

Upper Deck Editor Help Index

For information on how to use Help, press F1 or choose Using Help from the Help menu.

Overview

[License Statement](#)

[Features](#)

[Limitations and Known Bugs](#)

Default Key Assignments

[Cursor Movement](#)

[Selecting Text](#)

[Function Keys](#)

[Other Keys](#)

Menus

[File Menu](#)

[Edit Menu](#)

[Goto Menu](#)

[Search Menu](#)

[Macro Menu](#)

[Options Menu](#)

[Window Menu](#)

[Help Menu](#)

Functions

[about](#)

[alt_help](#)

[backspace](#)

[bc_error](#)

[bc_next_error](#)

[bc_prev_error](#)

[beginning_of_buffer](#)

[beginning_of_line](#)

[change_case](#)

[close_window](#)

[compare](#)

[copy](#)

[copy_block](#)

[copy_line](#)

[ctag](#)

[current_buffer](#)

[cursor_down](#)

[cursor_left](#)

[cursor_right](#)

[cursor_up](#)

[cut](#)

[cut_block](#)

[cut_line](#)

[delete](#)

[delete_line](#)

[delete_to_eol](#)

[delete_word_left](#)

[delete_word_right](#)

[editor_help](#)

[end_of_line](#)

[end_of_buffer](#)

enter
exit
exit_windows
find
find_again
find_files_containing
goto_line
goto_tag
indent_block
list_files_containing
lowercase
match_delimiter
msc_error
msc_next_error
msc_prev_error
next_window
new_filed
new_window
open_filed
page_down
page_up
paste
play_macro
prev_window
print
record_macro
replace
restore_window
revert
run
save_all
save_all_exit
save_all_exit_windows
save_file
save_file_close_window
sdk_function_help
sdk_help
select_all
select_char_left
select_char_right
select_line
select_line_down
select_line_up
select_page_down
select_page_up
select_to_beginning
select_to_bol
select_to_end
select_to_eol
select_word
select_word_left
select_word_right
show_message
split_line
stamp

tab_right
to_bottom
to_center
to_top
toggle_backup
toggle_insert
toggle_wordwrap
top_of_window
tpw_error
undo
unindent_block
unmark_block
uppercase
window_down
window_up
word_left
word_right
wrap_para
zoom_window

Cursor Movement

Left arrow	<u>cursor_left</u>
Right arrow	<u>cursor_right</u>
Up arrow	<u>cursor_up</u>
Down arrow	<u>cursor_down</u>
Ctrl+Left arrow	<u>word_left</u>
Ctrl+Right arrow	<u>word_right</u>
Ctrl+Up arrow	<u>window_down</u>
Ctrl+Down arrow	<u>window_up</u>
PgUp	<u>page_up</u>
PgDn	<u>page_down</u>
Ctrl+PgUp	<u>top_of_window</u>
Ctrl+PgDn	<u>bottom_of_window</u>
Home	<u>beginning_of_line</u>
End	<u>end_of_line</u>
Ctrl+Home	<u>beginning_of_buffer</u>
Ctrl+End	<u>end_of_buffer</u>

Selecting Text

Shift+Left arrow	<u>select_char_left</u>
Shift+Right arrow	<u>select_char_right</u>
Shift+Up arrow	<u>select_line_up</u>
Shift+Down arrow	<u>select_line_down</u>
Ctrl+Shift+Left arrow	<u>select_word_left</u>
Ctrl+Shift+Right arrow	<u>select_word_right</u>
Shift+PgUp	<u>select_page_up</u>
Shift+PgDn	<u>select_page_down</u>
Shift+Home	<u>select_to_bol</u>
Shift+End	<u>select_to_eol</u>
Ctrl+Shift+Home	<u>select_to_beginning</u>
Ctrl+Shift+End	<u>select_to_end</u>
Esc	<u>unmark_block</u>

Function Keys

F1	<u>editor_help</u>
Alt+F1	<u>sdk_help</u>
Shift+F1	<u>sdk_function_help</u>
F2	<u>save_file</u>
Shift+F2	<u>stamp</u>
F3	<u>open_file</u>
F4	<u>current_buffer</u>
Alt+F4	<u>exit</u>
Ctrl+F4	<u>close_window</u>
Ctrl+F5	<u>restore_window</u>
F6	<u>window_list</u>
Ctrl+F6	<u>next_window</u>
F7	<u>record_macro</u>
F8	<u>play_macro</u>
Ctrl+F10	<u>zoom_window</u>
F12	<u>wrap_para</u>

Other Keys

Alt+-	<u>close_window</u>
Alt+B	<u>window_list</u>
Alt+BkSp	<u>undo</u>
Alt+D	<u>delete_line</u>
Alt+I	<u>indent_block</u>
Alt+K	<u>delete_to_eol</u>
Alt+N	<u>next_window</u>
Alt+P	<u>print</u>
Alt+U	<u>unindent_block</u>
Alt+V	<u>about</u>
Alt+X	<u>save_all_exit</u>
BkSp	<u>backspace</u>
Ctrl+A	<u>find_again</u>
Ctrl+B	<u>to_bottom</u>
Ctrl+BkSp	<u>delete_word_left</u>
Ctrl+C	<u>to_center</u>
Ctrl+Del	<u>delete_word_right</u>
Ctrl+F	<u>find_files_containing</u>
Ctrl+G	<u>goto_line</u>
Ctrl+Ins	<u>copy</u>
Ctrl+L	<u>list_files_containing</u>
Ctrl+M	<u>match_delimiter</u>
Ctrl+N	<u>split_line</u>
Ctrl+R	<u>replace</u>
Ctrl+S	<u>find</u>
Ctrl+T	<u>to_top</u>
Ctrl+W	<u>toggle_wordwrap</u>
Del	<u>delete</u>
Enter	<u>enter</u>
Gray *	<u>undo</u>
Gray +	<u>copy</u>
Gray -	<u>cut</u>
Ins	<u>toggle_insert</u>
Shift+Del	<u>cut</u>
Shift+Ins	<u>paste</u>
Tab	<u>tab_right</u>

File Menu

<u>N</u> <u>e</u> <u>w</u>	Creates a new window for a new, untitled document.
<u>O</u> <u>p</u> <u>e</u> <u>n</u>	Opens an existing file for editing, or, if the file is already open in another window, switches to that window.
<u>C</u> <u>l</u> <u>o</u> <u>s</u> <u>e</u>	Closes the current window.
<u>S</u> <u>a</u> <u>v</u> <u>e</u>	Saves the contents of the current buffer to disk.
<u>S</u> <u>a</u> <u>v</u> <u>e</u> <u>A</u> <u>s</u>	Saves the contents of the current buffer with a new filename.
<u>S</u> <u>a</u> <u>v</u> <u>e</u> <u>A</u> <u>l</u> <u>l</u>	Saves the contents of all modified buffers to disk.
<u>I</u> <u>n</u> <u>s</u> <u>e</u> <u>r</u> <u>t</u>	Inserts the contents of a specified file in the current buffer.
<u>R</u> <u>e</u> <u>v</u> <u>e</u> <u>r</u> <u>t</u>	Rereads the file associated with the current buffer from disk.
<u>W</u> <u>r</u> <u>i</u> <u>t</u> <u>e</u> <u>B</u> <u>l</u> <u>o</u> <u>c</u> <u>k</u>	Writes the selected text to a specified disk file.
<u>P</u> <u>r</u> <u>i</u> <u>n</u> <u>t</u> <u>e</u> <u>r</u> <u>S</u> <u>e</u> <u>t</u> <u>u</u> <u>p</u>	Permits you to select a printer for the Print command and change options for the selected printer.
<u>P</u> <u>a</u> <u>g</u> <u>e</u> <u>S</u> <u>e</u> <u>t</u> <u>u</u> <u>p</u>	Permits you to specify the header, footer, margins and font for the Print command.
<u>P</u> <u>r</u> <u>i</u> <u>n</u> <u>t</u>	If a block is marked, prints the selected text; if no block is marked, prints the entire buffer.
<u>C</u> <u>o</u> <u>m</u> <u>p</u> <u>a</u> <u>r</u> <u>e</u>	Compares two files.
<u>R</u> <u>u</u> <u>n</u>	Executes a DOS command, program, or batch file.
<u>E</u> <u>x</u> <u>i</u> <u>t</u>	Prompts to save any modified buffers to disk and ends the editor session.

Edit Menu

<u>Undo</u>	Reverses the effects of a previous editing operation.
<u>Cut</u>	Deletes the marked block and moves it to the Windows clipboard.
<u>Copy</u>	Copies the marked block to the Windows clipboard.
<u>Paste</u>	Inserts text from the Windows clipboard at the current cursor position in the active window.
<u>Delete</u>	Deletes the marked block without moving it to the clipboard.
<u>Indent</u>	Indents the marked block one tab position.
<u>Unindent</u>	Unindents the marked block one tab position.

Goto Menu

Line

Matching Delimiter

Tag

Moves the cursor to the start of the specified line.

Moves the cursor to the matching brace, bracket, or parenthesis.

Moves the cursor to the start of a tagged function definition.

Search Menu

<u>Find</u>	Finds the next or previous occurrence of a specified text pattern or regular expression.
<u>Find Again</u>	Repeats the last Find operation, using the same search pattern, direction, and options.
<u>Replace</u>	Replaces one or more occurrences of a specified text string or regular expression.
<u>Find Files Containing...</u>	Searchs for all occurrences of a specified text pattern or regular expression in a group of files on disk and builds a list of files that contain the search pattern.
<u>List Files Containing...</u>	Lists the unopened files selected by the last Find Files Containing... operation.

Macro Menu

Record/Stop Recording
Play

Begins/ends a keyboard macro definition.
Plays back a keyboard macro.

Options Menu

<u>Current Buffer</u>	Displays the status of the buffer being edited in the current window, and permits you to modify the buffer's control settings.
<u>Editing Defaults</u>	Autoindent, tabs, word wrap margin.
<u>File Defaults</u>	Default keyboard configuration file, start-up directory, default extension, tag file, remove trailing spaces, backups, autosave, file locking, alternate help file.
<u>Preferences</u>	Display font, cursor style, typing replaces selection, cut/copy current line if no text is selected, restore session, undo levels.
<u>Key Assignments</u>	Assigns/unassigns editor functions to specific keys, loads/saves keyboard configuration files.

Window Menu

For more information, select one of the following command names:

New Window

Creates a new window on the buffer being edited in the current window.

Cascade

Arranges all open windows in an overlapping pattern so that each window's title bar is visible.

Tile

Arranges all open windows side by side.

List

Displays a list of open windows and permits you to close windows, save modified buffers, switch to a different window or open a new file.

Help Menu

For more information, select one of the following command names:

<u>Help</u>	Enters the editor's hypertext help system.
<u>SDK Help</u>	Accesses Microsoft Windows SDK Help.
<u>About</u>	Displays the Upper Deck Editor version number and copyright notice.

New Command

Function name: `new_file`

Creates a new window for a new, untitled document.

Open Command

Function name: open_file

Default key assignment: F3

Opens an existing file for editing, or, if the file is already open in another window, switches to that window. (To open another window on the buffer being edited in the current window, select New Window from the Window menu.)

The Open dialog box permits you to specify the name of the file you want to open.

To save typing, you can click on the arrow to the right of the **Filename** edit field to display a list of previous entries. Use the mouse or the arrow keys to choose an entry from the list. You may also press the Down arrow key to recall previous entries without displaying the list.

If you enter a filename with no extension in the **Filename** edit field, the default extension (if any) will be appended to the filename you enter. When the editor is started, the default extension for the Open dialog is the default extension specified in the File Defaults dialog box on the Options Menu. If you enter a different wild card file specification in the Filename edit field, and if the extension part of that file specification does not contain any wild card characters, the new extension will be used as the default extension for the rest of the session or until changed again. If you subsequently enter a filename with no extension, the new default extension will be appended. If you really want to specify a filename with no extension, type a period (.) at the end of the filename.

The current drive and directory are displayed in the **Directory** display field. The **Files** list box displays a list of all the files with the default extension in the current directory. The **Directories** list box displays a list of all attached drives and all subdirectories of the current directory. (The symbol .. indicates the parent of the current directory.) You may change the default drive and/or directory by double clicking on any of the items in the **Directories** list box, by entering the name of the new drive or directory in the **Filename** entry field, or by specifying a filename that contains one or more wild card characters.

If you click once on any of the filenames displayed in the **Files** listbox, the size of the file in bytes and the date and time of the last update to the file will be displayed under the **Files** listbox. Double clicking on the filename will close the dialog box and open the selected file for editing.

If the **Read only** option is selected, modifications to the buffer will not be permitted.

Close Command

Function name: `close_window`

Default key assignments: `Ctrl+F4`, `Alt+-`

Closes the current window. If the buffer in the current window is modified and is not being edited in another window, you will be prompted to save the contents of the buffer before the window is closed.

Save Command

Function name: `save_file`

Default key assignment: **F2**

Saves the contents of the current buffer to disk. If the buffer is untitled, you will be prompted to specify a filename. If the buffer has not been modified since it was last saved, it will not be saved (the message "Buffer not modified" will be displayed on the status line).

Save As Command

Saves the contents of the current buffer with a new filename.

Save All Command

Function name: `save_all`

Saves the contents of all modified buffers to disk.

Insert File Command

Opens a dialog box that prompts for a filename and Inserts the contents of the specified file at the current cursor location in the buffer displayed in the active window.

The Insert File dialog box permits you to specify the name of the file you want to insert. The current drive and directory are displayed in the **Directory** display field. The **Files** list box displays a list of all the files with the default extension in the current directory. The **Directories** list box displays a list of all attached drives and all subdirectories of the current directory. (The symbol .. indicates the parent of the current directory.)

You may change the default drive and/or directory by double clicking on any of the items in the **Directories** list box, by entering the name of the new drive or directory in the **Filename** entry field, or by specifying a filename that contains one or more wild card characters.

If you click once on any of the filenames displayed in the **Files** listbox, the size of the file in bytes and the date and time of the last update to the file will be displayed to the right of the **Filename** entry field. Double clicking on the filename will close the dialog box and insert the contents of the selected file at the current cursor location.

Revert Command

Function name: `revert`

If the current buffer is modified, prompts for confirmation and then rereads the file from disk. Any modifications made since the file was last saved are lost. The cursor is repositioned at the beginning of the buffer.

Write Block Command

Opens a dialog box that prompts for a filename and writes the selected text to the specified file. This command is disabled if no text is selected.

Printer Setup Command

Permits you to select a printer for the Print command and change options for the selected printer. The options available depend on the type of printer selected.

Page Setup Command

Permits you to specify the header, footer, margins and font for the Print command. The following special sequences may be used to insert the file name, the current date and time, or the page number into the header or footer:

- %f File name
- %d Current date and time
- %p Page number

The margins may be specified either in inches or in centimeters, depending on the value of **iMeasure** in the **[intl]** section of **win.ini**.

The Printer Font dialog permits you to select the typeface and size in points of the printer font.

The settings specified in the Page Setup and Printer Font dialogs are stored in **session.ini** and will be used for subsequent editor sessions until they are specifically changed.

Print Command

Function name: `print`

If a block is marked, prints the selected text; if no block is marked, prints the entire buffer.

Compare Command

Function name: compare

Compares two files, starting at the specified line in each file. If either file is already in an edit buffer, the current contents of the buffer (which may not be identical to the contents of the file on disk) are used in the comparison. You may select either a horizontal or vertical split screen display to view the results of the comparison.

The comparison is carried out on a line-by-line basis. If a difference is found, the first line that is not identical in both files is highlighted. A dialog box is displayed with a status message and a pushbutton marked **Next Match**. If you click on **Next Match**, the editor will attempt to resynchronize the comparison.

If resynchronization is successful, the matching lines will be highlighted in both files. The dialog box will display a status message and a pushbutton marked **Next Difference**. If you click on **Next Difference**, the editor will look for the next line that is not identical in both files.

This procedure may be repeated until there are no more differences or until resynchronization fails.

For resynchronization to succeed, a two-line match is required. The algorithm used will find the first possible match in File 1. If no two-line match is found within 50 lines, the message "No match found" is displayed and the comparison is ended.

Run Command

Function name: run

Executes a DOS command, program or batch file. If the **Redirect output** option is selected, the DOS redirection function will be used to redirect the program's output to the file specified in the **Output file** entry field.

If the **Save modified buffers first** option is selected, all modified buffers will be saved to disk before the DOS command, program or batch file is executed. If you use the **Run** command to run a DOS-based compiler, selecting the **Save modified buffers first** option will make sure you don't compile the old version of a file you have edited but not yet saved.

To save typing, you can click on the arrow to the right of the **Command** edit field to display a list of previously entered commands. Use the mouse or the arrow keys to choose a command from the list. You may also press the Down arrow key to recall previous entries without displaying the list.

If you specify redirection to a file that already exists, a warning will be displayed and you will be asked to confirm replacement of the file.

If output is redirected, after the command is completed you will be asked if you want to view the output file. If a prior version of the same file is already loaded in an edit buffer, you will be prompted to replace the contents of the existing buffer with the new version of the file.

You should bear in mind that some DOS programs, such as **chkdsk**, should not be used when Windows is running.

Internally the editor uses the command you enter to construct a batch file called **editrun.bat** in the directory that contains **ude.exe** and then invokes **command.com** to execute the batch file. The batch file begins by changing to the drive and directory shown in the **Current directory is** field in the Run command dialog box and then executes the specified command. If output is not redirected, a **pause** command is appended to the batch file to give you a chance to read the screen before the output is lost. The batch file is not automatically deleted. If the Run command is not working as you expect, you may be able to learn why by examining the contents of **editrun.bat**.

Exit Command

Function name: exit

Default key assignment: Alt+F4

Prompts to save any modified buffers to disk and ends the editor session.

Related topics:

[exit_windows](#)

[save_all_exit](#)

[save_all_exit_windows](#)

Undo Command

Function name: undo

Default key assignments: Alt+BkSp, Gray *

Reverses the effects of a previous editing operation. Each Undo command reverses the effects of one editor command or keystroke. The number of operations that can be undone may be specified with the Undo Levels setting on the Options Menu (the default is 100; the maximum is 300).

Any operation that moves the cursor or changes the text in the current buffer may be undone. File and window operations and changes to editor options may not be undone. When a buffer is saved to disk, the undo information for that buffer is lost., so only changes made since a buffer was last saved may be undone.

The Undo command does not affect the contents of the Clipboard.

A single undo buffer is maintained for each filebeing edited, even if the file is being edited in more than one window. The Undo command reverses the effects of the last editing operation performed on the file, even if that operation was not performed in the current window.

Cut Command

Function name: cut

Default key assignments: Shift+Del, Gray -

Deletes the selected text from the buffer displayed in the current edit window and moves it to the Clipboard.

The selected text must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

If the **Cut/copy current line if no text is selected** checkbox in the Preferences dialog box is checked, the current line will be used for this command if no text is selected.

Copy Command

Function name: copy

Default key assignments: Ctrl+Ins, Gray +

Copies the selected text from the current edit window to the Clipboard.

The selected text must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

If the **Cut/copy current line if no text is selected** checkbox in the Preferences dialog box is checked, the current line will be used for this command if no text is selected.

Paste Command

Function name: paste

Default key assignment: Shift+Ins

Inserts the contents of the Clipboard at the current cursor location in the buffer displayed in the current edit window.

The block of text to be inserted must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

Delete Command

Function name: delete

Default key assignment: Del

Deletes the selected text from the buffer displayed in the current edit window.

The selected text must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

Indent Command

Function name: indent_block

Default key assignment: Alt+I

Inserts a single tab character at the start of each non-blank line in the marked block. If the **Expand tabs** options is selected in the Edit Options dialog box, spaces will be inserted instead of a tab character. The number of spaces inserted is determined by the **Tab spacing** specified for the current buffer. The **Tab spacing** setting may be changed from the Current Buffer dialog box on the Options menu..

Unindent Command

Function name: unindent

Default key assignment: Alt+U

Deletes one tab character at the start of each line in the marked block if the first character in the line is a tab character.

Goto Line Command

Function name: goto_line

Default key assignment: Ctrl+G

Opens a dialog box that prompts for a line number and moves the cursor to the beginning of the specified line.

Goto Matching Delimiter Command

Function name: `match_delimiter`

Default key assignment: **Ctrl+M**

If the character at the insertion point is a brace, bracket, or parenthesis, the cursor is moved to the delimiter that matches it. If there is no matching delimiter, the cursor is not moved and the message "No match found" is displayed on the status line.

Goto Tag Command

Function name: goto_tag

Opens a dialog box that prompts for a tag and looks it up in the specified tag file. If the tag is found, opens the source file containing the definition of the tagged function and places the cursor at the start of the definition.

To use this function, you must supply a standard format tag file. The following formats are supported:

TagName c:\dir\filename.ext ^void TagName(void)

TagName c:\dir\filename.ext ?^void TagName(void)\$?

(The drive and directory may be omitted.)

The name of a default tag file may be specified in the [File Defaults](#) dialog box on the [Options Menu](#).

Related topic:

[ctag](#)

Find Command

Function name: find

Default key assignment: Ctrl+S

Finds the next or previous occurrence of a text pattern or regular expression in the current buffer. The search begins at the current cursor location and terminates when the end of the buffer is reached (or when the beginning of the buffer is reached if **Previous** is specified). The search is conducted on a line-by-line basis; patterns that span line boundaries will not be found.

When this command is invoked, if there is text selected in the current window (and if the selected text is short enough to be a plausible search pattern), the selected text will be copied to the **Pattern** edit field as a default. To save typing, you can click on the arrow to the right of the **Pattern** edit field to display a list of previously entered search patterns. Use the mouse or the arrow keys to choose a pattern from the list. You may also press the Down arrow key to recall previous entries without displaying the list.

The search pattern may contain embedded spaces.

If the **Ignore case** option is selected, upper case characters will be not differentiated from their lower case equivalents.

If the **Regular expression** option is selected, the characters %, \$, <, >, ?, *, @, [,], ~, - and \ take on special meaning.. See [Regular Expressions](#).

Case-sensitive searches are slightly faster than searches that ignore case, and literal searches (case-sensitive or not) are much faster than regular expression searches.

Related topics:

[Find Again](#)

[Replace](#)

[Find Files Containing...](#)

[Regular Expressions](#)

Find Again Command

Function name: find_again

Default key assignment: Ctrl+A

Repeats the last Find operation, using the same search pattern, direction, and options.

Related topic:

[Find](#)

Replace Command

Function name: replace

Default key assignment: Ctrl+R

Replaces the next occurrence of the specified text pattern or regular expression in the current buffer with the specified replacement string. The search begins at the current cursor location and terminates when the end of the buffer is reached. The search is conducted on a line-by-line basis; patterns that span line boundaries will not be replaced.

When this command is invoked, if there is text selected in the current window (and if the selected text is short enough to be a plausible search pattern), the selected text will be copied to the **Pattern** edit field as a default. To save typing, you can click on the arrows to the right of the **Pattern** and **Replacement** edit fields to display a list of previously entered strings. Use the mouse or the arrow keys to choose a string from the list. You may also press the Down arrow key to recall previous entries without displaying the list.

If the **Ignore case** option is selected, upper case characters will be not differentiated from their lower case equivalents when searching for the pattern. The replacement string is always substituted exactly as entered.

If the **Regular expression** option is selected, the characters %, \$, <, >, ?, *, @, [,], ~, - and \ take on special meaning when used in the search pattern. In addition, \t may be used in the replacement string to stand for a tab character. See [Regular Expressions](#).

If the **Confirm changes** option is selected, a dialog box will be displayed at the top of the screen to confirm each replacement. (This dialog box may be moved, if necessary, to get it out of the way.) If you uncheck the **Confirm** checkbox in the confirmation dialog box and then click on the **Yes** pushbutton, all remaining occurrences of the search pattern will be replaced. (If the **Confirm** checkbox is unchecked, clicking on **No** is equivalent to clicking on **Cancel**.)

Related topics:

[Find](#)

[Find Files Containing...](#)

[Regular Expressions](#)

Find Files Containing... Command

Function name: find_files_containing

Default key assignment: Ctrl+F

Searches all files matching the specification in the **Files** edit field for occurrences of the specified text pattern or regular expression and builds a list of the files that contain the pattern. The search is conducted on a line-by-line basis; patterns that span line boundaries will not be found. When the search is completed, the List Files Containing... dialog box is displayed.

When this command is invoked, if there is text selected in the current window (and if the selected text is short enough to be a plausible search pattern), the selected text will be copied to the **Pattern** edit field as a default. To save typing, you can click on the arrow to the right of the **Pattern** edit field to display a list of previously entered search patterns. Use the mouse or the arrow keys to choose a pattern from the list. You may also press the Down arrow key to recall previous entries without displaying the list.

The search patterns may contain embedded spaces.

Normally the entry in the **Files** edit field will contain one or more wild card characters. More than one file specification may be entered. If the drive and directory are not specified, the current drive and directory will be used. For example:

```
*.c *.dlg *.rc d:\include\*.h e:\docs\*.rtf
```

If the **Ignore case** option is selected, upper case characters will be not differentiated from their lower case equivalents.

If the **Regular expression** option is selected, the characters %, \$, <, >, ?, *, @, [,], ~, - and \ take on special meaning. See Regular Expressions.

Case-sensitive searches are slightly faster than searches that ignore case, and literal searches (case-sensitive or not) are much faster than regular expression searches.

Related topics:

[Find](#)

[Find Again](#)

[Replace](#)

[List Files Containing...](#)

[Regular Expressions](#)

List Files Containing... Command

Function name: list_files_containing

Default key assignment: Ctrl+L

Displays a list of the files selected by the most recent Find Files Containing... command. Only those files that have not yet been opened from this dialog box are listed. You may use the mouse or the cursor keys to choose a file from the list. The selected file is opened (just as if you had used the Open command on the File menu) and the cursor is moved to the first occurrence of the specified pattern. The Find Again command may be used to find subsequent occurrences (the options specified with the Find Files Containing... command will be used).

When a file is opened from this dialog box, its name is removed from the list. You can use this command repeatedly to work your way through the list of files containing the specified pattern.

Record/Stop Recording Macro Command

Function name: record_macro

Default key assignment: F7

Begins or ends a keyboard macro definition. The definition starts when this command is executed for the first time. All commands executed with the keyboard or the mouse are recorded until this command is executed again. **REC** is displayed on the status line while a macro definition is in progress. After you have defined a keyboard macro, you can use the Play Macro command to play it back.

A trivial example (using the default key assignments):

Press **F7**. (**REC** should be displayed on the status line.)

Type "this is a test".

Press **F7** again. (**REC** should disappear.)

The macro has been defined. Now, each time you press **F8**, the words "this is a test" will be inserted in the current edit window.

In Version 1.00 there are three limitations on keystroke macros:

Only one keyboard macro may be stored at a time.

Keyboard macros are limited to 256 recorded events. In general each keystroke or menu selection generates one recorded event; most messages that do not result in command execution or cursor movement are not recorded. If the limit of 256 recorded events is reached, the macro definition is terminated automatically.

Keyboard macros do not work inside dialog boxes.

Play Macro

Function name: play_macro

Default key assignment: F8

Plays back a keyboard macro defined with the Record/Stop Recording Macro command.

Current Buffer

Function name: `current_buffer`

Default key assignment: F4

Displays the status of the buffer being edited in the current window, and permits you to modify the buffer's control settings.

If the **Expand tabs with spaces** option is selected, when the **Tab** key is pressed a sufficient number of spaces will be inserted in the buffer to advance the cursor to the next tab position. If the **Expand tabs** option is not selected, a single tab character will be inserted.

The **Tab spacing** setting controls the fixed tab spacing for the buffer.

If the **Word wrap** option is selected, a carriage return and line feed will be inserted before the start of the last word on a line when the length of the line (allowing for expansion of tab characters) exceeds the value specified in the **Right margin** entry field. (For the purposes of the word wrap function, a word is defined as any string of characters delimited by a space or tab character.) If the **Word wrap** option is not selected, the **Right margin** setting has no effect.

If the **Autoindent** option is selected, pressing **Enter** will position the cursor under the first non-blank character on the line above; if not, pressing **Enter** will position the cursor in column 1.

The initial settings for **Expand tabs**, **Tab spacing**, **Autoindent** and **Right margin** are taken from the Edit Defaults in effect when the buffer is created.

If the **Read only** option is selected, further modifications to the buffer are not permitted. However, if a buffer is modified before the **Read only** option is selected, the modifications remain in effect and the modified buffer may be saved to disk at any time. The **Read only** option may also be selected in the Open command dialog box.

Editing Defaults

The settings specified in this dialog box are used as defaults when a new buffer is opened. After the buffer is opened, the settings for the buffer may be modified from the Current Buffer dialog box on the Options menu.

If the **Expand tabs with spaces** option is selected, when the **Tab** key is pressed a sufficient number of spaces will be inserted in the buffer to advance the cursor to the next tab position. If the **Expand tabs** option is not selected, a single tab character will be inserted in the buffer.

The **Tab size** setting controls the fixed tab spacing for the buffer.

The **Right margin** setting is used when word wrap is enabled.

If the **Autoindent** option is selected, pressing **Enter** will position the cursor under the first non-blank character on the line above; if not, pressing **Enter** will position the cursor in column 1.

File Defaults

The **Keyboard configuration file** specified in this dialog box will be loaded when the editor is first started. If no file is specified, or if the specified file does not exist, the editor's internal defaults will be used. After the editor is started, a different keyboard configuration file may be loaded at any time with the Key Assignments command on the Options Menu. The complete pathname of the file, including drive and directory, should be entered in this edit field.

If most of the files you will be editing are in one directory, you may specify that directory in the **Start-up directory** field. When the editor is first started, it will change to the specified directory.

If most of the files you will be editing have the same extension, you may specify that extension in the **Default extension** field. If you enter a filename with no extension in the **Filename** field of the Open command dialog box, the default extension will automatically be appended. If you really want to specify a filename with no extension, type a period (.) at the end of the filename.

The file specified in the **Tag file** field will be used as the default tag file for the ctag and goto_tag commands.

If the **Remove trailing spaces** option is selected, spaces and tab characters after the last non-blank character on a line will be truncated when a file is loaded into the editor.

If the **Make backup files** option is selected, a backup file will be made whenever a file is saved by renaming the original file with the extension **.bak**.

The **Autosave** parameters specify the autosave interval in minutes and in changes to the buffer. If both parameters are zero (the default), autosave will not be performed. See Autosave for details.

If the **File locking** option is selected, other applications (including other instances of the editor running on a network) will not be permitted to access files opened by the editor. When this option is selected, the number of files that may be open at the same time is limited to 35. For file locking to work, **share.exe** must be installed before starting Windows.

The **Alternate help file** specified in this dialog box is used with the alt_help function. This function permits you to access the specified Windows help file from within the editor.

Preferences

The display font and cursor style may be selected in this dialog box.

If the **Typing replaces selection** box is checked, selected text will be replaced by any typed character. This is customary in Windows applications, but can be startling if you are not used to it.

If the **Cut/copy current line if no text is selected** box is checked, the cut and copy commands will act on the current line if no text is selected.

If the **Restore session** box is checked, the files you were working on at the end of the last editor session will be loaded automatically when you start the editor.

The **Undo Levels** setting controls the number of operations stored in the undo buffer for each file. The default setting is 100; the maximum is 300. Setting **Undo Levels** to zero disables the Undo function.

Key Assignments Command

Opens a dialog box that permits you to assign editor functions to keys, remove existing key assignments, enable or disable menu accelerators, and load or save a keyboard configuration file.

To assign a function to a key, select the desired function in the **Function** list box.. A list of the keys currently assigned to the highlighted function will be displayed in the box labeled **Current Keys**. Then select the desired key in the **Key** list box. The name of the function currently assigned to the highlighted key will be displayed in the box labeled **Current Function**. Click on the **Assign** pushbutton to assign the highlighted function to the highlighted key.

To remove an existing key assignment, select the desired key in the **Key** list box. The name of the function currently assigned to the highlighted key (if any) will be displayed in the box labeled **Current Function**. Click on the **Unassign** pushbutton to remove the key assignment.

If you prefer, you may type the name of the key or function in the edit field above the corresponding list box. If you do, however, be sure to spell the name exactly as it appears in the list box. The list of **Current Keys** will not be correct unless the specified function is actually highlighted in the **Function** list box; similarly the **Current Function** field will not be correct unless the specified key is actually highlighted in the **Key** list box.

If the **Enable menu accelerators** option is selected (the default), pressing **Alt+F** will activate the File menu, pressing **Alt+E** will activate the Edit menu, etc. That means that you cannot assign specific editor functions to **Alt+F**, **Alt+E**, **Alt+G**, **Alt+S**, **Alt+M**, **Alt+O**, **Alt+W** and **Alt+H**. If you want to assign specific editor functions to these key combinations, deselect the **Enable menu accelerators** option. In that case you can still access the menus with the mouse or by pressing the **Alt** key by itself to activate the menu bar.

Certain key combinations (**Alt+F4**, **Alt+Tab**, **Ctrl+Esc**) are used by Windows and may not be reassigned. WordStar-style key sequences are not supported, except for the default action of the menu accelerator keys.

To save the current keyboard configuration, enter the name of the file in which the configuration is to be saved in the **Name of keyboard configuration file** edit field and press the **Save** pushbutton. The setting of the **Enable menu accelerators** option is saved in the keyboard configuration file.

At startup, the editor loads the default keyboard configuration file, specified in [File Defaults](#). If no configuration file is specified, or if the specified file does not exist, the editor's default key assignments are used.

To load a previously saved keyboard configuration file during an editing session, enter the name of the file in the **Name of keyboard configuration file** edit field and click on the **Load** pushbutton.

Clicking on the **OK** pushbutton will accept the changes you have made and return you to the current edit window. Clicking on the **Cancel** pushbutton will restore the key assignments and the setting of the **Enable menu accelerators** checkbox that were in effect before the dialog was invoked. Unless specifically saved, changes to the key assignments will remain in effect only for the duration of the current editing session.

New Window Command

Creates a new window on the buffer being edited in the current window. The new window becomes the current window, and the cursor is placed at the beginning of the buffer.

Cascade

The Cascade command restores the current edit window to its normal size (if it was maximized) and arranges all the edit windows so that they appear like cards stacked one behind the other, with the current edit window on top.

Related topic: [Tile](#)

Tile

The Tile command restores the current edit window to its normal size (if it was maximized) and arranges all the edit windows side by side, like floor tiles. The current edit window is placed at the top left.

Related topic: [Cascade](#)

Window List Command

Default key assignment: Right mouse button

Displays a list of open windows. If the buffer in a window has been modified since it was last saved to disk, the word **modified** appears after the filename in the list box.

Selecting the **Switch To** pushbutton or double-clicking the left mouse button on one of the windows in the list will close the Window List dialog box and switch to the selected window.

Selecting the **Close** pushbutton will close the window highlighted in the list box.

Selecting the **Cancel** pushbutton will close the Window List dialog box.

Selecting the **Open...** pushbutton will close the Window List dialog box and invoke the **Open** command, permitting you to open a file not found on the list.

Selecting the **Save** pushbutton will save the buffer in the window highlighted in the list box.

Selecting the **Save All** pushbutton will save all modified buffers.

Editor Help

Default key assignment: F1

Enters the editor's hypertext help system.

SDK Help

Default key assignment: Alt+F1

Accesses the Microsoft Windows Software Development Kit help system. For this command to work, the SDK help file (**sdkwin.hlp** or **win31wh.hlp**) must be available in a directory listed in the PATH environment variable.

About

Displays the Upper Deck Editor version number and copyright notice and information about the mode in which Windows is running, the amount of free memory and the percentage of free system resources.

License Statement

This software is copyrighted and protected by United States copyright law and international treaty provisions. You may not rent, lease, sell, sublicense, assign or otherwise transfer this software or the accompanying documentation. You may, however, make archival copies of the software to protect against loss or damage.

The software and documentation are provided "AS IS" WITHOUT WARRANTY OF ANY KIND. UPPER DECK SYSTEMS SPECIFICALLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THIS SOFTWARE AND DOCUMENTATION REMAINS WITH YOU. IN NO EVENT SHALL UPPER DECK SYSTEMS BE LIABLE FOR ANY LOSS OF PROFIT, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER COMMERCIAL DAMAGE ARISING OUT OF THE USE OF THIS SOFTWARE OR DOCUMENTATION, INCLUDING BUT NOT LIMITED TO COMPENSATORY, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY OR OTHER DAMAGES.

Microsoft is a registered trademark of Microsoft Corporation. Windows is a trademark of Microsoft Corporation. BRIEF is a registered trademark of UnderWare, Inc. WordStar is a trademark of WordStar International. Other brand and product names are trademarks or registered trademarks of their respective holders.

Features

The Upper Deck Editor is a programmer's editor designed for the Microsoft Windows environment. It provides the following features:

The editor uses all memory available to Windows (including virtual memory in 386 Enhanced Mode) to edit large files and to keep a large number of buffers and windows open at once. File size and the number of buffers and windows that may be open at the same time are limited only by available memory. Clicking the right mouse button anywhere in the edit window displays a list of the open windows, permitting you to move around easily when many windows are open.

Full 300-level undo capability permits you to reverse the effects of any editing operation.

Buffers may be marked "Read only" to guard against accidental modification, and a Revert command is provided. A flexible autosave function is provided to help minimize loss of data when a system crash or power failure occurs.

The editor uses the Windows Clipboard to cut, copy, and paste text within and between files. Text may also be transferred to and from other applications.

Functions to find and replace text patterns and regular expressions are provided. In addition, functions are provided to find and list all files that contain a specified text pattern or regular expression. History lists are maintained to facilitate entry of search and replacement patterns.

A file comparison function is provided, with either horizontal or vertical split screen display and automatic resynchronization when differences are detected.

DOS commands, programs and batch files may be run from within the editor, with or without redirection of output to a file.

Functions are provided to parse compiler error messages and jump to the line in the source file that caused the error, with the error message displayed on the status line. Currently the Microsoft C and Borland C++ and Turbo Pascal for Windows compilers are supported.

A function is provided to match parentheses, brackets and braces

Keyboard macros may be defined to automate repetitive editing tasks.

An auto-indent mode and functions to indent and unindent marked blocks of text are provided to help with the formatting of program source files. A word wrap mode with adjustable right margin is provided for simple word processing tasks.

A function is provided to print selected text or an entire buffer. A Page Setup dialog box permits you to specify a header and footer as well as the margins and font to be used for the Print command.

Functions are provided to write a marked block to disk and to insert a file at the current cursor location.

The editor's key assignments may be changed on the fly, and the current keyboard configuration may be saved in a file. Multiple keyboard configuration files are supported. You can specify a default keyboard configuration file to be loaded automatically when the editor is started, and you can load a new keyboard configuration file at any time during an editing session. If desired, menu accelerators may be disabled so that the keys normally used to pull down the menus may be assigned other functions.

The normal Windows convention that typing replaces selected text may be disabled from the Preferences dialog box. The cut and copy commands may be configured to act on the current line if no text is selected. The screen font and cursor style are user-selectable.

If the "Restore session" option is selected, the files that you were working on at the end of the last editor session will be loaded automatically when you start the editor.

Context-sensitive on-line help is provided. In addition, you can access Microsoft Windows SDK help directly from the editor's help menu if the appropriate file is available in a directory listed in the PATH environment variable.

Limitations and Known Bugs

The Upper Deck Editor is a line-oriented editor designed to be used to edit text files. Like the Windows Clipboard, the editor expects lines in text files to be terminated with a carriage return and line feed. If this is not the case, the file may not be read or saved correctly.

When a file is edited and saved, each line in the saved file will always end with a carriage return and line feed, even if it did not before and was not specifically modified.

The official limit on line length is 1024 characters, but there is no real reason why much longer lines (up to 32K) should cause any problem.

Cut, copy and paste operations and block deletions are restricted to blocks smaller than 64K bytes in size.

Files may not exceed 4 billion lines, but the editor imposes no limit on file size in bytes, on the number of files that may be edited at once, or on the number of windows that may be open at the same time. Even virtual memory is finite, however, and if you attempt to do too much you will eventually exhaust the available resources. When that happens the editor will display a message box indicating that there is insufficient memory to complete the requested operation. You should respond to this by closing one or more files; if a file is displayed in more than one window you should close all of the windows associated with it. Sometimes simply saving a file that has been heavily modified will free a substantial amount of memory, since the undo information associated with the file will be discarded.

Regular Expressions

The Find, Replace, and Find Files Containing... commands support the use of regular expressions (wild card characters) in the search pattern. The following characters are supported:

<code>?</code>	Matches any character.
<code>*</code>	Matches zero or more occurrences of any character.
<code>@</code>	Matches zero or more occurrences of the previous character or expression.
<code>%</code> or <code><</code>	Matches the beginning of a line.
<code>\$</code> or <code>></code>	Matches the end of a line.
<code>[. . .]</code>	Matches any of the characters listed between <code>[</code> and <code>]</code> . A hyphen (<code>-</code>) may be used to specify a range of characters. For example, <code>[abc]</code> matches <code>a</code> , <code>b</code> , or <code>c</code> ; <code>[a-z]</code> matches any lower case letter; <code>[A-Za-z]</code> matches any upper or lower case letter.
<code>[~ . . .]</code>	Matches any character except those listed between <code>[~</code> and <code>]</code> . A hyphen (<code>-</code>) may be used to specify a range of characters. For example, <code>[~A]</code> matches any character but <code>A</code> ; <code>[~abc]</code> matches any character except <code>a</code> , <code>b</code> , or <code>c</code> ; <code>[~A-Za-z]</code> matches any non-alphabetic character.
<code>\</code>	Escape character indicating that the following character should be taken literally rather than used as a wild card character. For example, <code>*</code> may be used to search for an asterisk; <code>\\</code> may be used to search for a backslash character.
<code>\t</code>	Matches a tab character.
<code>\f</code>	Matches a formfeed character.

Some examples:

<code>%\$</code>	Finds a blank line.
<code><></code>	Finds a blank line.
<code><[\t][\t]@></code>	Finds a line with nothing on it but white space (blanks and/or tabs).
<code>delete_@line</code>	Finds <code>"deleteline"</code> , <code>"delete_line"</code> , <code>"delete__line"</code> , etc.
<code>the[mn]</code>	Finds <code>"them"</code> or <code>"then"</code> .
<code><{</code>	Finds <code>"{"</code> at the start of a line.

Delimiters

A word is defined as a string of characters delimited by a space or tab character or one of the following:

`_ () ' " , # $ % ! % ^ & * { } [] \ ? / | < > ; . + - = ~ :`

This definition is used in the following functions:

delete_word_left
delete_word_right
select_word
select_word_left
select_word_right
word_left
word_right

Autosave

If the autosave function is enabled, each modified buffer is saved at regular intervals in a temporary file in the editor's home directory. Under normal circumstances these temporary files are automatically deleted when the editor exits. If a system crash or power failure occurs, however, the editor will not exit normally and these temporary files will not be deleted. This permits you to recover work that would otherwise be lost.

The temporary files created by the autosave function have names that begin with the character ~, followed by a unique number, with the extension **.sav**. For example:

~4289352.sav

For identification purposes, the autosave function adds a line in the following format at the beginning of each temporary file:

/* AUTOSAVE C:\DIR\FILENAME.EXT 08-12-91 7:35pm */

The added line contains the word **AUTOSAVE** followed by the complete pathname of the corresponding file and the date and time of the autosave, formatted as a C language comment. The rest of the temporary file, starting with line 2, stores the contents of the buffer at the time the autosave was performed.

To recover from a system crash or power failure, examine each file with the extension **.sav** in the editor's home directory. You can load all of these files into the editor at once using the Program Manager's Run command. For example, if the editor's home directory is **c:\ude**:

c:\ude\ude c:\ude*.sav

You should examine each **.sav** file carefully. Normally some changes will be lost, since autosave is not performed after each change. If you decide to keep the **.sav** file, you can use the Save As command on the File Menu to save it under its original name. Don't forget to delete the line added by the autosave function at the beginning of the file.

The autosave interval may be specified in the File Defaults dialog box on the Options Menu. The interval may be specified in minutes or in changes to the buffer since autosave was last performed, or both. If both autosave parameters are zero (the default), autosave will not be performed.

File Locking

If the **File locking** option is selected in the File Defaults dialog box on the Options menu, other applications (including other instances of the editor running on a network) will not be permitted to access files opened by the editor. When this option is selected, the number of files that may be open at the same time is limited to 35. For file locking to work, **share.exe** must be installed before starting Windows.

about

Displays the Upper Deck Editor version number and copyright notice and information about the mode in which Windows is running, the amount of free memory and the percentage of free system resources.

alt_help

Accesses a user-specified alternate help file. The alternate help file is specified in the [File Defaults](#) dialog box on the Options menu.

Related topics:

[editor_help](#)

[sdk_function_help](#)

[sdk_help](#)

backspace

Default key assignment: BkSp

If text is selected, deletes the selected text; otherwise deletes the character to the left of the cursor. If the cursor is at the beginning of the line, the line containing the cursor is appended to the previous line.

bc_error

With the cursor positioned on any line of a Borland C++ compiler error output file, jumps to the line in the source file that caused the error, with the error message displayed on the status line. For this function to work properly, the current line in the error output file must contain an actual error or warning message. The cursor does not need to be at the beginning of the line.

The function show_message may be used at any time to display the full text of the current error or warning message without switching back to the error output file.

After you have used this function to establish a starting point in the error output file, you may use bc_next_error and bc_prev_error to move directly to the next or previous error.

Related topic:

msc_error

bc_next_error

Finds the next error message in a Borland C++ compiler error output file and jumps to the line in the source file that caused the error, with the error message displayed on the status line. For this function to work properly, you must first use [bc_error](#) to establish a starting point in the error output file.

The function [show_message](#) may be used at any time to display the full text of the current error or warning message without switching back to the error output file.

Related topics:

[bc_prev_error](#)

[msc_error](#)

bc_prev_error

Finds the previous error message in a Borland C++ compiler error output file and jumps to the line in the source file that caused the error, with the error message displayed on the status line. For this function to work properly, you must first use [bc_error](#) to establish a starting point in the error output file.

The function [show_message](#) may be used at any time to display the full text of the current error or warning message without switching back to the error output file.

Related topic:

[bc_next_error](#)

[msc_error](#)

beginning_of_buffer

Default key assignment: Ctrl+Home

Moves the cursor to the beginning of the buffer.

beginning_of_line

Default key assignment: Home

Moves the cursor to the beginning of the current line.

bottom_of_window

Default key assignment: Ctrl+PgDn

Moves the cursor to the bottom line of the window. If possible, the cursor column remains the same.

change_case

Converts all lower case characters in the marked block to upper case and all upper case characters to lower case. All other characters are left unchanged.

If no block is marked, this function acts on the character at the current cursor position.

Related topics:

[lowercase](#)

[uppercase](#)

close_window

Default key assignments: Ctrl+F4, Alt+-

Closes the current window. If the buffer in the current window is modified and is not being edited in any other window, you will be prompted to save the contents of the buffer before the window is closed. Identical to the Close command on the File menu.

Related topic:

[save_file_close_window](#)

compare

Identical to the Compare command on the File Menu.

copy

Default key assignments: Ctrl+Ins, Gray +

Copies the selected text from the current edit window to the Clipboard.

The selected text must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

If the **Cut/copy current line if no text is selected** checkbox in the Preferences dialog box is checked, the current line will be used for this command if no text is selected.

Identical to the Copy command on the Edit Menu.

Related topics:

[copy_block](#)

[copy_line](#)

copy_block

Copies the selected text from the current edit window to the Clipboard.

The selected text must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

Related topics:

[copy](#)

[copy_line](#)

copy_line

Copies the line containing the cursor from the current edit window to the Clipboard.

Related topics:

[copy](#)

[copy_block](#)

ctag

Parses the function name at the current cursor location in the active window and looks it up in the current tag file. If a matching tag is found, opens the source file containing the definition of the tagged function and places the cursor at the start of the definition.

To use this function, you must supply a standard format tag file. The following formats are supported:

TagName c:\dir\filename.ext ^void TagName(void)

TagName c:\dir\filename.ext ?^void TagName(void)\$?

(The drive and directory may be omitted.)

The name of a default tag file may be specified in the File Defaults dialog box on the Options Menu. To change to a different tag file during an editing session, use the goto_tag function.

current_buffer

Default key assignment: F4

Identical to the Current Buffer command on the Options Menu.

cursor_down

Default key assignment: Down arrow

Moves the cursor down one line. If possible, the cursor column remains the same. Does nothing if the cursor is on the last line of the buffer.

cursor_left

Default key assignment: Left arrow

Moves the cursor one character to the left. If the cursor is at the beginning of a line, it is moved to the end of the previous line. Does nothing if the cursor is at the beginning of the buffer.

cursor_right

Default key assignment: Right arrow

Moves the cursor one character to the right. If the cursor is at the end of a line, it is moved to the beginning of the following line. Does nothing if the cursor is at the end of the buffer.

cursor_up

Default key assignment: Up arrow

Moves the cursor up one line. If possible, the cursor column remains the same. Does nothing if the cursor is on the first line of the buffer.

cut

Default key assignments: Shift+Del, Gray -

Deletes the selected text from the buffer displayed in the current edit window and moves it to the Clipboard.

The selected text must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

If the **Cut/copy current line if no text is selected** checkbox in the Preferences dialog box is checked, the current line will be used for this command if no text is selected.

Identical to the Cut command on the Edit Menu.

Related topics:

[cut_block](#)

[cut_line](#)

cut_block

Deletes the selected text from the buffer displayed in the current edit window and moves it to the Clipboard.

The selected text must be less than 64K bytes in size. If it is larger, the message "Block too big" will be displayed and the operation will not be performed.

Related topics:

[cut](#)

[cut_line](#)

cut_line

Deletes the line containing the cursor from the buffer displayed in the current edit window and moves it to the Clipboard. The cursor is placed at column 1 of the next line.

Related topics:

[cut](#)

[cut_block](#)

delete

Default key assignment: Del

Deletes selected text or, if no text is selected, the character at the cursor.

Related topic:

[Delete](#)

delete_line

Default key assignment: Alt+D

Deletes the line containing the cursor. The cursor is moved to the first column of the following line.

delete_to_eol

Default key assignment: Alt+K

Deletes the text from the current cursor position to the end of the line.

delete_word_left

Default key assignment: Ctrl+BkSp

If cursor is within a word, deletes the text from the current cursor position to the beginning of the word. If the character to the left of the cursor is a space or tab character, deletes the text from the current cursor position to the previous non-blank character. If the character to the left of the cursor is one of the characters that delimit a word (but not a space or tab character), deletes that character.

Related topic:

[Delimiters](#)

delete_word_right

Default key assignment: Ctrl+Del

If the cursor is within a word, deletes the text from the current cursor position to the end of the word, plus any following spaces or tab characters. If the character at the cursor is a space or tab character, deletes the text from the current cursor location up to the next non-blank character. If the character at the cursor is at one of the characters that delimit a word (but not a space or tab character), deletes that character.

Related topic:

[Delimiters](#)

end_of_buffer

Default key assignment: Ctrl+End

Moves the cursor to the end of the buffer.

end_of_line

Default key assignment: End

Moves the cursor to the end of the current line.

exit

Default key assignment: Alt+F4

Prompts to save any modified buffers to disk and ends the editor session.

Related topics:

[exit_windows](#)

[save_all_exit](#)

exit_windows

Prompts to save any modified buffers to disk and issues an **ExitWindows** call to end the current Windows session. The Windows session is not terminated unless all applications agree to terminate.

Related topics:

[exit](#)

[save_all_exit_windows](#)

find

Default key assignment: Ctrl+S

Identical to the Find command on the Search Menu.

find_again

Default key assignment: Ctrl+A

Identical to the Find Again command on the Search Menu.

find_files_containing

Default key assignment: Ctrl+F

Identical to the Find Files Containing... command on the Search Menu.

goto_line

Default key assignment: Ctrl+G

Identical to the Line command on the Goto Menu.

goto_tag

Prompts for a tag and looks it up in the specified tag file. If the tag is found, opens the source file containing the definition of the tagged function and places the cursor at the start of the definition.

To use this function, you must supply a standard format tag file. The following formats are supported:

```
TagName c:\dir\filename.ext ^void TagName(void)
```

```
TagName c:\dir\filename.ext ?^void TagName(void)$?
```

(The drive and directory may be omitted.)

The name of a default tag file may be specified in the [File Defaults](#) dialog box on the [Options Menu](#).

Related topic:

[ctag](#)

editor_help

Default key assignment: F1

Enters the editor's help system. Identical to the Editor Help command on the Help menu.

Related topics:

[alt_help](#)

[sdk_function_help](#)

[sdk_help](#)

enter

Default key assignment: Enter

In insert mode, inserts a carriage return and line feed at the current cursor location. If autoindent is enabled for the current buffer, the cursor is placed under the first non-blank character on the previous line.

In overwrite mode, the cursor is simply moved to the first column of the next line. If autoindent is enabled, the cursor is moved to the first non-blank character on the next line.

indent_block

Default key assignment: Alt+I

Identical to the Indent command on the Edit Menu.

list_files_containing

Default key assignment: Ctrl+L

Identical to the List Files Containing... command on the Search Menu.

lowercase

Converts all upper case characters in the marked block to lower case. All other characters are left unchanged.

If no block is marked, this function acts on the character at the current cursor position.

Related topics:

[change_case](#)

[uppercase](#)

match_delimiter

Default key assignment: Ctrl+M

Identical to the Matching Delimiter command on the Goto Menu.

msc_error

With the cursor positioned on any line of a Microsoft C compiler error output file, jumps to the line in the source file that caused the error, with the error message displayed on the status line. For this function to work properly, the current line in the error output file must contain an actual error or warning message. The cursor does not need to be at the beginning of the line.

The function [show_message](#) may be used at any time to display the full text of the current error or warning message without switching back to the error output file.

After you have used this function to establish a starting point in the error output file, you may use [msc_next_error](#) and [msc_prev_error](#) to move directly to the next or previous error.

Related topic:

[bc_error](#)

msc_next_error

Finds the next error message in a Microsoft C compiler error output file and jumps to the line in the source file that caused the error, with the error message displayed on the status line. For this function to work properly, you must first use [msc_error](#) to establish a starting point in the error output file.

The function [show_message](#) may be used at any time to display the full text of the current error or warning message without switching back to the error output file.

Related topics:

[bc_error](#)

[msc_prev_error](#)

msc_prev_error

Finds the previous error message in a Microsoft C compiler error output file and jumps to the line in the source file that caused the error, with the error message displayed on the status line. For this function to work properly, you must first use [msc_error](#) to establish a starting point in the error output file.

The function [show_message](#) may be used at any time to display the full text of the current error or warning message without switching back to the error output file.

Related topics:

[bc_error](#)

[msc_next_error](#)

new_file

Identical to the New command on the File Menu.

new_window

Creates a new window on the buffer being edited in the current window. The new window becomes the current window, and the cursor is placed at the beginning of the buffer. Identical to the New Window command on the Window Menu.

next_window

Default key assignments: Ctrl+F6, Alt+N

Switches to the next window on the editor's circular list.

open_file

Default key assignment: F3

Identical to the Open command on the File Menu.

page_down

Default key assignment: PgDn

Moves the cursor down one page. If possible, the cursor column remains the same. Does nothing if the cursor is at the end of the buffer.

page_up

Default key assignment: PgUp

Moves the cursor up one page. If possible, the cursor column remains the same. Does nothing if the cursor is at the beginning of the buffer.

paste

Default key assignment: Shift+Ins

Identical to the Paste command on the Edit Menu.

play_macro

Default key assignment: F8

Identical to the Play command on the Macro Menu.

prev_window

Switches to the previous window on the editor's circular list.

print

Default key assignment: Alt+P

Identical to the Print command on the File Menu.

record_macro

Default key assignment: F7

Identical to the Record/Stop Recording command on the Macro Menu.

replace

Default key assignment: Ctrl+R

Identical to the Replace command on the Search Menu.

restore_window

Default key assignment: Ctrl+F5

Restores the current window to its normal (unmaximized) size.

revert

Identical to the Revert command on the File Menu.

run

Identical to the Run command on the File Menu.

save_all

Saves the contents of all modified buffers to disk.

Related topics:

[save_file](#)

[save_all_exit](#)

save_all_exit

Default key assignment: Alt+X

Saves the contents of all modified buffers to disk and ends the editor session.

Related topics:

[exit](#)

[save_all](#)

[save_all_exit_windows](#)

save_all_exit_windows

Saves the contents of all modified buffers to disk and issues an **ExitWindows** call to end the current Windows session. The Windows session is not terminated unless all applications agree to terminate.

Related topics:

[exit_windows](#)

[save_all](#)

[save_all_exit](#)

save_file

Default key assignment: F2

Saves the contents of the current buffer to disk. If the buffer is untitled, you will be prompted to specify a filename. If the buffer has not been modified since it was last saved, it will not be saved (the message "Buffer not modified" will be displayed on the status line). Identical to the Save command on the File Menu.

Related topic:

[save_file_close_window](#)

save_file_close_window

Saves the contents of the current buffer to disk and, if the save was successful, closes the current edit window. If the buffer is untitled, you will be prompted to specify a filename. The buffer will not be saved if it has not been modified.

sdk_function_help

Default key assignment: Shift+F1

Looks up the token at the current cursor location in the Microsoft Windows Software Development Kit help system.

Under Windows 3.0 only, the token at the current cursor location must be the name of a Windows message or API function, and " message" or " function" is appended to the token before it is passed to the help system. This is required by the design of the Windows 3.0 SDK Help Index and the inability of the Windows 3.0 help engine to search on partial keys.

The SDK help file (**sdkwins.hlp** or **win31wh.hlp**) must be available in a directory listed in the PATH environment variable.

Related topic:

[alt_help](#)

[editor_help](#)

[sdk_help](#)

sdk_help

Default key assignment: Alt+F1

Invokes the Microsoft Windows Software Development Kit help system. For this command to work, the SDK help file (**sdkwin.hlp** or **win31wh.hlp**) must be available in a directory listed in the PATH environment variable. Identical to the SDK Help command on the Help menu.

Related topic:

[alt_help](#)

[editor_help](#)

[sdk_function_help](#)

select_all

Selects all the text in the current buffer. Exactly equivalent to the sequence beginning_of buffer
select_to_end.

select_char_left

Default key assignment: Shift+Left arrow

Selects the character to the left of the cursor, or cancels the selection if the character is already selected. If the cursor is at the beginning of a line, it is moved to the end of the previous line. Does nothing at the beginning of the buffer.

select_char_right

Default key assignment: Shift+Right arrow

Selects the character to the right of the cursor, or cancels the selection if the character is already selected. If the cursor is at the end of a line, it is moved to the beginning of the following line. Does nothing at the end of the buffer.

select_line

Selects the current line.

select_line_down

Default key assignment: Shift+Down arrow

Extends the current selection down one line, or cancels the selection if the text is already selected. If possible, the cursor column remains the same. Does nothing if the cursor is on the last line of the buffer.

select_line_up

Default key assignment: Shift+Up arrow

Extends the current selection up one line, or cancels the selection if the text is already selected. If possible, the cursor column remains the same. Does nothing if the cursor is on the first line of the buffer.

select_page_down

Default key assignment: Shift+PgDn

Extends the current selection down one page, or cancels the selection if the text is already selected. If possible, the cursor column remains the same. Does nothing if the cursor is at the end of the buffer.

select_page_up

Default key assignment: Shift+PgUp

Extends the current selection up one page, or cancels the selection if the text is already selected. If possible, the cursor column remains the same. Does nothing if the cursor is at the beginning of the buffer.

select_to_beginning

Default key assignment: Ctrl+Shift+Home

Selects the text from the current cursor position to the beginning of the buffer,.

select_to_bol

Default key assignment: Shift+Home

Selects the text from the current cursor position to the beginning of the current line, or cancels the selection if the text is already selected.

select_to_end

Default key assignment: Ctrl+Shift+End

Selects the text from the current cursor position to the end of the buffer,.

select_to_eol

Default key assignment: Shift+End

Selects the text from the current cursor position to the end of the current line, or cancels the selection if the text is already selected.

select_word

Default key assignment: double click left mouse button

Selects the word containing the cursor.

select_word_left

Default key assignment: Ctrl+Shift+Left arrow

Extends the selection to the beginning of the word to the left of the cursor, or cancels the selection if the text is already selected.

select_word_right

Default key assignment: Ctrl+Shift+Right arrow

Extends the selection to the beginning of the word to the right of the cursor, or cancels the selection if the text is already selected.

show_message

Displays the full text of the current error or warning message.

split_line

Default key assignment: Ctrl+N

Inserts a line break at the current cursor location without advancing the cursor.

stamp

Default key assignment: Shift+F2

Inserts the date and time at the current cursor location.

tab_right

Default key assignment: Tab

In insert mode, if the **Expand tabs with spaces** options is selected for the current buffer, inserts a sufficient number of spaces at the current cursor location to advance the cursor to the next tab position. If the **Expand tabs with spaces** option is not selected, inserts a single tab character at the current cursor location.

In overwrite mode, moves the cursor to the next tab position.

to_bottom

Default key assignment: Ctrl+B

Moves the line containing the cursor to the bottom of the screen. The position of the cursor in the buffer is not changed.

Related topics:

[to_center](#)

[to_top](#)

[window_down](#)

[window_up](#)

to_center

Default key assignment: Ctrl+C

Moves the line containing the cursor to the center of the screen. The position of the cursor in the buffer is not changed.

Related topics:

[to_bottom](#)

[to_top](#)

[window_down](#)

[window_up](#)

to_top

Default key assignment: Ctrl+T

Moves the line containing the cursor to the top of the screen. The position of the cursor in the buffer is not changed.

Related topics:

[to_bottom](#)

[to_center](#)

[window_down](#)

[window_up](#)

toggle_backup

Toggles the **Make backup files** option in the File Defaults dialog box.

toggle_insert

Default key assignment: Ins

Toggles between insert mode and overwrite mode.

In insert mode (the default), each typed character is inserted in front of the character at the current cursor location; in overwrite mode, each typed character replaces the character at the current cursor location. In overwrite mode, **OVR** is displayed on the status line.

toggle_wordwrap

Default key assignment: Ctrl+W

Enables/disables word wrap in the current buffer

When word wrap is enabled, if the word being typed extends past the right margin specified for the current buffer, it will be moved to the next line. For the purposes of word wrap, a word is defined as any continuous string of non-blank characters. When word wrap is enabled, **WRAP** is displayed on the status line.

Word wrap is enabled/disabled on a buffer-by-buffer basis.

The right margin may be changed from the Current Buffer dialog box on the Options menu.

top_of_window

Default key assignment: Ctrl+PgUp

Moves the cursor to the top line of the window. If possible, the cursor column remains the same.

tpw_error

With the cursor positioned on any line of a Borland Turbo Pascal for Windows error output file, jumps to the line in the source file that caused the error, with the error message displayed on the status line. For this function to work properly, the current line in the error output file must contain an actual error or warning message. The cursor does not need to be at the beginning of the line.

The function show_message may be used at any time to display the full text of the current error or warning message without switch back to the error output file.

unindent_block

Default key assignment: Alt+U

Identical to the Unindent command on the Edit Menu.

undo

Default key assignment: Alt+BkSp, Gray *

Identical to the Undo command on the Edit Menu.

unmark_block

Default key assignment: Esc

Deselects the selected text and returns the cursor to its location before the text was selected. If no text is selected, does nothing.

uppercase

Converts all lower case characters in the marked block to upper case. All other characters are left unchanged.

If no block is marked, this function acts on the character at the current cursor position.

Related topics:

[change_case](#)

[lowercase](#)

window_down

Default key assignment: Ctrl+Up

Moves the cursor up one line and at the same time moves the text in the window down one line, so the cursor remains in the same relative position in the window. This function is useful if you want to move the cursor in the buffer but still keep it in the middle of the window.

Related topics:

[to_bottom](#)

[to_center](#)

[to_top](#)

[window_up](#)

window_list

Default key assignments: F6, right mouse button

Identical to the List command on the Window Menu

window_up

Default key assignment: Ctrl+Down

Moves the cursor down one line and at the same time moves the text in the window up one line, so the cursor remains in the same relative position in the window. This function is useful if you want to move the cursor in the buffer but still keep it in the middle of the window.

Related topics:

[to_bottom](#)

[to_center](#)

[to_top](#)

[window_down](#)

word_left

Default key assignment: Ctrl+Left arrow

Moves the cursor to the beginning of the previous word or to the end of the previous line, whichever comes first.

word_right

Default key assignment: Ctrl+Right arrow

Moves the cursor to the beginning of the following word or to the end of the line, whichever comes first.

wrap_para

Default key assignment: F12

Reformats the current paragraph (or the previous paragraph, if the cursor is between paragraphs) and leaves the cursor at the end of the reformatted paragraph. Does nothing if word wrap is turned off.

zoom_window

Default key assignment: Ctrl+F10

Maximizes the current edit window.

Paragraph

A paragraph is a group of non-blank lines preceded by a blank line and followed by a blank line.