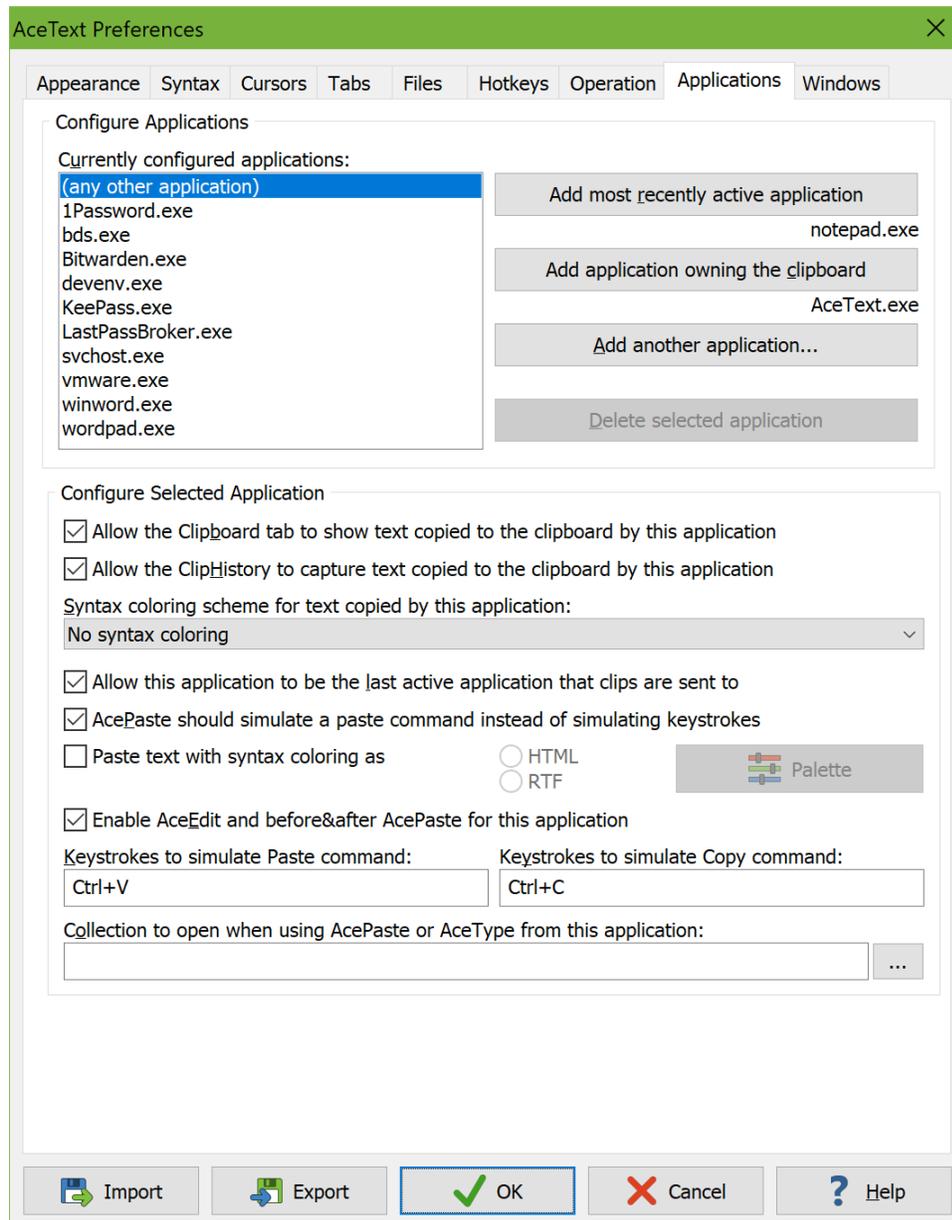
When you use AcePaste or AceType to paste directly into an application, AceText makes use of the Windows clipboard to send the clip into the application. This means that whatever data was held by the clipboard is lost. It also means that you can readily paste the same text again, since it is already on the clipboard.

If the target application is AceText-aware, however, AceText uses a different method that does not involve the clipboard to send the clip to the target application. This preserves the data that is held by the clipboard.

If you regularly use AceText with both AceText-aware and AceText-unaware applications, it may be confusing that sometimes AceText puts the clip onto the clipboard when you use AcePaste, and sometimes it does not. Therefore, you can select that AceText should also put the clip onto the clipboard when pasting into an AceText-aware application.

24. Applications Preferences

AceText's  AcePaste,  AceType and  AceEdit commands, as well as its ability to directly send clips to an application, require AceText to interact with other applications. To do so, AceText makes certain assumptions about the application. These assumptions are valid for most, but not all, Windows applications. On the Applications tab in the Preferences, you can configure how AceText should interact with specific applications.



Any Other Application

The topmost item in the list of currently configured applications is always “(any other application)”. The settings you make for this item are used when pasting into a window that is not configured in the Windows Preferences and is not part of an application configured here. Basically, these are your default settings that you can override by configuring a specific application or window.

Preconfigured Applications

Several applications are configured when you first use AceText. This includes the password managers 1Password, KeePass, and LastPassBroker. These are configured to make sure AceText does not accidentally reveal any of your passwords. If you use a different password manager then you should add that too and turn off the “Clipboard tab” and “ClipHistory” options. If your password manager is a browser extension then you won’t be able to configure it separately in AceText. The browser itself will be seen as owning the clipboard. To stop AceText from capturing text copied by browser extensions you need to configure it not to capture anything copied by the browser. You should also consider whether a password manager that transfers passwords via the clipboard is really safe. Just as you can configure AceText to ignore text copied by password managers, a piece of malware could do the opposite and specifically sniff out text copied by password managers. The Windows clipboard is designed to easily share data between applications, not to keep secrets.

A special preconfigured application is svchost.exe. This is executable is part of Windows’s internals. It owns the clipboard when Windows 10 synchronizes the clipboard with your Microsoft account. It is preconfigured to stop the ClipHistory from capturing text that you copied on another computer.

The configurations for Visual Studio (devenv.exe) and RAD Studio (bds.exe) select the C# and Delphi syntax coloring scheme for these applications. When AceText captures text from these programs into the ClipHistory or you paste text as a new clip from these applications then the pasted clip uses the C# or Delphi syntax coloring scheme.

VMware is configured to copy and paste via the clipboard using Ctrl+G Ctrl+V and Ctrl+G Ctrl+C key combinations. The extra Ctrl+G key combination is needed to make VMware direct input to the guest operating system after AceText activates the VMware window.

Microsoft Word (winword.exe) is configured to paste text with syntax coloring as HTML. While Word can also paste RTF, Word seems to interpret colors differently in RTF than AceText does. When using the HTML format, colors appear the same in Word as in AceText.

Microsoft WordPad is configured to paste text with syntax coloring as RTF. WordPad cannot paste HTML. WordPad interprets RTF colors in the same way as AceText does. But WordPad does not correctly handle palettes with the “Editor: plain text” colors set to “Default”. You should select a palette that uses specific colors for plain text. The default configuration for WordPad has AceText use the “harmonized shades print” palette which uses pure black on pure white for plain text.

Adding Applications

AceText identifies applications by the name of their executable file. The Notepad application included with Windows, for example, is identified by its file Notepad.exe which can be found in the c:\Windows folder.

You could configure how AceText interacts with Notepad by clicking the “Add another application” button, and selecting Notepad.exe.

Large applications developed with modular programming techniques may consist of multiple executables. AceText may identify the executable by a different one than the .exe you start it with. To configure such an application, switch to the application by clicking on it, and then switch back to the Preferences screen in AceText. Click the button “Add most recently active application” to configure the application you just briefly switched to. The application’s executable file name will be indicated below the button, and added to the list. This button uses the same method as the  AcePaste and  AceType commands to identify the target application for pasting.

The AceEdit button, the Clipboard tab and the ClipHistory use a slightly different method. They identify the application by checking which executable file last copied something onto the Windows clipboard. Normally this is the same .exe as the application switching method identifies. But it can be different for large, modular applications. To configure a particular application, simply switch to it, copy something to the clipboard in it, and switch back to the Preferences screen in AceText. Click the button “Add application owning the clipboard” to configure the application that you just told to copy something to the clipboard. The application’s executable file name will be indicated below the button, and added to the list.

Due to security restrictions imposed by Windows, AceText cannot identify applications that are running “as administrator” (unless AceText itself is also running as administrator, which we do not recommend). AceText indicates such applications with the (as administrator) placeholder. If you want to configure such applications, you’ll have to do so on the Windows tab in the Preferences.

It is possible for the clipboard to have no owner at all. AceText indicates that situation with the (none) placeholder. This is normal if you closed the application that copied the text that is still held by the clipboard. But some applications deliberately try to hide their ownership of the clipboard by using a temporary window handle to copy text to the clipboard and then immediately destroying that window handle before other applications like AceText can look at the clipboard. The desktop application for the BitWarden password manager, for example, behaves this way. This means you cannot use the settings on the Applications or Windows tabs to prevent AceText from capturing passwords copied by BitWarden. Instead you need to use the setting on the Operation tab to stop the ClipHistory from capturing text from all such applications.

Each application can be configured only once. AceText lists them in alphabetic order.

Clipboard Security

The Clipboard Tab in AceText allows you to directly view the contents of the Windows clipboard, and even edit them. There may be situations in which you’d rather not make the contents of the clipboard visible, in case somebody is looking over your shoulder. A particular example are password management utilities that help you remember passwords. Many of these copy the password to the clipboard, expecting you to paste it wherever the password is needed. Obviously, you don’t want AceText to show your passwords in plain view. Therefore, you should add your password manager to AceText’s list of configured applications, and turn off the “allow Clipboard tab to show text copied to the clipboard by this application” option. If an application is not configured for special treatment in AceText, the Clipboard tab will show whatever it copies to the clipboard.

You should also turn off the “allow the ClipHistory tab to capture text copied to the clipboard by this application” option for password managers and other applications copying sensitive stuff. By default, the ClipHistory tab captures any text you copy to the clipboard, and stores up to 1,000 items. You should also

turn off this option for applications that improperly use the clipboard as a temporary holding space, or from which you copy things you never need AceText to remember for you. This helps to declutter the ClipHistory, making it easier for you to find the important stuff that AceText should remember.

If you turn off both these options, then AceText assumes the application places sensitive data on the clipboard. If you use the  Paste as New Clip command while the application owns the clipboard then AceText pastes the text as a secret clip.

Syntax Coloring Scheme

You can select any syntax coloring scheme that you have installed as the default for the application. When AceText captures text that was copied by the application, whether that is automatically into the ClipHistory or manually by the  Paste as New Clip command, the clip will use this syntax coloring scheme.

You can download additional syntax coloring schemes via the button on the Appearance Preferences. You can create your own with the syntax coloring scheme editor.

Pasting into Applications

You can turn on  Double-Click Sends Clip on the main toolbar or in the Operation menu to make AceText paste clips directly into the most recently active application when you double-click the clips. By default, any application is fair game for pasting this way. If you have certain applications that you never paste into from AceText, you can configure them and turn off the option “allow this application to be the last active application that clips are sent to”. Then, double-clicking on a clip will never paste into that application. If it is the most recently active one, AceText will paste in the second most recently active one. (And if you disabled that one too, in the third most recently active, ad infinitum.). This option only affects the double-click or Ctrl+Enter direct sending. Even when it’s off, you can still use AcePaste, AceType or a simple paste command to paste AceText clips into the application.

When pasting directly in an application, whether because you double-clicked on a clip or used  AcePaste or  AceType, AceText can perform the paste in two ways. One way is for AceText to copy the clip to the clipboard, and then simulate a key combination to make the application paste the clip. For unconfigured applications, AceText will use this method, and simulate a Ctrl+V keystroke. If you turn on the option to use this method, you can also configure the keyboard shortcut AceText should simulate. This is the shortcut that you would use to manually paste into the application.

The other way to “paste” directly is for AceText to simulate keystrokes for the entire clip. AceText will activate the target application, and pretend that you typed in the whole clip’s text on the keyboard. The disadvantage of this method is that AceText can only send up to 5,000 characters this way, because Windows limits the number of keystrokes that can be simulated. This method also doesn’t support “before and after” clips. It will combine the “before” and “after” parts into a single block of text when sending such clips.

In the Hotkeys Preferences you can configure  AcePure to paste the text on the clipboard directly into the target application. If you do so then AcePure too uses the application’s settings to either simulate the keyboard shortcut for pasting or to simulate the keystrokes to type the text on the clipboard into the target application.

Paste Text with Syntax Coloring

When pasting a clip that uses a syntax coloring scheme or that has emphasis directly into an application, AceText can format the clip as HTML or as RTF to preserve the clip's colors or emphasis. This is only possible when pasting via the clipboard by simulating a Paste command. You can experiment with which format works best with each target application. Some support HTML or RTF but not both. Some do support both but give better results with one over the other.

If you do enable HTML or RTF you can also select a color palette that AceText should use to base the HTML or RTF colors on. You should select a palette that fits well with the colors used in the target application. When pasting into a word processor while working on a document intended for printing, for example, you'll probably want to select one of AceText's printing palettes that have dark text on a white background.

Grabbing Text from Applications

There are two situations in which AceText will try to grab text from an application. The first is when you use the  AceEdit command. AceText will simulate a keyboard shortcut to make the active application copy the selected text to the clipboard, so you can edit it on the Clipboard tab.

The other situation occurs when you try to paste a “before and after” clip into an application that is not AceText-aware using either  AcePaste,  AceType or the double-click method. Since the Windows clipboard can only hold one bit of text, and not two like “before and after” clips use, AceText can't paste such clips directly. Instead, AceText will simulate a keyboard shortcut to make it copy the selected text to the clipboard. When the application does this, AceText updates the text on the clipboard to put the “before and after” clip around it. Then AceText simulates the paste shortcut as it always does.

For unconfigured applications, AceText will simulate a Ctrl+C keyboard shortcut to make the application copy the clip.

Configuring Keystrokes

If you've turned on the AcePaste option, you can enter the keystrokes AceText should simulate to make the application paste the clipboard's contents. By default, this is the Ctrl+V keyboard shortcut, which is the standard shortcut for pasting in Windows applications. Shift+Insert is also a common shortcut. If your application doesn't have a keyboard shortcut, but does have an Edit|Paste menu item, you can enter Alt+E P as the keystrokes. Essentially, you need to enter the keystrokes that you would type when telling the application to paste. If an application can only paste using the mouse, you'll need to turn off the option to simulate the paste command.

If you've turned on the AceEdit option, you can also configure the keystrokes AceText should simulate to make the application copy the selected text to the clipboard. The standard Windows shortcut is Ctrl+C. Another common key combination is Ctrl+Insert. The Edit|Copy menu item is commonly accessed with Alt+E C.

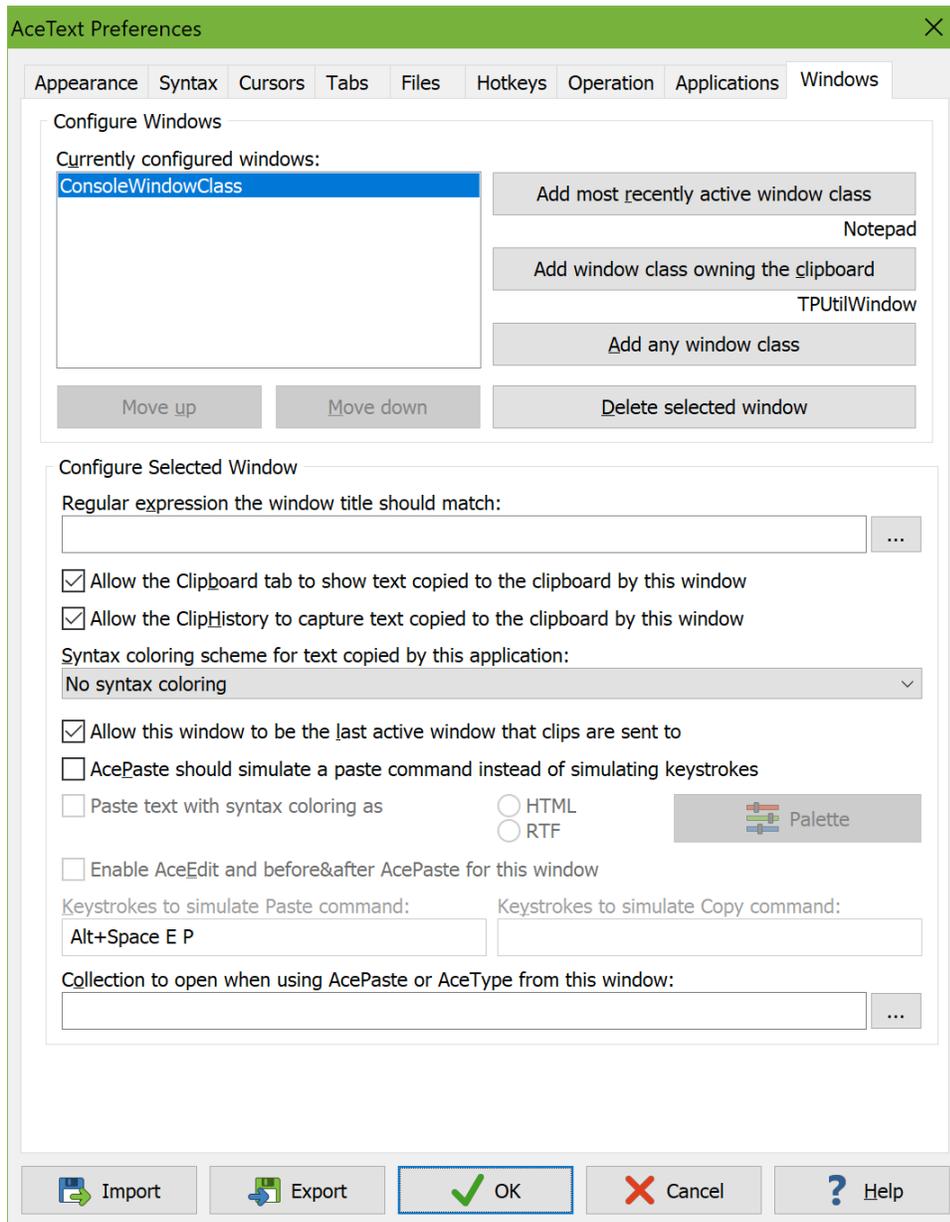
To enter the keystrokes, click on the edit field, and then simply press the keys. E.g. press Ctrl+V on the keyboard to enter the Ctrl+V key combination. If you press multiple key combinations, AceText will add them in sequence. Press Backspace if you pressed the wrong key.

Collection To Open When Using AcePaste or AceType

When pressing the  AcePaste or  AceType hotkey, AceText normally pops up showing whichever collection it was showing last time you were using it. You can make it open a specific collection when using AcePaste or AceType while working with an application. Simply specify the path to a collection that you have previously saved. You can click the (...) button to select the file. This way you can have one collection for each application, and always have the right collection ready for each application. You can make AceText open your HTML collection when using your HTML editor, for example.

25. Windows Preferences

AceText's  AcePaste,  AceType and  AceEdit commands, as well as its ability to directly send clips to an application, require AceText to interact with other applications. To do so, AceText makes certain assumptions about the application. These assumptions are valid for most, but not all, Windows applications.



Normally, you'll use the Applications tab in the Preferences screen to configure AceText to work correctly with your applications. The Windows tab provides similar settings, but configures top-level windows rather than applications. This is more complicated as applications may have multiple windows, but it can be more flexible if different windows in an application behave differently.

If a window matches a window configuration from the Windows tab in the preferences and that window is part of an application that you have configured on the Applications tab in the Preferences, then only the window configuration is used. The application configuration is only used for windows that do not match a window configuration. This system allows you to use the Applications tab to configure an application as a whole, while using the windows tab to use different settings for specific windows in the application.

Preconfigured Windows

The only window that's preconfigured when you first use AceText is `ConsoleWindowClass`. This is the window class used by the command shell that is part of Windows. The only special treatment it gets are different keystrokes for pasting (Edit|Paste in the system menu). The AceEdit option is disabled, since the command shell doesn't have a normal "copy" command.

Adding Windows

AceText identifies windows by their class names. Window class names are used by programmers when building Windows applications. The names are not necessarily meaningful and don't have to be unique between applications.

To configure a window, switch to the window by clicking on it, and then switch back to the Preferences screen in AceText. Click the button "Add most recently active window class" to configure the window you just briefly switched to. The class name will be indicated below the button, and added to the list. This button uses the same method as the AcePaste and AceType commands to identify the target application for pasting.

The  AceEdit button, the Clipboard tab and the ClipHistory use a slightly different method. They identify the application by checking which window last copied something onto the Windows clipboard. Many applications use an invisible helper window to manage the stuff they copy to the clipboard. Those windows often have generic class names used by many applications (developed with the same programming tools) and no captions. You're better off configuring such applications on the Applications tab. Applications developed with Delphi, for example, which include AceText and all our other applications, use an internal window named `TPutilWindow` for handling clipboard messages.

If you want to identify windows purely by the text displayed in their title bars, click the "Add any window class" button. This window configuration will apply to all windows in all applications. You should only do this when using a regular expression to check the window title.

You can add the same window more than once. If you do, you should use the window title regex to make sure only one window configuration applies to each window. If more than one configuration matches the class name and title, the one higher in the list is used.

Window Titles

Most applications indicate the name of the application and the active document in the title bar of their main windows. AceText can use a regular expression to check the window title. If the regex matches (part of) the title then the window configuration is used. If it does not then AceText checks the other window configurations.

If you simply want to check the title for some literal text, you can type that in normally. If the text includes symbols or punctuation, then you need to escape the characters `+*?.\[] () { }` with a backslash. E.g. the regex `\(I\.B\.M\.\)` matches `(I.B.M.)` literally.

If you have RegxBuddy installed, you can click the (...) button to edit the regular expression with RegxBuddy.

There are two main purposes for checking the Window title with a regex. The first reason is that dealing with window classes can be complicated. The regex allows you to use the “add any window class” button and then type in the name of the application as the regex to match the title against.

The other purpose is to make AceText sensitive to the application’s context. A plain text editor that you use to edit HTML and C++ files, for example, may indicate the name of the active file in its caption bar. If you add two window configurations for this application, one using the regex `\.html` and the other using the regex `\.(c|cpp|h|hpp)\b` then you can set your HTML collection to be opened when you’re editing HTML files, and your C++ collection when you’re editing C++ files.

Window Configuration

The actual configuration settings for windows are the same as the application configuration settings.

26. AceText-Aware Applications

AceText is designed as a sidekick to the applications you regularly work with to create documents and to communicate. AceText makes it quick and simple to store large and small bits of text for later reuse, and to transfer multiple bits of text between applications.

AceText is compatible with (almost) all Windows applications. However, some of its functionality will work better when used in conjunction with an application that was made AceText-aware by its developers.

What “AceText-aware” means

To make AceText’s tasks possible, AceText needs to be able to communicate with other applications. The Microsoft Windows platform provides several mechanisms to communicate. Two mechanisms that you, the user, have control over are the Windows clipboard and drag and drop. The Windows clipboard is a central storage area, where applications can store data and later retrieve it, whenever you use the “copy” or “paste” commands. Almost all Windows applications can use the Windows clipboard. Drag and drop enables you to directly copy or move data from one application to another, using the mouse. Since drag and drop is a bit difficult to implement for programmers, not all software supports it.

When two applications communicate, they need to agree in which format they will send and receive data. This is where the difference between applications that are AceText-aware, and those that are not, sets in. Microsoft Windows defines several standard data formats, including plain text. The standard plain text format allows AceText to capture text from and send text to any application, either via the clipboard, or via drag and drop.

However, AceText supports additional kinds of clips, such as rectangular text blocks and “before and after” text. Since those are not standard Windows data formats, only software that was made compatible with AceText by its programmers will recognize those data formats. If your software was not then AceText will detect this and fall back to the plain text format. Rectangular blocks and “before and after” clips are then sent as plain text.

AceText-aware software

At the time of this writing, only software published by Just Great Software (AceText’s publisher) is AceText-aware. This includes EditPad Lite, a basic text editor that is free for personal use, and EditPad Pro, a very powerful and convenient text editor. The combination of AceText and EditPad Lite or Pro definitely makes text editing a much more pleasant (or less boring) job, and certainly will reduce the time you spend doing it. See <https://www.editpadpro.com/> for more information.

How to make an application AceText-aware

Making an application AceText-aware can only be done by the company or people that created it. A single programmer can do it in a couple of hours. So you will need to request this functionality from the developers of the software that you use. If they have any questions about this, invite them to contact AceText technical support. We will be happy to assist them with the implementation.

If you are a programmer, and want to make your software AceText-aware, read the documentation on AceText’s clipboard format and AceText’s automation interface. At a very minimum, you need to make your

application recognize AceText's clipboard format, so your application can paste AceText's clips and receive them via drag-and-drop the way the user expects.

Implement AceText's automation interface to be able to directly respond to AceType and AcePaste events. This will make AceType and AcePaste work together with your application more reliably. Without the interface, AceText copies the clip to the clipboard and simulates a Ctrl+V keystroke in your application.

The automation interface also provides your application with great control over AceText. Your application can tell AceText to open or close a collection, retrieve its clips, add and update clips, etc. To see this in action, download a copy of EditPad Pro.

AceText Clipboard Format

The information below is intended for programmers who want to make their software AceText-aware.

The minimum requirement for an application to be AceText-aware, it to recognize AceText's clipboard format. Then the application can paste AceText's clips and receive them via drag-and-drop the way the user expects. Download the AceText 4 Client Demos to get sample source code that you can use as the basis for your own applications. One demo is written in C# using the .NET framework, and the other is written in Delphi using the Win32 API.

AceText places clips on the clipboard as a UTF-8 string holding an XML structure. AceText 4 uses the XML schema at <http://www.acetext.com/acetext40.xsd>. AceText 2 and 3 use the XML schema at <http://www.acetext.com/acetext20.xsd>. The structure is designed with both backward and forward compatibility in mind. For maximum compatibility, if your application encounters a clip with a `version` attribute in the `acetext` tag with a value greater than that defined in the XML schema, it should silently ignore any elements and attributes that are not defined in the schema.

When a single clip is copied to the clipboard, the string used to identify the clipboard format is "AceText Clip". Use this string when calling the Win32 API function `RegisterClipboardFormat()`, or when calling the `GetData()` and `GetDataPresent()` methods of the `IDataObject` interface in the .NET framework. The clipboard contents will be a UTF-8 string holding an XML file with a single `clip` element of type `Clip` in the `acetext` root element, as defined in the XML schema. An example:

```
<?xml version="1.0" encoding="UTF-8"?>
<act:acetext xmlns:act="http://www.acetext.com/acetext20.xsd" version="2.0">
  <clip kind="text" label="Clip Label" acetype="abbreviation" date="2006-06-
12T16:21:07">
    <text>Text of the clip copied to the clipboard</text>
  </clip>
</act:acetext>
```

When the user copies multiple clips to the clipboard, AceText still places the "AceText Clip" format on the clipboard. It will contain the clip that was visible in AceText when the clips were copied. If your application is unable to paste multiple clips, it should paste this clip.

AceText will also place the "AceText Collection" format on the clipboard. The data for this format is also an UTF-8 string with XML data. The `acetext` root element will contain one `collection` element of type `Collection`, as defined in the XML schema. This is the exact same format used by `.atc` files when AceText saves a collection into a file. An example:

```
<?xml version="1.0" encoding="UTF-8"?>
<act:acetext xmlns:act="http://www.acetext.com/acetext20.xsd" version="2.0">
  <collection sort="none">
    <clip kind="text" label="Clip Label" acetype="abbreviation" date="2006-06-12T16:21:07">
      <text>Text of the clip copied to the clipboard</text>
    </clip>
    <folder label="More clips" sort="old-new">
      <clip kind="beforeafter" label="Before & after clip" date="2006-06-12T16:28:13">
        <before>Before text</before>
        <after>After text</after>
      </clip>
      <clip kind="binary" label="Binary clip" date="2006-06-12T16:28:40">
        <bytes>DEADBEEF</bytes>
      </clip>
    </folder>
  </collection>
</act:acetext>
```

Automation Interface

The information below is intended for programmers who want to make their software AceText-aware.

In addition to adding support for AceText's clipboard format to your application, you should implement AceText's COM Automation to enable your application and AceText to communicate. Then, AceText can send AcePaste and AceType events directly to your application.

Demo Applications Implementing AceText's COM Automation

You can download the AceText 4 Client Demos to get sample source code that you can use as the basis for your own applications.

EditPad Lite and EditPad Pro are real-world examples of applications that fully integrate with AceText. If you have AceText installed, the Clip Collection panel in EditPad shows the collection that is active in AceText. This allows you to directly work with your AceText collection within EditPad.

Importing AceText's Type Library

AceText's installer automatically registers AceText's automation interface with Windows. The uninstaller automatically unregisters it. You can also do this manually. If you installed AceText for all users, open a command prompt with administrator privileges. Then run AceText with the `/regserver` or `/unregserver` parameter to register or unregister it. If you installed AceText for your own user account only, open a command prompt under your own user account and run AceText with `/regserverperuser` or `/unregserverperuser`.

To automate AceText via COM, you need to import its type library. It is stored in AceText.exe, which is installed under `C:\Program Files\Just Great Software\AceText` by default.

Delphi (Win32)

In Delphi, select Component|Import Component from the menu. Select “Import a Type Library” and click Next. Select “AceText API” version “4.0” from AceText.exe and click Next. Choose a palette page and a directory, make sure Generate Component Wrappers is ticked, and click Next. Install into a new or existing package as you prefer. A new component called TAceTextIntf appears on the component palette. This component implements the methods and events you can use to communicate with AceText.

Drop TAceTextIntf on a form or data module. Call the component’s Connect() method to connect to AceText. If AceText is not yet running, Windows will start it. Call the AceTextVersion method to ask which version of the API AceText supports. It may return 1, 2, or 4. Future versions of AceText may return a greater number. The number determines which TAceTextIntf methods you can call.

If AceTextVersion returns 4 or greater then call ClientVersion(4) to tell AceText that you support version 4 of the clip variant structure. Do not make this call if AceTextVersion returns 1 or 2.

To support AceText 1 or 2, double-click the OnQuerySupportedVersion event of the TAceTextIntf component in the Object Inspector. This event is only triggered if you don’t call ClientVersion before calling ConnectWindow. If you followed the above instructions then this can only be when AceTextVersion returned 1 or 2. Set the Version parameter in the event handler to the same value 1 or 2 to indicate you support version 1 or 2 of the clip variant structure.

In the OnShow event handler of all forms in your application, call TAceTextIntf.ConnectWindow(). Specify the Handle property of the form as the parameter. In the OnHide event handler of all your forms, call TAceTextIntf.DisconnectWindow(). This ensures that AceText knows which windows belong to your application. You should also assign the other event handlers of the TAceTextIntf component to make your application fully AceText-aware.

After calling ClientVersion or ConnectWindow you the TAceTextIntf methods that take an OleVariant as a parameter or return one. You must always use the version of the variant structure that you indicated support for. If you followed the above instructions then AceText will do the same.

C# (.NET framework)

In Visual Studio.NET, right-click on “References” in the Solution Explorer, and pick “Add Reference”. Switch to the COM tab, and choose “AceText API”. After adding the reference, import the AceText namespace. Then you can easily access the AceTextIntf interface. Create a new instance of this interface to connect your application to AceText. This will cause Windows to start AceText if it is not yet running. Call AceTextIntf.AceTextVersion() to ask which version of the API AceText supports. It may return 1, 2, or 4. Future versions of AceText may return a greater number. The number determines which TAceTextIntf methods you can call.

If AceTextVersion returns 4 or greater then call ClientVersion(4) to tell AceText that you support version 4 of the clip variant structure. Do not make this call if AceTextVersion returns 1 or 2.

To support AceText 1 or 2 create an instance of the delegate IAceTextEventIntf_ShutdownNotificationEventHandler and add it to the ShutdownNotification event of the AceTextIntf instance. The event is triggered when you call ConnectWindow and you didn’t call ClientVersion. If you followed the above instructions then this can only be when AceTextVersion returned 1

or 2. Set the Version parameter in the event handler to the same value 1 or 2 to indicate you support version 1 or 2 of the clip variant structure.

Whenever your application displays a window, call `AceTextIntfClass.ConnectWindow()`. For a WinForms application, specify the Handle property cast to a `uint` as the parameter. For a WPF application, instantiate a new `WindowInteropHelper` for your WPF window. Cast its Handle property to a `uint` and pass it to `ConnectWindow()`. When hiding the window, call `DisconnectWindow()` passing the same handle cast to a `uint`. This ensures that AceText knows which windows belong to your application. Casting an `IntPtr` to a `uint` is safe if the `IntPtr` is a top-level window handle. Windows ensures window handles never exceed 32 bits.

Automation Interface Reference

The information below is intended for programmers who want to make their software AceText-aware. Once you understand the basic principles of using AceText's COM automation interface, use the reference below to learn exactly how to handle the events AceText will fire, and which functions you can call to control AceText.

AceTextEventIntf Event Interface

AceText's event interface has four events. Your application must provide event handlers or event sinks for all of them.

QuerySupportedVersion(out uint Version)

If you do not call `ClientVersion` first then AceText fires this event when you call `ConnectWindow`. You must set the Version parameter to 1, 2, or 4 to indicate the version of the AceText Clip Variant Structure that your application supports. The version you request should not be greater than the version returned by `AceTextVersion`. If you request a greater version then AceText will actually use the version returned by `AceTextVersion`.

ShutDownNotification

This event is fired when the user attempts to shut down AceText while your application is still connected to it. AceText cannot shut down while an application is still connected to it. Your application should call `DisconnectWindow` for each window still connected, and then disconnect the COM automation link itself. If your application fails to do this then AceText cannot shut down and will minimize to the tray instead.

QueryAceType(uint Wnd, out BSTR AceType)

Fired when the user presses the AceType hotkey while the top-level window with handle `Wnd` has keyboard focus. Set `AceType` to the word the text cursor points to, or to the selected text. Whatever you assign to `AceType` is used as the default abbreviation. If a clip with the exact abbreviation exists, it will be pasted instantly. Set `AceType` to an empty string if your application cannot provide a meaningful default.

AcePaste(uint Wnd, OleVariant Clip)

Fired when the user completes an `AcePaste` or `AceType` operation. Paste the clip stored in the Variant into the active control of the top-level window with handle `Wnd`. If you fail to paste the clip, `AcePaste` will not work with your application. The clip may be a temporary clip that is not stored in any collection if AceText had to substitute any parameters in the clip. If your application needs to store the pasted clip it should store the Variant passed to `AcePaste` and not rely on the ID number in the Variant as that ID may not exist anywhere else.

CollectionChanged(unit CollectionID)

Fired when the user switches between collections in AceText or edits the active collection. Handle this event if you want your application to show a live copy of the collection that is active in AceText. Call GetCollection to actually retrieve collection.

AceTextIntf interface

AceText's automation interface provides many functions that you can call to provide tight integration between AceText and your software. However, for most purposes, you will only implement the four event handlers above, and only call the first two methods below. The remaining methods are for advanced integration. All JGsoft applications implement the basic AceText integration, but only EditPad Pro uses the advanced integration.

ConnectWindow(uint Wnd)

You must call ConnectWindow for each top-level window (i.e. form) your application shows. Only then will AceText know that this window is AceText-aware and call any of the events for that window. Calling ConnectWindow triggers the QuerySupportedVersion event if you did not first call ClientVersion.

DisconnectWindow(uint Wnd)

Call DisconnectWindow whenever you hide a window that you called ConnectWindow for.

AceTextVersion: int

This function returns the version of the API that the user's copy of AceText supports. It will also support all previous versions but those may not be able to store all the data that a clip may hold. AceText 1.x.x returns 1. AceText 2.x.x and 3.x.x return 2. AceText 4.x.x returns 4. Future versions may return 4 or a higher number.

StartAcePaste(uint Wnd)

Activate AcePaste, as if the user had pressed the AcePaste hotkey. Use this function to add an AcePaste button or menu item to your application. Set Wnd to the handle of the top-level window that has input focus when the user selects the AcePaste. AceText will send the pasted clip to that window. Make sure to specify the top-level window, and not the active edit control.

StartAceType(uint Wnd)

Works just like StartAcePaste, but initiates AceType instead.

ShowAceText(int Flags)

Make AceText show itself. The value of Flags determines whether the AceText Editor, the AceText Tower, or both are shown. 1 shows only the editor, 2 shows only the tower, 3 shows both editor and tower. 5 shows the editor, leaving the tower visible if it was visible last time. 6 shows the tower, leaving the editor visible if it was visible last time. If Flags is 0, AceText will show whichever windows were visible last time. Usually, you should specify 0.

HideAceText

Tell AceText to hide itself. AceText will only hide if it was made visible by your call to ShowAceText in the first place.

OpenCollection(BSTR Filename): uint

Opens the AceText Collection. Filename must be a full path. Returns the CollectionID for the collection, or zero if the collection could not be opened. If the collection is already open, its tab is simply activated. The collection will not be re-read from disk.

NewCollection(BSTR Filename): uint

Creates a blank AceText Collection. Either set Filename to a full path to use a default file name, or leave Filename blank. If you specify a default, and auto save it on, the collection will be automatically saved at regular intervals. Otherwise, the collection will not be saved until the user chooses a file name.

CloseCollection(uint CollectionID)

Closes the collection indicated by CollectionID. Set CollectionID to the value returned by OpenCollection or NewCollection. If autosave is off, and the collection was modified, the user will be asked to save it.

GetCollection(uint CollectionID): OleVariant

Returns an AceText Variant Structure holding the entire collection identified by CollectionID. Set CollectionID to the value returned by OpenCollection or NewCollection, or specify one of the standard collections. 0 retrieves the active collection, 1 retrieves the ClipHistory, 2 retrieves All Collections, and 3 retrieves the Recycle Bin.

CheckCollectionModified(uint CollectionID): DATE

Returns an OLE date indicating the time the collection identified by CollectionID was last modified. Use this function to avoid needless calls to GetCollection.

AddClip(uint ParentFolderID, uint PrevClipID, OleVariant Clip): uint

Adds the AceText Variant Structure (a folder with or without contents, or a clip) to an AceText collection. ParentFolderID must either be a CollectionID or a FolderID number. PrevClipID must be either zero, or the ID of a clip or a folder inside ParentFolderID. Returns the ID of the newly added clip or folder. Note that AceText will ignore any IDs specified on the Clip Variant. All the items in the Variant will get a new ID.

DeleteClip(uint ClipID)

Deletes the clip or folder identified by ClipID. If it is a folder, all clips and folders help by it are deleted as well. You cannot delete entire collections this way. Deleted clips and folders are NOT added to the recycle bin.

ReplaceClip(uint ClipID, OleVariant Clip): uint

Efficient way of deleting a clip or folder, and then inserting another one at the same position. It returns the ClipID or FolderID of the replacement. If the clip or folder ID that you pass in the variant structure is the same as that of the original clip or folder, then the replacement clip or folder keeps that ID. Otherwise ReplaceClip() generates a new ID just like AddClip() does. When replacing a folder that contains other clips or folders, those nested clips or folders keep their IDs if the Clip variant structure contains the same number of nested clips and folders and they have the same IDs. Link IDs can be preserved in the same way.

MoveClip(uint ClipID, uint ParentFolderID, uint PrevClipID)

Efficient way of deleting a clip or folder, and then re-inserting the same clip or folder at another position. Since the clip is moved, its ClipID and LinkID remain the same. MoveClip can move clips inside a single collection and between different collections.

CanModify(uint ClipID): uint

Returns 1 if the clip or folder identified by ClipID can be modified. Returns 0 otherwise. "Modifying" means editing the clip or folder itself, or adding clips or folders to the folder. You can modify all clips and folders, except those on the All Collections and Recycle Bin tabs.

CanContainFolders(uint FolderID): uint

Returns 1 if the folder or collection indicated by FolderID can contain other folders. Returns 0 otherwise. All collections and folders can contain other folders, except the ClipHistory. CanContainFolders returns 1 for the

All Collections and Recycle Bin tabs, even though you cannot add folders to them (because they can contain folders nonetheless).

GetActiveClip: uint

Returns the ClipID of the clip that is currently being displayed by AceText. Returns 0 if a folder is active, or if nothing is active.

GetActiveFolder: uint

Returns the FolderID of the folder that is currently selected in AceText. This means the user can readily edit the folder's label. Returns 0 if a clip is active, or if nothing is active.

SetActive(uint ItemID)

Activates the clip, folder or collection indicated by ItemID. If the item is inside a collection other than the active one, that collection is made active.

GetClip(uint ClipID): OleVariant

Returns the clip, folder or collection indicated by ClipID. In case of a folder or collection, all clips and folders inside it are also present in the returned variant structure.

New Functions in AceText 2.0.0

These functions are only available if AceTextVersion returns 2 or greater.

GetActiveCollection: BSTR

Returns the full path to the file where the active collection is saved. Returns an empty string when the active collection is unsaved, and when the Clipboard tab is active.

GetAllCollections: BSTR

Returns a list of all open collections, including special ones like the ClipHistory and All Collections. The list is delimited by line breaks. Each collection is indicated with a caption=id pair. The caption is what appears on the collection's tab in AceText. The ID is the value you can pass as the CollectionID parameter to GetCollection or SetActive.

ExpandParameters(uint ClipID): OleVariant

Looks up the clip indicated by ClipID. Returns that clip if it does not have parameters. Returns a copy of the clip with parameters substituted if it does. Prompts the user to supply values for custom parameters if needed. If AceTextVersion returns 4 or greater then you should call HasParameters before calling ExpandParameters. If you do not call HasParameters then ExpandParameters will still work correctly. But the Substitute Parameters dialog may get stuck behind other Windows. The user will then have to activate it by clicking on AceText's taskbar window.

New Functions in AceText 4.0.0

These functions are only available if AceTextVersion returns 4 or greater.

ClientVersion(ref int Version)

If AceTextVersion returns 4 or greater then you should call ClientVersion to indicate which version of the variant structure that you want to use. You should set Version to 4 to support all the clip features in AceText 4. But AceText 4 (and future versions) do allow you to set Version to 1 or 2. If you do so, AceText 4 will use

version 1 or 2 of the variant structure. You will still be able to call all the functions that AceTextVersion indicates support for. The Version argument is passed by reference to enable forward compatibility. Future versions of AceText may define API levels higher than 4. If you request a version that is greater than the version supported by the user's copy of AceText then AceText will change the Version parameter to the highest version that is actually available.

HasParameters(uint ClipID): int

Checks whether the clip identified by ClipID has an parameters. Returns 0 if the clip does not exist. Returns 1 if the clip does not have any parameters. You can then retrieve the clip by calling GetClip or ExpandParameters. Returns 2 if the clip has parameters but does not require a prompt. Call ExpandParameters to retrieve a copy of the clip with all parameters substituted. Returns 3 if AceText needs to show the Substitute Parameters dialog. In this case HasParameters makes AceText visible. Your application should call the SetForegroundWindow Win32 API function and pass it the result of AceText's GetWindowHandle function. Then call ExpandParameters to make AceText show the Substitute Parameters dialog. If AceText was hidden before you called HasParameters then ExpandParameters hides AceText again. ExpandParameters also calls SetForegroundWindow to reactivate the window that had keyboard focus when HasParameters was called.

GetWindowHandle: uint

If your application has keyboard focus and it wants to move keyboard focus to AceText then you should call the SetForegroundWindow Win32 API function and pass it the result of the GetWindowHandle function. If your application is 64-bit then it should convert the 32-bit unsigned integer returned by GetWindowHandle into a 64-bit handle by zero-extending the number so that the upper 32 bits of the 64-bit handle are zero and the lower 32 bits are the 32-bit number returned by AceText.

LinkIDtoString(uint64 LinkID): BSTR

Converts the LinkID that is present as an unsigned 64-bit integer in the variant structure of a clip or folder into a string of up to 11 base64url characters. You can create a minimal URL to point to the clip or folder by appending the result of LinkIDtoString to `acertext:clip/`. This string is also how Link IDs are stored in the XML of AceText collection files. LinkIDtoString does not check whether the LinkID actually exists. It simply converts the 8 bytes of the LinkID parameter in big endian format to base64url without any leading letters A.

LinkToClip(uint ClipID): OleVariant

Returns an URL pointing to the clip, folder or collection indicated by ClipID as would be generated by  Edit|Copy Link.

GetClipFromLinkID(uint64 LinkID): OleVariant

Returns the clip, folder or collection indicated by LinkID. In case of a folder or collection, all clips and folders inside it are also present in the returned variant structure. You should only look up clips by LinkID if your application needs to keep a reference to clips that persists between AceText sessions. Otherwise you should use GetClip to look up the clip by ClipID which is guaranteed to be unique within the AceText session but will change when the collection is closed and reopened.

GetClipFromLinkString(BSTR LinkString): OleVariant

Look up a clip clip, folder or collection using a string returned by LinkIDtoString or LinkToClip.

CanContainSecrets(uint FolderID): uint

Returns 1 if the folder or collection indicated by FolderID can contain secret clips. Returns 0 otherwise. Only collections that are encrypted with a password (and folders within those collections) can contain secret clips.

AceText Variant Structure

When communicating with AceText through its COM automation interface, your application will send and receive clips, folders and collections through an OLE Variant. The Variant is an array, with a dynamic number of elements. Each element is another Variant, which can potentially be arrays of their own.

In Delphi, you can use the Variant type, which is a dynamic type. Just use the array syntax such as `ClipText := MyVariant[5]` and the Delphi compiler automatically performs the proper casting.

In C#, the AcePaste event's Clip parameter is of type object. You need to explicitly cast the object into an object[]. Then you can access the elements in the array, which you will again need to cast explicitly. See the sample source code on how to do this.

When AceText passes a Variant to your application it uses the version of the structure reported by your application through the `ClientVersion` function (AceText 4) or the `QuerySupportedVersion` event handler (AceText 3 and prior) or the version reported AceText through the `AceTextVersion` function, whichever is less. Your application should do the same.

ClipID and LinkID

In all versions of AceText all clips, folders, and collections have an ID number that is guaranteed to be unique across all open collections during the current session of AceText. It is the first element (index 0) in every array in the variant structure. This is the number that you pass as the `ClipID`, `FolderID`, or `CollectionID` argument to AceText's functions to indicate the clip, folder, or collection that you want the function to work on. If a collection is closed and reopened then the collection itself and all clips and folders in it get new `CollectionID`, `ClipID`, and `FolderID` numbers. So you can use this to uniquely reference an item in AceText, but only while its collection is open in AceText.

In AceText 4 clips and folders additionally have an immutable `LinkID` number. When you add a clip or folder AceText generates a new semi-random `LinkID` for it that is guaranteed to be unique within the collection and across all collections that are open in AceText when the `LinkID` is generated. When you duplicate a clip or folder the duplicate gets a new `LinkID`. The `LinkID` is stored in the AceText collection file. Because collection files can be copied and edited outside AceText, there is no guarantee that the `LinkID` remains unique. Use the `LinkID` if you want to keep a reference to an item in a collection that persists with the collection. Only the `Link` and `GetClipFromLink` functions in AceText's COM automation interface use the `LinkID`.

AceText ignores the first element (index 0) and the `LinkID` element when you pass an `OleVariant` to a function. You should set both elements to 0. AceText will generate new ID numbers and ensure uniqueness. The only exception is the `ReplaceClip()` function which keeps the replaced clip's `ClipID` and `LinkID` unchanged if you put the same IDs that the clip already has in the `OleVariant` you pass to the `Clip` argument.

Variant Holding a Single AceText Clip

The AcePaste event handler will always send a Variant that holds a single clip. If you don't use advanced AceText integration, this is all your application needs to support. For a single clip, the Variant is a flat array. All elements must be present.

Version 4

Index	Type	Description
0	uint	ClipID number that uniquely identifies the clip during this AceText session.
1	int	Clip kind.
2	int	Flags.
3	BSTR	Clip label. Can be an empty string.
4	BSTR	AceType abbreviation. Can be an empty string.
5	BSTR or byte[]	The clip's text or the before text in case of a "before and after" clip or an array of bytes for binary clips.
6	BSTR	The clip's "after" text in case of a "before and after" clip. Empty string for all other clip types.
7	DATETIME	The clip's date and time.
8	BSTR	Syntax coloring scheme. Can be an empty string.
9	BSTR	URL that allows the clip to reference another resource. Can be an empty string.
10	uint64	Permanent LinkID number that can be used to create a URL pointing to the clip.
11	int[]	Array with emphasis change positions counting UTF-16 code points in the text with 0 being the position at the start of the text. NULL if the clip is not of kind "text with emphasis".
12	BSTR[]	Array with emphasis colors changed to at each change position. NULL if the clip is not of kind "text with emphasis".

Version 2

Index	Type	Description
0	uint	ClipID number that uniquely identifies the clip during this AceText session.
1	int	Clip kind.
2	int	Flags.
3	BSTR	Clip label. Can be an empty string.
4	BSTR	AceType abbreviation. Can be an empty string.
5	BSTR	The clip's text, or the before text in case of a "before and after" clip. Can be an empty string.
6	BSTR	The clip's "after" text, or the clip's URL, in case of a "before and after" or "web snippet" clip, respectively. Empty string for all other clip types.
7	DATETIME	The clip's date and time.

Version 1

Version 1 of the Variant for a single clip is the same as version 2 but with 7 elements instead of 8. The last element with the date and time is not supported. The "web snippet" kind is also not supported.

Clip Kinds

The second element (index 1) in the variant array is an integer indicating the kind of clip. Some clip kinds are only supported by some versions.

Number	Description	Versions
1	Plain text	All
2	Rectangular text	All
3	Binary data	All
4	Before and after text	All
5	Web snippet	Version 2 only
6	Text with emphasis	Version 4 only
99	Secret	Version 4 only

In AceText 2, a clip can only have an URL if it is of type "web snippet". In AceText 4, all clips can have an URL, making the "web snippet" clip type redundant. But AceText 4 still supports clip kind #5 if you use version 2 of the variant structure.

Flags

The third element (index 2) in the variant array is a bit mask setting certain options for the clip. The table below indicates all possible combinations of the bit mask.

Number	Description	Versions
0	None of the options are set	All
1	Indented clip	All
2	Clip with parameters	Version 4 only
3	Indented clip with parameters	Version 4 only

AceText 4 only expands parameters if the Parameters checkbox is ticked. AceText 3 and prior didn't have this checkbox. They always expanded parameters. If you use version 4 of the variant structure then you determine the state of the Parameters checkbox via the flags. If you use version 1 or 2 of the variant structure to send a clip to AceText 4 then AceText 4 ticks the Parameters checkbox if (and only if) the clip's text actually contains any parameters.

Link ID

The variant array for clips and folders stores the link ID as an unsigned 64-bit integer. You can convert this number to the ASCII string described in the [Edit|Copy Link](#) help topic using the following steps.

1. Store the ID as a big endian 64-bit number.
2. Convert the 8 bytes to 11 characters using base64url (which is the same as base64 but using - and _ instead of + and /) without padding.
3. Strip any leading A characters.

Variant Holding a Folder or Collection

The Variant that holds a folder is also an array. The total number of elements is variable. There is a fixed number of elements at the start of the array that must always be present. Their number depends on the version of the variant structure as listed in the tables below.

Then, for each element in the folder, the Variant has one element that is a variant array itself. In other words, a Variant defining a folder with two clips has 6 elements. The last two elements are variant arrays too. The array is not flattened. A Variant defining a folder with two clips has 6 elements, not $4 + 2 \cdot 8 = 20$ elements.

Version 4

Index	Type	Description
0	uint	FolderID number that uniquely identifies the folder during this AceText session.
1	int	Must be zero to indicate a folder. Use this element to check whether the Variant holds a clip or a folder.
2	BSTR	Folder label. Can be an empty string.
3	BSTR	URL that allows the folder to reference another resource. Can be an empty string.
4	uint64	Permanent LinkID number that can be used to create a URL pointing to the folder.
5	int	Number of items in the folder. Can be zero.

Versions 1 and 2

Index	Type	Description
0	uint	Unique session ID of the folder.
1	int	Must be zero to indicate a folder. Use this element to check whether the Variant holds a clip or a folder.
2	BSTR	Folder label. Can be an empty string.
3	int	Number of items in the folder. Can be zero.

Index

- .jgscs, 164, 165
- 1Password, 185
- abbreviation, 123
- accessibility software, 141
- AceEdit, 123
- AcePaste, 50, 122, 175
- AcePure, 124, 175
- AceText editor, 53
- AceText Editor, 52
- AceText Tower, 52
- AceText-aware, 193
- AceType, 50, 123, 175
- adapt case, 129
- add folder from disk, 64
- adjust case, 95
- adjust quotes, 97
- after text, 8
- all collections, 38
- alphabetic order, 49
- alphabetic sort, 112
- alphanumeric sort, 112
- always visible, 54
- apostrophe, 98
- appearance preferences, 141
- append, 61
- applications preferences, 184
- archive cliphistory, 82
- arrange, 12
- atom, 14
- auto adjust case, 95
- auto adjust quotes, 98
- auto append, 61
- auto indent, 41
- auto split, 62
- automatic capture, 60
- automatic save, 45, 171
- automation, 193, 195, 197
- aware of AceText, 193
- backtick, 98
- bds.exe, 185
- before and after text, 8
- binary data, 8
- blank lines, 116
- blind, 141
- blinking, 152
- blinking cursor, 141
- block, 55
- blockquote, 111, 112
- Borland colors, 155
- browse to url, 87
- browser, 87
- calendar, 134
- CanIncludeInClipboardHistory, 60
- capture, 60
- capture text, 8
- caret, 141
- case, 95
- case sensitive, 129
- CF_ACETEXT, 194
- CF_CLIPBOARD_VIEWER_IGNORE, 60
- character class, 65, 154
- characters, 85, 105
- check spelling, 54, 162
- clip, 8, 12, 51, 65, 66, 67, 68, 69, 70, 71, 72, 73, 85, 88, 90, 93
- clip menu, 85
- clipboard, 35, 72, 73, 90, 91
- clipboard format, 193, 194
- clipboard viewer ignore, 60
- cliphistory, 82
- ClipHistory, 36, 60, 171, 175
- clock icon, 121
- close, 125
- close collection, 44
- collection, 8, 12, 44, 45, 75, 80, 81, 82, 83
- collection menu, 75
- coloring, 10
- colors, 141, 155
- COM interface, 193, 195, 197
- combine, 12, 69
- comma-delimited values, 75
- comment block, 109
- ConsoleWindowClass, 191
- consolidate blank lines, 116
- convert, 95, 96, 97, 98, 100, 102, 103, 104, 105, 107
- convert menu, 95
- copy, 8, 66, 73, 91
- copy link, 88
- create clip from text file, 65
- create folder from disk, 64
- create portable installation, 120
- CSCS, 164
- CSV, 75
- cursor, 141
- cursor configuration, 150

cursor preferences, 166
 custom syntax coloring schemes, 164
 cut, 8, 72, 90
 dark theme, 118
 date, 91, 134
 delete, 71
 delete blank lines, 116
 delete collection, 83
 delete duplicate lines, 114
 delete duplicates, 71
 delimiters, 69, 70
 devenv.exe, 185
 diacritics, 21
 dictionaries, 54
 DisabledHotkeys, 178
 dot matches newline, 129
 Double-Click Sends Clip, 50
 down, 67
 downloading custom syntax coloring schemes, 164
 drag and drop, 8, 12
 duplicate clips, 66, 71
 duplicate lines, 114
 duplicate text, 90
 edit clip, 53
 edit menu, 88
 editor, 52, 53
 email, 108, 111, 112
 emphasis, 10
 empty recycle bin, 82
 encrypt, 81, 82
 enter text, 8
 ExcludeClipboardContentFromMonitorProcessing, 60
 exit, 125
 export collection, 75
 extract label, 70
 eyesight, 141
 feeds, 14, 17, 31
 file, 65
 file folder, 64
 file types, 153
 files, 8, 93
 files preferences, 171
 filter, 128, 134
 find, 126, 127, 128, 129
 find duplicate, 132
 find incremental, 132
 flag, 151
 flash card, 120
 focus clip tree, 53
 focus editor, 53
 focus tower, 53
 folder, 12, 63, 64
 folder from disk, 64
 font, 55, 141, 143
 format, 108, 109, 110, 111, 112, 113, 114, 116
 format menu, 108
 forum, 27
 generate password, 85
 generate random clip, 85
 globally unique identifier, 85
 grave accent, 98
 group clips, 69
 GUID, 85
 guillemet, 98
 gutter, 155
 hanging indent, 117
 help, 27, 119
 help menu, 119
 hidden icon menu, 121
 highlight all, 132
 highlight matches, 129
 highlighting, 10
 hotkeys, 175
 HTML, 75
 hue, 158
 icon, 121
 icons, 141
 incremental search, 129, 132
 indent, 41
 indent wrapped lines, 117
 indentation spaces, 104
 insert date and time, 91
 insert file, 93
 install onto removable drive, 120
 installing custom syntax coloring schemes, 164
 internet, 14, 17
 invert case, 95
 jgscs, 164, 165
 join, 12, 69
 KeePass, 185
 keyboard, 21, 138, 175
 keyboard focus, 53
 label extraction, 70
 LastPassBroker, 185
 left-to-right, 55, 143, 151
 light theme, 118
 limit line length, 54, 109
 line break indicators, 58
 line length, 54, 108, 109
 line numbers, 117
 lines, 113, 114, 116
 link to clip, 88
 live spell check, 54, 162

- lock, 80, 81, 82
- lock collection, 82
- long lines, 54, 109
- loop automatically, 129
- lowercase, 95
- luminance, 158
- MAC address, 85
- main toolbar, 44
- manage, 12
- menu, 75, 84, 85, 88, 90, 95, 108, 117
- merge clips, 69
- Microsoft Word, 185
- mnemonic, 123
- move clips, 66
- move collection, 75
- move down, 67
- move in reverse, 68
- move text, 90
- move to bottom, 68
- move to top, 68
- move up, 67
- natural sort, 112
- new clip, 8, 64
- new collection, 44
- new folder, 63, 64
- new subfolder, 64
- news feeds, 31
- newsfeeds, 14, 17
- notes, 8
- number lines, 117
- okina, 98
- on top of other apps, 54
- open collection, 45
- open file, 8
- open url in browser, 87
- operation menu, 84
- operation preferences, 180
- options, 55
- options menu, 117
- order alphabetically, 49
- page, 35, 36, 38, 40, 41
- palette, 155
- paragraph, 54, 108, 109
- paragraph indicators, 58
- parameters, 135
- password, 81, 82
- password generator, 85
- password managers, 185
- paste, 8, 60, 91, 122
- paste as new clip, 73
- paste in sequence, 51
- paste many clips, 51
- percentage signs, 135
- persistent selections, 55
- Pilcrow symbol, 58
- placeholders, 135
- plain text, 8
- portable installation, 120
- preferences, 141, 143, 155, 166, 168, 171, 175, 180, 184, 190
- primes, 102, 103
- primes => quotes, 103
- print, 45, 93
- PureText, 175
- purge, 71
- purge duplicates, 71
- quick paste, 122
- quick type, 123
- quit, 125
- quotation style, 104
- quote block, 111
- quoted text, 111, 112
- quotes, 96, 97, 98, 100, 102, 103, 104, 111, 112
- quotes => primes, 102
- RAD Studio, 185
- random characters, 85
- random clip, 85
- read only, 80
- read RSS, 14, 17
- recent collections, 45
- rectangular selections, 55
- rectangular text block, 8
- recycle bin, 40, 82, 171
- redo, 72
- reflow paragraphs, 108
- RegexBuddy, 133
- RegexMagic, 133
- regular expression, 69, 70
- regular expressions, 129, 133
- removable drive, 120
- remove, 71
- remove duplicates, 71
- remove quote characters, 112
- rename collection, 75
- reopen collection, 45
- replace, 126, 128, 129
- replace all, 129
- replace smart quotes, 102
- restore, 122
- reverse clips, 68
- reverse lines, 113
- rewrap paragraphs, 108
- right-to-left, 55, 143, 151
- ROT-13, 107

RSS feeds, 14, 17, 31
RTF, 75
saturation, 158
save, 171
save collection, 45
save text, 93
screen reader, 141
script, 55, 143
search, 126, 127, 128, 129
search duplicate, 132
search highlight, 132
search incremental, 132
search options, 129
search-and-replace, 129
secret, 85
separate, 69
SequencePaste, 51
set password, 81
shortcuts, 138, 175
shut down, 125
smart => straight quotes, 100
smart quotes, 98
sort alphabetically, 49, 112
sort alphanumerically, 112
sort collection, 49
sort collection tabs, 83
spaces, 57, 104, 113
spell check, 54, 162
split, 62, 90
split clips, 69
statusbar, 141
stay on top, 54
store text, 8
straight => smart quotes, 98
straight quotes, 100
substitution, 135
svchost.exe, 185
syntax coloring, 10, 164
syntax colors, 155
syntax highlighting, 10
syntax preferences, 153
system tray icon, 121
tab, 35, 36, 38, 40, 41
tab-delimited values, 75
tabs, 57, 83, 104
tabs preferences, 168
take notes, 8
terminate, 125
text, 8, 12, 19, 25, 51, 90, 91, 93, 111, 112
text cursor configuration, 149
text file, 65
text layout, 143
text menu, 90
text with emphasis, 10
theme, 118
time, 91, 134
to bottom, 68
to top, 68
toggle comment, 111
toolbar, 44, 59, 141
tower, 52
trash can, 40
tray icon, 121
trim whitespace, 113
TSV, 75
type library, 195
uncomment block, 110
undo, 72
ungroup clips, 69
unquote block, 112
untitled, 41
up, 67
uppercase, 95
url, 87
usb stick, 120
UUID, 85
variables, 135
variant, 202
visible, 54
Visual Studio, 185
Visual Studio colors, 155
visualize line breaks, 58
visualize spaces, 57
VMware, 185
web snippet, 8
whitespace, 57, 113
whole words only, 129
windows preferences, 190
Windows+Letter, 178
Windows+Letter keys, 175
winword.exe, 185
word wrap, 54, 109
WordPad, 185
wrap indent, 117
wrapped paragraphs, 108
write RSS, 14, 17