2 *Release Notes and some History*

This chapter contains the release notes to all the ClassEditor versions. It documents new features, the development steps taken, major bugs that have been fixed and bugs that are known.

Version 0.3

The first solid release escaped on 15.02.1995. Now it should really work as advertised. It still has many shortcomings but as long as you stick to the rules it won't corrupt your files¼as far as I can tell. All the classes of this release have been edited using the ClassEditor. Seems like that worked correctly.

New Features

Even more features out here. But many more are missing.

· **Nicer Icons.**
Some new artwork has been done and now it looks prettier all around the screen.

· **Auto parsing.**
You don't have to save the files anymore to see methods that have been added via the 'cheat window'. This is not really a new feature¼it's more a fixed bug :-)

· **FAT.**
We are quad- fad now (NeXT, Intel, HP and Sparc).

Bugs Fixed in Release 0.3

The last release should have been stable, but it was not. So I only focused on fixing the stupid mistakes that were still inside.

· **Pasteboard corruption.**
ClassEditor should never corrupt the pasteboard anymore. We use our own for internal work now.

· **Crashing with Miniwindows.**
Garbage collection should work properly now and you won't crash with a open miniwindow under any circumstances.

· **Parsing the Docus.**
Fixed the parsing bug that caused ClassEditor to corrupt the documentation files when the 'See also' references where used as they should be.

Known Problems

Some of the known problems won't hurt anybody at this moment ± I hope. But I know that they are in there.

- **Fails to open at Lines.**
  The app does not know how to open files at a certain line. So e.g. clicking on the ProjectBuilders error lines won't bring up the right method or even class. Soon to come!
- **Stupid Search Pathes.**
  The CEClassEditor.m file has some search pathes hardcoded. When they work but when saving everything will end up inside the place where the file you really opened came from. This is nasty.
- **No reverting.**
  You can't revert to the saved version without closing and reopening a class. ClassEditor will not recognize that a certain file was change by another application. Last saved...you win.
- **No Categories.**
  ClassEditor does not load category source files automatically. This is left to the user. Either merge all the sources (you *can* have multiple categories in there) or seperate the headerfiles on a per category basis.
- **Missing Methods.**
  You can only browse the methods that are inside the *saved* MyClass.h file. Be sure to save whenever you have added methods. Sometimes you need to reparse the method by hand.
- **No multiline Methods.**

Multiline methods are not supported. So please remember that. However, inside the documentation you *can* reformat a method to be multiline, but the selector still has to be inside a single line !

- **Missing Source.**
  If your source and header definitions for a method are not the same...Class Editor will not be able to find the source. You have to edit the source from inside the cheat window to match the definiton. But no saving is required.

- **Missing Headerfile = Crash.**
  In most cases you will crash when opening a *.m* file without having a *.h* file at hand.

- **No RTFD Docus.**
  At the moment we can't deal with RTFD documentation. Support is build in...but not enabled yet.

- **Ugly docu-creation.**
  The docus our auto-create mechanism spits out are not perfect. They *do* conform to NeXTs style but they do *not*:
  - add methods docu in a sorted order after the *first* creation.
  - add instance variables.
  - add methods to the overview
  - add any other dependance info

- **Won't remove Method Docus.**
  The **checkDocumentation:** method does not remove method documentation that don't have a matching definiton inside the headers. This is not a bug but a feature. Otherwise you would alwayy lose delegate methods descriptions and such.
  But we could add a panel that would allow the user to specify the methods that should be removed.

- **ASCII tabbing.**

When viewing source sometimes you have to live with stupid tab settings. Most of the time they are 1 Tab = 1 space. :-(
I really should have a look at it.

- **Windows out of sync.**
Sometimes you might think that the windows are out of sync. Mostly they are not...they only look like they would.

- **Jumping Selections.**
Heavy switching between the windows might cause that you won't end up at the same place where you left a certain window. Please check twice if your cursor is really at the position where you expect it to be.

- **Undo not possible.**
Sometimes the app won't be able to undo your changes. Remember that Undo works only for the currently edited method inside the real class winodw (not the cheat window).

- **Preferences won't save.**
No code for remembering the prefs has been written yet.

- **GUI Fakes.**
Not everything you see actually works. Using the split view under the browser, for removing the browser view, will corrupt the layout..etc..pp.

- **Closing Windows.**
When quitting windows are not closed in the same order as they appear on the screen. This might cause some confusing window swaps...but it works. With the current design it is not wise to siple close the topmost windows..one by one.
Future releases should fix that.

- **Code uglyness.**
Now this is really a project that is just intended to work in th first place. Maybe someday I might decide to make it a nice project ± code wise.

Development

Still a lot of work left¼but I have done something after all.

Feb. 95: Wrote some categories to the Text object to speed up the parsing nightmare. Some MiscKit classes had to go that way too. (4 days of coding)

Version 0.2

The second public release was on 25.01.1995. I tried to make a useful and save app out of the ugly first release. It doesn't corrupt my sources anymore.

New Features

What's new this time?

- **Emacs Keybindings.**
  With the help of the MiscShell palette I added the Emacs text object (by Julie Zelenski). Thanks to the IB that comes tih EOF I was able to fix the palettes resize bug.

- **Auto Fileformat detection.**
  ClassEditor now does handle RTF or ASCII versions of the souce files.
- **Proper exit and Window closing.**
  Closing the browser window frees the class document. You are get an alert if the contents hasn't been save yet. The app deals with quitting as it should according to NeXTs guide lines.
- **More Styles.**
  Added a "Bug Notes" and math symbol font style.
- **Documentation creation.**
  If there was no documentation available we will now create one from scratch.
- **Docu-Template.**
  To speed up the process of docu writing you can now access a docu-template that includes the most important text styles so you can cut/copy the layout (rulers and fonts).
- **Preferences.**
  Font settings work...but you can't save them. Sorry.
- **Mutlimethod Documentation.**
  You can have one documentation for many methods. Like: -setValue:, -setValue:after:.
  All you have to do is to stick to the layout templates. Take a look at the ExampleClass. You will find multimethod and multiline methodname documentation in there.
- **Undo.**
  You can Undo all the changes made inside the source and documentation. But you can only Undo *both* at once.
- **App.info.**
  The app comes with a useful *ClassEditor.info* file (similar to .info files of the Installer.app).

My *NewAppInspector* bundle for the Workspace will like this little extra description. Once I'll have the time to polish and finish it I will release this project too.

Bugs Fixed in Release 0.2

There were so many stupid bugs inside v0.1 that I won't list all the small fixes. These are the most serious bugs fixed.

- **Saving.**
  Changes should now be saved correctly. v0.1 lost changes to the visible method when edited in both windows.
- **Memory Leaks.**
  Now we really free our main editors. You need to switch between applications to trigger the private garbage collection system. But when developing this should happen quite often anyway.
- **Pasteboard Bug.**
  The app uses a private pasteboard hack now so in most case you should notice an internal pasteboard work.
- **Syncing.**
  Both windows should be in sync most all of the time. If they are not...they should be after switching the windows.

Development

Ok. Here is what I did to kick out this version.

Jan. 95: Spent most of the time fixing ugly code sections and making the programm useful. (3 days of coding)

Version 0.1

The first public release of 17.01.95. Therefore everything is new. Nothing will work as expected. This is a fast hack to trigger of some discussion and get some feedback in the early stage.

New Features

Here is a short summary of all the parts that are working now. See the introduction chapter for a list of ideas on future development efforts.

- **Browsering by Methods.**
  The app allows you to manipulate the source code on a per-method basis.
- **Loading Files**
  Open multiple sets of *.m*, *.h* and *.rtf* files.
- **View the Documentation.**
  You can view and edit the a method description side by side with its implementation.
- **Change Fonts.**
  Use the "Style" menu for quick access to the fonts you need for a nice docu sheet. The

most important fonts are accessable trough cmd-key shortcuts for rapid typing.

Development

Here I'll try to show you on which parts I did focus my work in this release.

Jan. 95: Mainly hacked up the NIB file and spent some time learning about the powerful search methods of the Text object :-). This one could really stand many cool categories or subclasses. (2 days of coding)