



# **QuickCursor**

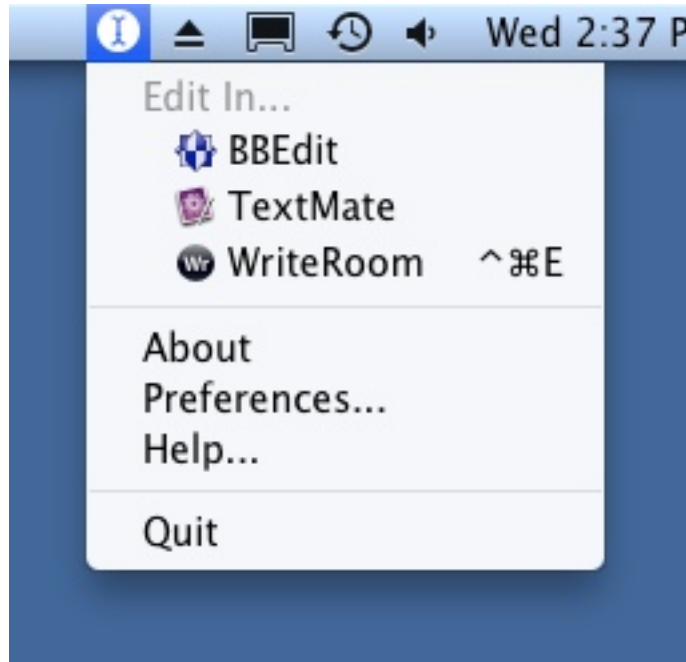
## User's Guide

<b>Welcome</b>	<b>3</b>
<b>Working with QuickCursor</b>	<b>3</b>
To edit text in your favorite text editor	3
To paste edited text back to the original app	4
To assign a global keyboard shortcut to an editor	4
To play nice with clipboard manager utilities	4
To set the temporary file's extension	4
To add compatible text editors	5
Supported text editors	6
<b>QuickCursor Tips &amp; Tricks</b>	<b>7</b>

# Welcome

QuickCursor brings your favorite text editor<sup>1</sup> to any app<sup>2</sup> with a global keyboard shortcut. Unlike manually copying and pasting between apps, QuickCursor automates the entire process to save you time.

## Working with QuickCursor



### To edit text in your favorite text editor

- Click in an editable text area in any app and select a range of text that you want to edit, or don't select any text to edit the entire text area.
- Choose the text editor that you want to use from QuickCursor's **Edit In...** menu item group. In QuickCursor's preferences you can define individual global keyboard shortcuts to start your favorite editors.
- Your text editor will be started and the text copied into a new document.

---

<sup>1</sup> Must be an editor that supports the ODB Editor Protocol.

<sup>2</sup> App must expose standard Mac OS X Copy & Paste menu items to Universal Access.

## To paste edited text back to the original app

- Choose Save (Command-S) in your editor app and then close (Command-W) the editor's document.
- When QuickCursor sees the document closed it will switch you back to your original app and paste in the updated text.

## To assign a global keyboard shortcut to an editor

- Click QuickCursor's icon on the OS X menu bar.
- Choose the **Preferences...** menu item.
- Choose an editor from the **Shortcuts** popup menu.
- Click in the **Shortcuts** text field and enter your keyboard shortcut.
- You will now be able to use that shortcut to begin editing text in any app in your chosen editor.

## To play nice with clipboard manager utilities

- Use the "Disable Clipboard Restore" control. QuickCursor will no longer restore the clipboard's previous contents.

## To set the temporary file's extension

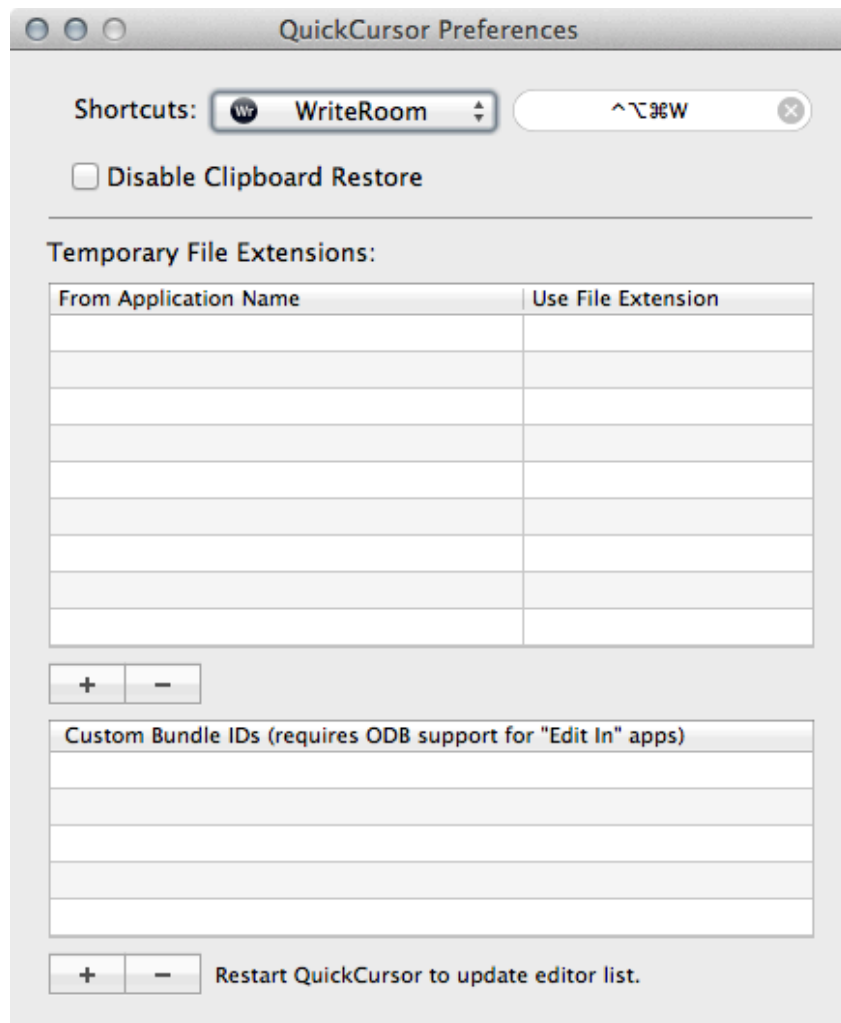
When QuickCursor moves text from a source app into your text editor, it saves that text in a temporary file that your editor then opens. By default that file has a .txt file extension. To use a different extension (to effect syntax highlighting for example):

- Click QuickCursor's icon on the OS X menu bar.
- Choose the **Preferences...** menu item.
- Add a new row to the **Temporary File Extensions** table.
- Type an application name such as "Safari" in the **From Application Name** field.
- Type a file extensions such as "markdown" in the **Use File Extension** field.
- Now when you use QuickCursor to edit text from Safari the temporary file that is used will have the .markdown file extension.

## To add compatible text editors

New in version 2.6: You can add compatible “Edit In...” text editors to QuickCursor by adding their bundle IDs. QuickCursor supports all the text editors listed in the next section by default.

- Click QuickCursor’s icon on the OS X menu bar.
- Choose the **Preferences...** menu item.
- Add a new row to the **Custom Bundle IDs** table.
- Type in the app’s bundle ID (for example, WriteRoom’s is ‘com.hogbaysoftware.WriteRoom.mac’).
- Quit and relaunch QuickCursor. The app, if installed and run at least once, should be in the list.
- **Note:** Add other text editors at your own risk. They are not guaranteed to work. A compatible app must support the ODB Editor Protocol as a text editor to work with QuickCursor. If you’re unsure about a particular app, contact its developer.



## Supported text editors

QuickCursor requires that **Edit In...** text editor apps support the [ODB Editor protocol](#). If the **Edit In...** menu item group isn't displaying any items make sure that you first install at least one of these supported editor apps first.

- [Aquamacs](#) For compatible emacs-based editors such as Aquamacs 2.2+, you may need to kill the buffer to send the text back to the original app. See the emacs wiki on this topic: <http://www.emacswiki.org/emacs/KillingBuffers>
  - If you are familiar with [EditorClient](#), you can use QuickCursor with most other Emacs installs too!
- [BBEdit](#) If you are using the BBEdit trial QuickCursor won't work unless BBEdit is already running. This happens because the registration screen that shows when BBEdit starts blocks access to the Paste menu item.
- [Byword](#)
- [CotEditor](#)
- [Espresso](#)
- [Fraise](#)
- [MacVim](#)
- [MultiMarkdown Composer](#)
- [Smultron](#)
- [SubEthaEdit](#)
- [TextMate](#) When "Editing in..." TextMate, you must Save (cmd+s) and then explicitly Close (cmd+w) the TextMate editor window to have your text sent back to the source app. Normal behavior should be to send the text back with a Save and Quit. Bug has been reported to TextMate.
- [TextWrangler](#)
- [WriteRoom](#)

# QuickCursor Tips & Tricks

- For new users: The Mac App Store will install a QuickCursor Dock item during the download. If you have added QuickCursor your Login Items (System Preferences > Users & Groups) then you can safely remove the app icon from your dock and not worry about it taking up extra space.
- QuickCursor works by automating Copy & Paste operations using Universal Access' "Enable access for assistive devices" option. If something goes wrong and your edited text isn't sent back to the original app you can generally manually paste it into another app (always works when using the "Disable Clipboard Restore" option in preferences).
- QuickCursor can have trouble finding and working with text inside of some runtime environments. Known issues include text inside of Eclipse (or other apps that use the Java Runtime Environment) and Wunderkit.
- If you are a programmer and want to modify the way that QuickCursor works for your own personal use, I've made the basic source code available on [GitHub](#). Please don't re-distribute clones or similar apps based on this code. I hope you enjoy hacking away.