

CoolEdit

COMMAND HISTORY AND FILENAME COMPLETION UTILITY FOR WINDOWS

Help File Ver 0.359

Quick Start

Introduction

Command History

Filename Completion

Configuration - Cool Edit Box Properties Dialog Box

Advanced Configuration - cooledit.ini

System requirements and program limitations

Copyright and disclaimer

Quick Start

While CoolEdit is active it subclasses all Windows Edit Controls and provides the following extra features when an edit control has the focus.

<alt-enter>	displays the <u>Cool Edit Box Properties dialog box</u> where the behaviour of CoolEdit can be modified for the current edit control.
<up-arrow>	fills the edit control with the previous command from the history buffer.
<down-arrow>	fills the edit control with the next command from the history buffer.
<ctrl-up-arrow>	fills the an edit control that is part of a Combo Box with the previous command from the history buffer.
<ctrl-down-arrow>	fills the an edit control that is part of a Combo Box with the next command from the history buffer.
<space-bar>	<p>The first time <space-bar> is pressed a space is inserted in the edit control. Each subsequent space works through possible completions according to the following rules and in the following order:</p> <ul style="list-style-type: none">looks for lines in the history buffer that match the contents of the edit control up to the cursor position.if the cursor is in the first word in the edit control looks for a .EXE, .COM, .BAT or .PIF files in the current directory and the MSDOS path that match the contents of the edit control up to the cursor position.looks for directory names that match the word containing the cursor in the edit control up to the cursor position.looks for file names that match the word containing the cursor in the edit control up to the cursor position.
<ctrl-space-bar>	undoes previous word completions.

Individual history lists and properties are maintained for all edit controls and saved when CoolEdit is closed or Windows terminates.

Introduction

Remember MSDOS? Remember those great little utilities that used to remember what you typed at the C:> prompt and let you retrieve your previous commands without having to retype them?

CoolEdit does for Windows what those utilities used to do for DOS, it remembers everything you type in an edit box and gives it back to you when you need it. Because of the way it's written, using only the documented Windows API, it works within any application that uses standard edit boxes. The File|Run Dialog in Program Manager now has a history, so do the File|Open commands in Notepad, Write, Word, Excel and many other applications.

CoolEdit maintains a separate history for every edit box and you can configure individual boxes to take maximum advantage of the features available, you can even disable CoolEdit for particular boxes.

It doesn't stop there either! CoolEdit provides intelligent filename completion so just start typing a filename or directory, press the space bar twice and CoolEdit starts searching for the filename you want. Filename completion operates on local and network drives.

CoolEdit remembers your commands even when you leave Windows and switch your computer off.

Command History

Accessing the history list

Using the command history couldn't be simpler, just hit the up-arrow key to scan backwards or the down-arrow key to scan forwards through your previous entries in an edit box. If you reach the beginning of the history list CoolEdit wraps to the end, if you reach the end it wraps to the beginning.

You can also recover entries through word-completion, just type the first few characters of a previous entry, hit the space bar twice and CoolEdit searches through the history list for the first command that matches the characters you've typed, hit the space bar again it will find the next command that matches. If you get over-enthusiastic on the space bar and fly past the entry you want you can just press <Ctrl>-<SpaceBar> and CoolEdit will backtrack over the history.

The [Cool Edit Box Properties dialog box](#) contains a complete list of the history for a particular edit control, double clicking on one of the items in the list will replace the edit text with that item.

When items are added to the history list

Items can be added to the history list for an edit box when either the edit box is destroyed (for example when a you close a dialog box) or when the edit box loses the 'focus' (for example when you click on another control). The [Cool Edit Box Properties dialog box](#) allows you to configure CoolEdit to add items to the history list for a particular box when either the control is destroyed or loses the focus.

Deleting items from the history list

The [Cool Edit Box Properties dialog box](#) contains a complete list of the history for a particular edit control. To delete an item from the list highlight the item and then click the *Delete History Item* button.

Filename Completion

CoolEdit's filename completion features are designed for edit boxes where a filename or command is required. The [Cool Edit Box Properties dialog box](#) allows you to configure CoolEdit to use filename completion in a particular edit box.

See the entry for the <space-bar> key in the [Quick Start](#) for information on how to invoke filename completion.

Simple filename completion

Simple filename completion tries to match the characters that you have typed with a valid filename or directory on a local or network disk. CoolEdit looks at the current word and attempts to find an existing directory or file that matches the characters up to the cursor.

Command completion

Command completion operates on the first word in an edit control by attempting to find a file with a .COM, .EXE, .BAT or .PIF file in the current directory or any of the directories in the MSDOS PATH environmental variable that matches the characters up to the cursor.

cooledit.ini

The cooledit.ini file contains a number of sections that provide advanced configuration options. You should only modify the sections documented here.

[ComboBoxClasses]

[BadApps]

[Panic]

If the cooledit.ini becomes corrupted it may be restored from the [installed_cooledit.ini](#).

Installed cooledit.ini

If your cooledit.ini file becomes corrupt CoolEdit may fail when it tries to load. The default cooledit.ini is listed below, you can copy it to the clipboard and then save it to \windows\cooledit.ini from the clipboard viewer or a text editor. All history lists will be lost if you do this.

```
[ComboBoxClasses]
nComboBoxClasses=2
ComboBoxClasses0=ComboBox
ComboBoxClasses1=ThunderComboBox

[BadApps]
nBadApps=0

[Panic]
HelpFile=c:\cooledit\cooledit.hlp
CrisisFile=c:\windows\cooledit.log
HistoryLength=16
MinItemLength=3
ClassNameLength=32
DefaultFlags=7
DefaultHFlags=3

[EditBoxClasses]
nEditBoxClasses=1
EditBoxClasses0=Edit

[Control]
Diagnostics=0
Cools=0000
LongestString=32
```

[ComboBoxClass]

When searching for the parent window of an edit control CoolEdit will ignore any combo box controls. The entries in this section define the style names of controls that should be treated as combo boxes.

Entries

nComboBoxClasses=*n*

n is the number of class names in the section

ComboBoxClasses*n=foobar*

n is the ordinal number of the entry which should be between 0 and **nComboBoxClasses**-1. *foobar* is the Windows class name of the control.

[BadApps]

Some applications may hang when CoolEdit is active. The BadApps section list all applications which are incompatible with CoolEdit. Edit controls in these applications will not be subclassed. There are currently no know programs that are incompatible with CoolEdit.

Entries

nBadApps=*n*

n is the number of applications listed.

BadApps*n*=*foobar*

n is the ordinal number of the entry which should be between 0 and **nBadApps**-1.
foobar is the Windows module name of the application.

[Panic]

The [Panic] section of the cooledit.ini file contains a number of constants that are set by CoolEdit at installation time. You should only modify entries that are documented here.

Entries

HelpFile=foobar

foobar is the path of the CoolEdit Help File.

CrisisFile=foobar

foobar is the path of the Crisis File. If CoolEdit encounters a situation where it considers that it would jeopardise system stability it will write a diagnostic string to the crisis file and disable itself. After a crisis the About... dialog box will display 'CoolEdit has panicked'.

HistoryLength=n

n is the maximum number of strings that can a history list can hold, it must be an integer value between 1 and 1024. This option is most conveniently changed in the Option dialog box.

MinItemLength=n

n is the length of the smallest string that will be added to the history list. Strings containing less than *n* characters will be ignored.

DefaultFlags=n

n is the decimal representation of an integer derived by logically ORing the following hexadecimal values which correspond to checkboxes in the Cool Edit Box Properties dialog box.

0x01	<u>Enabled</u>
0x02	<u>Use history buffer</u>
0x04	<u>Use filename completion</u>
0x08	<u>Treat first word as command</u>
0x20	<u>Search current directory</u>
0x40	<u>Search in path</u>

The default flags are assigned to an edit control when it is first subclassed by CoolEdit.

DefaultHFlags=n

n is the decimal representation of an integer derived by logically ORing the following hexadecimal values which correspond to checkboxes in the Cool Edit Box Properties dialog box.

0x01	<u>Control is destroyed</u>
0x02	<u>Control loses focus</u>
0x04	<u>Case sensitive searchingDlgCase</u>
0x08	<u>Resolve filename</u>

The default flags are assigned to an edit control when it is first subclassed by CoolEdit.

Module name

The *Module name* control indicates the module name of the application that owns the current edit box. The module name is defined by the developers of the application.

The *Module Name*, *Parent Name* and *Item ID* are used to uniquely define an edit control.

Parent name

The *Parent name* control indicates the name of the window that encloses the current edit box. The parent name will usually be the title of the enclosing dialog box.

The *Module Name*, *Parent Name* and *Item ID* are used to uniquely define an edit control.

Item ID

The *Item ID* control indicates the numeric ID of the current edit box. The item ID is a unique integer defined by the developers of the application.

The *Module Name*, *Parent Name* and *Item ID* are used to uniquely define an edit control.

Enabled

The *Enabled* control indicates if CoolEdit is enabled for the current edit box. If the *Enabled* control is not checked CoolEdit will not operate on the edit box.

You may wish to use this checkbox to disable CoolEdit for an edit box that contains a password.

Use history buffer

The *Use history buffer* control indicates if CoolEdit maintains a history list for the current edit box. If the *Use history buffer* control is not checked CoolEdit will not remember your previous commands.

Use filename completion

The *Use filename completion* control indicates if CoolEdit should use filename completion when attempting to fill in the current edit box. If the *Use filename completion* control is not checked CoolEdit will not attempt to complete filenames in the edit box.

Treat first word as command

The *Treat first word as command* control indicates if CoolEdit should use command name completion in the current edit box. If the *Treat first word as command* control is not checked CoolEdit will not attempt to match the first word in the edit control to a .EXE, .COM, .BAT or .PIF file.

This control is only enabled if the *Use filename completion* control is checked.

You will probably want to enable this control in the Program Manager and File Manager Run dialogs.

Search current directory

The *Search current directory* control indicates if CoolEdit should look for commands in the current directory.

This control is only enabled if the *Treat first word as command* control is checked.

You will probably want to enable this control in the Program Manager and File Manager Run dialogs.

Search in path

The *Search in path* control indicates if CoolEdit should look for commands in the MSDOS path.

This control is only enabled if the *Treat first word as command* control is checked.

You will probably want to enable this control in the Program Manager and File Manager Run dialogs.

Case sensitive searching

The *Case sensitive searching* control indicates if CoolEdit should use case sensitive searching when attempting to complete words in the history buffer.

This control is only enabled if the *Use history buffer* control is checked.

Control is destroyed

The *Control is destroyed* control indicates if CoolEdit should add the text in the edit box to the history list when the control is destroyed. An edit box is usually destroyed when the dialog box containing the edit control is removed from the screen.

Either the *Control is destroyed* or the *Control loses focus* control box must be checked.

This control is only enabled if the *Use history buffer* control is checked.

Control loses focus

The *Control loses focus* control indicates if CoolEdit should add the text in the edit box to the history list when the control loses the focus. An edit box will lose the focus when the cursor bar (caret) moves out of the control and either another control displays the caret or a control becomes selected.

Either the *Control is destroyed* or the *Control loses focus* control box must be checked.

This control is only enabled if the *Use history buffer* control is checked.

Resolve filename

The *resolve filename* control indicates if CoolEdit should resolve a filename into a full path specification before inserting it into the history list.

This control is only enabled if the *Use history buffer* control is checked.

You will probably want to enable this control in most File Open dialogs.

Help

The *Help* Button launches Windows Help to display this file.

OK

The *OK* Button confirms the values in the Properties Dialog Box and closes the dialog.

About...

The *About* Button displays copyright and version information about CoolEdit.

Cancel

The *Cancel* Button closes the Properties Dialog Box and abandons any changes you have made to the controls.

History

The *History* list box displays the contents of the history buffer for the current edit control. Items are displayed in reverse chronological order. You can replace the contents of the edit control with an item in the history list by double clicking on the item.

This control is only enabled if the *Use history buffer* control is checked.

Delete History Command

The *Delete History Box* Button deletes the highlighted item in the *History* list box.

This control is only enabled if the *Use history buffer* control is checked.

Maximum history list size

This control indicates the maximum length of a history list. Values may be between 1 and 1024.

History lists are loaded when a control is created so the overhead associated with a large history list involves a small delay when an edit box is created.

Minimum length of history item

This control indicates the smallest string that will be added to a history list. Any string with less than this number of characters will not be added to a history list.

Help

The *Help* Button launches Windows Help to display this file.

Cancel

The *Cancel* Button closes the Options Dialog Box and abandons any changes you have made to the controls.

OK

The *OK* Button confirms the values in the Options Dialog Box and closes the dialog.

Version information

The information listed here is:

The name of the program.

The version number and compilation date.

The copyright notice.

Memory information

This control shows the approximate number of bytes of memory that have been dynamically allocated by the program.

Subclassed control information

This control shows how many edit controls are active compared to all the edit controls that the program knows about.

List of subclassed controls

This control lists all the known edit controls. The information given is:

The module name, this is the same as the *Module name* control in the Cool Edit Box Properties dialog box.

The parent windows name, this is the same as the *Parent name* control in the Cool Edit Box Properties dialog box.

The item ID, this is the same as the *Item ID* control in the Cool Edit Box Properties dialog box.

The window handle of the control, this is non-zero for active controls.

The control properties flags. The meaning of these flags is as listed for the **DefaultFlags** and **DefaultHFlags** entries in the [Panic] section of the cooledit.ini file.

Help button

The *Help* Button launches Windows Help to display this file.

OK button

The *OK* Button closes the about dialog.

Copyright

Copyright © 1993 Duncan Charlton. All rights reserved.

Disclaimer

THIS PRODUCT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHOR ASSUMES NO RESPONSIBILITY FOR ANY DAMAGE OR LOSS CAUSED BY THE USE OF THIS PRODUCT.

System requirements and program limitations

System Requirements

80386+ computer
Windows 3.1+

Program limitations

CoolEdit works by *subclassing* the Windows **Edit** class, so that all messages to an **Edit** control are passed to CoolEdit before being passed on to the control. For this reason the following limitations apply to CoolEdit.

CoolEdit will only work with controls which use the Windows **Edit** class. Controls created by applications which implement their controls independently will not be recognised or subclassed by CoolEdit.

In order for CoolEdit to subclass a control it must be active *before* the control is created. You should install CoolEdit in the 'StartUp' group in Program Manager so that it can subclass controls later.

Technical Notes

Windows, Procedures, Classes and Message Processing.

Every window with Windows has a section of code that deals with messages relating to events caused by the user,

Properties Dialog Box

Click on an area of the Dialog Box for more information....

Open this dialog by pressing <alt-enter> in an edit box.

Cool Edit Box Properties

Module name: **PROGMAN**
Parent name: **Run**
Item ID: **119**

Options

- Enabled
- Use history buffer
- Use filename completion
- Treat first word as command
- Search current directory
- Search in path

History

Resolve filename
 Case sensitive searching

Add to item history list when...

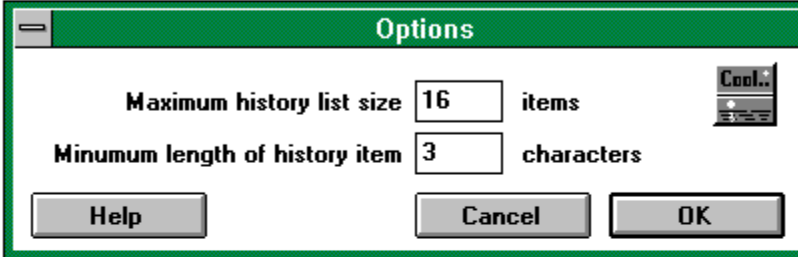
- Control is Destroyed
- Control loses focus

Buttons: Help, OK, About..., Cancel

Delete History Command

Options Dialog Box

Click on an area of the Dialog Box for more information....



The image shows a screenshot of a dialog box titled "Options". The title bar is green and contains a minus sign on the left and the word "Options" in the center. The main area of the dialog box is white and contains two rows of settings. The first row is "Maximum history list size" followed by a text input field containing the number "16" and the word "items". The second row is "Minumum length of history item" followed by a text input field containing the number "3" and the word "characters". In the top right corner of the dialog box, there is a small button labeled "Cool.." with a small icon below it. At the bottom of the dialog box, there are three buttons: "Help", "Cancel", and "OK".

Options

Maximum history list size items **Cool..**

Minumum length of history item characters

Help **Cancel** **OK**

About Dialog Box

Click on an area of the Dialog Box
for more information....

