



Carets & Cursors Index



Menus

[Purpose Menu](#)

[Choosing Cursors Menu](#)

[Choosing Carets Menu](#)

[Blink Choices Menu](#)

Information

[Class Names described](#)

[Spy Feature](#)

[Customizing Carets and Cursors](#)

[Automatic Small Startup Feature](#)

[Update History](#)

Problems

[Refresh CCdata.DTA if problems](#)

Purpose

Carets & Cursors is a copyright product of Instant Replay Corporation. Salt Lake City, Utah. You may contact Instant Replay at 1-800-388-8086.

Carets & Cursors provides custom control over the text editing carets and mouse cursors for all windows applications. This will provide easier recognition and more meaningful representations.

A system wide replacement caret can be chosen, providing consistency across applications.

It is possible to assign various cursors to window classes. Class definitions are easily added or removed and class assignments are made with visual selection menus. The replacement caret and all cursor class assignments are saved to disk (CCdata.DTA) for the next time Carets & Cursors is used.

Different assignments can be made for any window class; even within the same application. For example, assignments can cause the mouse cursor to change shape when the mouse is placed on a button, edit windows, a dialog item, etc.

Choosing Cursors

To choose a new mouse cursor, select "Cursors..." from the main menu. The cursor selection window will then appear. Notice that the available cursor choices are listed at the top of the window. These are bitmap representations of the cursors. The cyan color, usually the outside part, represents the transparent or "see through" area. The white and black represent the visible cursor, and the light red area represents the reverse video area of the cursor. For information on changing any of these cursors, see "Customizing" in the help index .

If you take the mouse and click on any of the cursors, that cursor is then placed in the selected cursor box located near the lower right part of the window. Go ahead and try it with some of the cursors. You can always cancel any changes made to the cursor selection window by using the "Cancel" button.

When you select a cursor, you are assigning that cursor to the current class name located in the box to the left. Most windows, like buttons, have only one cursor associated with them. There are some window classes that have more than one cursor. When an application uses more than one cursor in the same window class they are creating "smart" cursors that adapt to the objects displayed in the window. An example is the main editing window of Word for Windows "OpusWwd." If you use one of these "smart" window classes, Carets & Cursors will attempt to determine the dominant cursor. For example, with Word for Windows Carets & Cursors has determined that the IBeam cursor is dominant and will replace it and no other cursor used in that class.

It may be desirable to have all cursor always replaced in some windows. Choose the Replace Always radio button for any window class to accomplish this.

The nice part about using class names is that Carets & Cursors can be molded dynamically. For example, you can assign a unique cursor such as a hand to buttons. Then, when you point at a button the cursor will change instantly to a hand, providing feedback that you are about to push a button.

You can select a different current class name from the class name list box. The class name list box contains all of the class names you have entered into Carets & Cursors. These names along with their assigned cursors are stored in the CCdata.DTA file. For more information on window classes see "Class Names described" in the index.

Before getting into the topic of adding class names to the class names list box, you may want to try assigning a cursor to the "EDIT" class name. To do this make sure that "EDIT" is selected as the current class name by clicking on it in the class name list box. If you are a first time user, "EDIT" is probably already the current class name. Then choose a cursor by clicking on one of the cursors located at the top of the window. Finally, to save this assignment, click the OK button.

"EDIT" is a common class of window used in many applications for entering text. For example, the Notepad accessory program that comes with windows uses this class name. So to try your new cursor, run the Notepad program.

You can maintain the class name list with the Add, Change, Delete and Spy buttons. To delete a class name from the list, choose the name as the current class name and press the Delete button. To Change a name, select the name as the current class name, then change the name in the current class name box and finally press the change button. To add a new class name, enter the new name into the current class name box and press the Add button. After adding a new class name you will need to choose a cursor for the new class name.

Otherwise the original application cursor designated by the AP mark will continue to be used.

You can assign the original application cursor back to a class name by choosing the AP mark , but you will need to restart the application if it is already running to restore the original.

Spy Feature

If you do not know the window class names used in a program, which is very likely, you can find or "bag" them with the Point Spy button located in the Cursors dialog window. Just press the Point Spy button and your cursor is armed. The program manager will become available so you can run the program you want to spy on. If the program is already running, press Ctrl+Esc for the "Task List" and go to the program. As you move to each different window class the armed cursor will beep. Find the one whose cursor you want to replace and press the HOME key. You will return back to Carets & Cursors. The windows class name will be placed in the edit box. You will need to press the ADD button to add the window to the list. You can then choose a replacement cursor.

The List Spy button will list all windows and their class names. It provides interesting information about programs and their windows.

Choosing Carets

To choose a new keyboard caret, select "Carets..." from the main menu. The caret selection window will then appear. Notice that the available caret choices are listed at the top of the window. These carets are displayed in the Window Text Color in order to see the actual colors of these carets based on the window text color setup for windows. If you change the window text color, then the colors for these carets will also change because of the way windows displays carets. For information on changing any of these carets, see "Customizing" in the help index .

If you take the mouse and click on any of the carets, that caret is then placed in the Replacement Caret selection box located near the bottom part of the window. Go ahead and try it with some of the carets. You can always cancel any changes made to the caret selection window by using the "Cancel" button.

When you select a caret, you are assigning that caret as the system wide caret, overriding any carets used by windows and windows applications. This gives you the ability to establish a consistent caret of your own choosing across windows applications. The only caret not overridden is the special case "gray" caret, created in applications when they passing a handle value of 1 to the CreateCaret function. The replacement caret choice is stored in the CCdata.DTA file. Unlike cursors, not all window classes use carets. Carets are the equivalent of keyboard cursors in DOS programs. Carets are used basically for entering text.

Some programs that appear to be using carets are actually not. Good examples are Word for Windows and the Write accessory program included with Windows. The caret marker in Word for Windows is not the windows caret. It is a graphical marker specific to Word for Windows. WordPerfect for Windows does use the standard caret. You may need to experiment with some programs in order to find out if they are using a windows caret. If Carets & Cursors does not replace the text marker you can be certain the application program is not using the windows caret. Carets & Cursors can replace all carets created by any windows program.

"EDIT" is a common class of window used in many applications for entering text. For example, the Notepad accessory program that comes with windows uses this class along with a windows caret. To try any caret selection, run the Notepad program.

You can assign the original application carets back to all programs by choosing the AP caret.

Even with programs like Word for Windows that use their own text marker, it is still likely that you can control the blinking speed, or even disable blinking, with Carets & Cursors. This is because well written programs check the current caret blinking speed in order to emulate the windows caret.

To undo a caret selection use the Cancel button. To accept and save changes use the OK button.

Non-Replace Width options are used to change the applications caret width even when the caret is not actually replaced. The caret is not replaced when the AP "application" button is selected or when the application caret height is greater than 16 pixels.

Blink Choices

To modify caret blinking, select "Blink..." from the main menu. Then for standard caret blinking, select the ENABLE caret blinking radio button. For non-blinking carets select the DISABLE caret blinking radio button. Use the blinking rate scroll bar to adjust how fast the caret will blink when blinking is enabled.

To undo changes use the Cancel button. To accept and save changes use the OK button.

Users of the popular Noblink Accelerator, a cursor enhancement utility for DOS based programs from Instant Replay Corp. have inundated Instant Replay with requests for a similar utility for windows in order to disable the "obnoxious" blinking for text editing. However, not all programs use the windows caret for text editing, though most do. Some programs such as Word for Windows use their own pseudo caret. Even so, the program may interrogate the established blinking rate for the windows caret. in order to mock the blinking rate. Because of this, disabled blinking is usually effective or partially effective even with these programs. For example, Carets & Cursors can disable caret blinking within Word for Windows, but if the caret is not moved for a long period of time it will disappear until it is moved.

Class Names

A window class controls its appearance and behavior, including the shape and size of the cursor used when the mouse is moved into the window. Many applications use predefined system classes for their windows. Programs can also have their own window classes with their own definitions. Carets & Cursors monitors and overrides cursor definitions within window classes according to your choices. To discover the names of application specific classes you will probably need to use the SPY button in the Cursor selection window. The following are examples of system global class names.

BUTTON,
EDIT
LISTBOX
COMBOBOX
SCROLLBAR
STATIC
MDICLIENT

Use the Point Spy button in the Cursors Dialog window to find window class names. Some class names are numbers. For example, #32770. If you want to control the cursor for one of these classes, you must include the # when you enter the class into Carets & Cursors.

Customizing Carets and Cursors

It is not recommended that you modify the Carets and Cursors included with this product. However, since these are standard resources the following information will help if you are so inclined. Modification of any Instant Replay copyright notices is forbidden.

1. You will need a resource development product such as the Whitewater Resource Toolkit or Borlands Resource Workshop.
2. Caret and cursor images are contained in CCLIB.DLL. Make a backup of this file!
3. Carets are bitmap images that start at 1051.
4. Cursors are cursor resources that start at 1051.
5. There are equivalent cursor bitmap images for use in the Cursor Selection Window that start at 11051. These will need to be changed also so that the cursor menu selections match the actual cursor.

Refreshing the CCdata.DTA file

The CCdata.DTA file contains all of your classname caret and cursor assignments. If this file becomes corrupted Carets & Cursors may not function properly. To refresh this file, simply delete it. Carets & Cursors will create a new CCdata.DTA file if one is not present.

Startup Feature

**** This feature is not available in the DEMO Version ****

You can have Windows start Carets & Cursors automatically when Windows is started. When you do so C & C will start iconized.

To have Windows start Carets & Cursors automatically, place the following entry in the WIN.INI file under [windows].

load=c:\carcur\cc.exe

Of course the path may be adjusted as necessary.

Carets & Cursors will start minimized to an ICON when initialized in this way. You may want to remove the Carets & Cursors group from the program manager.

More than one program name can be specified on the load line by separating each name by a space.

Update History

12/1/91 Robert Adamson
Original Release of Version 1.0.

12/23/91 RA

Version 1.1

Implemented support for multiple application cursors within the same window class. Complex programs such as Word for Windows can change the cursor within the same window class. For example, moving the cursor over a frame or picture is indicated by changing the cursor shape. Version 1.1 replaces only the first or main cursor for a given window class.

1.0 would replace the hour glass wait cursor and the dragged ICON cursors if the "PMITEM" class was assigned a C&C cursor because ICONS are converted to cursors when dragged. Both problems are fixed in version 1.1.

1/17/92 RA

Version 1.2

Improved support for "smart" or multiple cursors in the same class. Fixes applied on 12/23/91 were too dumb for "smart" cursors. For example, the first cursor is no indicator of the dominant cursor. C&C now determines the dominant cursor for replacement. Tested extensively with Word for Windows 2.0 "OpusWwd" main editing window class.

Added Dominant and Replace Always radio button choices. Some windows, such as PBpaint have multiple dominant cursors that should always be replaced.

Boundary cursors for stretching were also improperly replaced by the dumb version of 12/23/91. Tested extensively in Borlands integrated environment editors.

WordPerfect for Windows users complained that the replacement carets were too small when used in large font areas. C&C now provides for automatic mutant width enlargement of caret when aligned on large characters. Enlargement is determined by the Non-Replace Width options on the Cursor Dialog. Version 2.0 of C&C is scheduled to have proportional carets.

Added Point Spy feature. The Listing Spy facility is interesting but very difficult to use. Trying to associate windows and names was too hard. The Point Spy makes life much better. Just arm the cursor, find the window and bag the class name.

1/30/92 RA

Width enlargement caused a problem with Ami Pro 2.0 where a solid block would extend from the left side of the screen. This was caused by Ami Pro pseudo caret using abnormal height values. The solution was to recognize pseudo cursor by abnormal heights. Also tested non-blinking caret and dominant cursors extensively in Ami Pro.

2/14/92 RA

Version 1.3

Hiking through a canyon in Southern Utah's Dixie I stumbled across dinosaur tracks that had been stretched. When I got back, I used a clear mind to write Proportional Stretching and Shrinking Carets. The Carets now retain their shape while enlarging or shrinking automatically for font sizes.

Replaced two unpopular cursors with a large pointing hand and a pencil for presentation

purposes.

2/27/92 RA

Changed automatic startup feature to use Load= instead of Run= in the win.ini file.

Removed obnoxious periodic pop-up message in DEMO Version. Disabled Automatic Small Startup feature instead. Pop-up Message itself was adversely effecting evaluations. Also discovered through complaints that if a mouse button was pressed when the pop-up occurred, then windows froze with message intact. This was not intentional! Easy to fix but opted for a less intrusive approach. Anyone needing Automatic Startup feature is beyond evaluating , so this seems like a fair approach.