Edit Comment Version 1.2

What it is

Edit Comment is ShareWare. The fee is $25. I sincerely thank you for your support.

Please send registration fee to:
Larry Gerndt
877 Heatherstone Way #201
Mountain View, CA 94040

What it does

This utility allows you to edit the comment associated with a Think C file under control of Projector (or SourceServer), Apple's source code control system. Projector accomplishes its source control by attaching a resource of type 'ckid' to each file it controls. The information in the 'ckid' resource includes the current revision number and a comment about the changes in that revision. This is what Edit Comment edits. Edit Comment adds an Edit Comment... menu item to Think's Edit menu which, when selected, extracts the comment text from the 'ckid' of the file you're currently editing and puts up an editing window that shows the current revision and displays the comment text in an editable field. This makes it easy to log the changes you make to a file as you're making them, and they stay in the 'ckid' resource, so that when you go to check in the file, you can leave the comment field in Projector's Check In window blank and just check in.

The motive

My company develops exclusively with Think C, and recently we began using Projector. One nice thing about Projector is that you can browse through the revision tree of a file to see what was changed in each revision by reading the check-in comments of each revision. We've found this to be helpful in many circumstances, such as trying to figure which revision broke something, etc.

Often however, I change numerous files in small ways before going to check them into Projector. Then, at check-in time, I often find that I can't remember what changes I made. I then have a choice of entering the comment "not sure what I did to this file, but it'll probably work better now" or doing a comparison (gag me). So, I decided to be a responsible member of society and switch to Projector to edit the comment every time I changed one thing. That got old real fast. I thought of maybe just accumulating the comments at the top of my file, but that was cumbersome too. That's why I wrote this utility. It makes it easy to enter and accumulate comments about the changes you're making right when you're making them, and you never have to copy and paste them into projector later, or try to remember all the changes you made when you go to check in.

How to install

Simply run the installer and click the Patch button. It will prompt you to find the Think C application, and will then add some resources to it which implement this utility. There is no INIT to install, no rebooting, and it does not change or remove any Think C resources. To remove the Edit Comment resources, click Unpatch.

How to use

Once this utility is installed, you'll notice a new Edit Comment... menu item in Think's Edit menu. When you have a file open, simply select Edit Comment... to bring up the comment editor. You'll see the comment text and may edit it. You'll also see some other projector-related information about the file displayed in this window, the most usefull perhaps being the revision number. While editing the comment, all other Think windows and menu items are disabled. When you're done editing, close the window either with the mouse or with CMD-W or CMD-S. Editing the comment does not change the mod date of the file. You will only be able to change the comment if the file is either checked out to you or if the file is a modified read-only file (i.e. you selected Modify Read-only from Think's File menu. By necessity, the Edit Comment window behaves as a movable modal dialog.

Implementation details for programmers

Programmers will probably want to know what exactly is installed. I simply add three resources to Think C: a 'WDEF' 0, a 'WIND' 400, and my main code module, called 'ECmt' 128. That's it. Now on to how it works. As Projector users probably know, all files under Projector control have a 'ckid' resource, which actually contains the comment field. When you go to Projector's Check In window and hit the Info button (the "?" button), you can see and edit the comment (if any) that's currently in the 'ckid' resource of the selected file. All I had to do was figure out a way to provide this same capability from within Think C on the fly (i.e without switching to Projector). By using Think C's published API for third party extensions, I knew I could get the name of the file displayed in the front window. But how could I edit that file if it's already open for write? Very fortunately, and I think wisely, Think C does not keep the file open. It simply opens it every time you save it, then closes it again. So there's no problem editing the file's 'ckid' resource while the window is open. Great! Now how do I implement the patch? I tried many variations of a trap-patching INIT and was unsatisfied. The patches were too intrusive and not exclusive enough to Think C. I settled on what I think is the best approach: I defined a WDEF 0, which is called when Think C creates or changes a window. This WDEF always calls the system WDEF, but then does its own stuff: During the very first wNew message, I load and execute my code ('ECmt' resource), which patches MenuSelect to call my code after calling the real MenuSelect, and adds an Edit Comment... item to THINK's Edit menu. This is done once only. From that point on, when the Edit Comment... item is selected, my patch to MenuSelect intercepts it and calls my code to provide the editing window. Since all my code runs within a menu selection, so to speak, Think C is frozen, so my window acts as a movable modal dialog. You can switch applications, but cannot do anything in any other Think windows. You'll probably also notice that if you move the edit window, any Think C windows underneath it are not updated until you close the edit window. There was simply no way I knew of to update those windows while I'm in control. I do make sure they're updated when I close the window by walking through Think's window list and invalidating all windows. You'll probably see the flicker from this. Finally, note that since each application has its own trap table, the patch to MenuSelect is completely isolated to Think C (a good quality).

Compatibility

Works with Think C 5.x, 6.0, and I use it with CMaster™ and have had no problems.

This program is provided as is, and is not guaranteed.

EMail:
America Online: OpLarry
APPLELINK: OPCODE

Please send registration fee to:
Larry Gerndt
877 Heatherstone Way #201
Mountain View, CA 94040