Release 0.4, 05.06.1995 by Thomas Engel (tomi@shinto.nbg.sub.org)

1 *The ClassEditor*

ClassEditorRunning.tiff ¬

This editor allows you to modify class definitions on a per-method basis. In additon to offering a method browser it will make it very simple to keep the documentation in sync with your code. Both elements are shown side by side.
The ideal situation would be to maintain a documentation file from the first moment since creating a new class or method. ClassEditor is able to create the docu tempates for you and helps maintaining them in a NeXT conform layout.

In the worst case you should think about this project as a RFD (request for discussion). Maybe someone at NeXT might find new ideas for NeXTSTEP 4.0 in there.

What it does

ClassEditor opens a set of *MyClass.m*, *MyClass.h* and *MyClass.rtf* files and displays them. You can edit and view the class on a per-method basis or see all three files at once inside the "cheat window".

Modifying existing classes that are not spread accross more then those three files is quite safe. Adding methods is possible from inside the "cheat window" but introduces some rough edges. See the bugs section for details.

264278_paste.eps ¬

Reliability Issues

Although the app has improved over the previous versions it still contains some bugs that might cause the app to crash or even corrupt your data.
I am aware of rare cases where the documentation got corrupted. Source files have not been damaged as far as I can tell. Drag operations might crash the app on a random basis due to a strange behavior of the Pasteboard class.

If you use this application to modify important files be sure that you read the *Release Notes* and *Warranty* section first. Remember that there is no guaranty for nothing.

754626_paste.eps ¬

The Package

Version 0.4 comes in a package that includes:

- ClassEditor.app. A FAT binary for NeXT, Intel, HP and Sparc hardware.
- An *Examples* directory containing working test setups of interface, implementation and documentation files for some dummy classes.
- Online help¼minimal
- Full source code
- The documentation and an ASCII version (README) of this introduction

Attention: You need gnutar and gzip und unpack those files. Both programs come bundled with NeXTSTEP since version 3.2.

On where to find the latest releases please see the '*About this Project*' section.

Features

Currently there are only very simple things this application will do for you. Now this is just an early alpha version¼don't expect it to do magical things anytime soon.

- Select methods and view or edit the description and implementation.
- Create missing pieces of class and method documentation using automatic text generation and useful ready-to-copy templates.
- Add "See also" references with a simple drag&drop.

- Use the "Style" menu to get fast access to the right fonts for the nasty documentation work.
- Undo the last changes inside a methods source or documentation.
- Use Emacs keybindings to navigate inside the text areas.
- View ASCII sources in a colorized style which makes editing a lot easier.
- Select the "Plain C-Stuff" entry from the mode popup and view (or edit) all three files inside the "cheat window". Add defines, typedefs, instance variables or new methods here.

Some features will definitly be improved to help me get along with all those missing class docus (yes¼MiscSwapKit docus are already waiting for too long).

The *Release Notes* chapter can give you a complete overview of all the implemented features. In also includes a detailed description of the currently known bugs.

Bugs and Birds

It is not fair to speak about them in a little sub paragraph. There are so many of them that it would justify a whole chapter.

This app is a piece of brain stroming Ð a running ToDo list. It is not a product. Peter L. would call it another piece of German software: nice idea but bound to never get finished.

Well maybe he is kind of right. Now lets face the most serious problems of this app:

- When working inside the "cheat window" you should take care of all the changes

you make. You should follow a very strict coding style. Otherwise ClassEditor won't be able to parse your code and might even corrupt your files. See the Examples for working layouts.

- RTF headers and sources will only work if the entire method names are typeset with a single font. Take a look at the RTFExampleClass to see how rich sources should look like to work with ClassEditor.
  Since v0.4 you have ASCII code colorization which makes RTF source more useless because there is no easy way to provide smart colorization of RTF¼which should preserve your private colorization.
- Many user interface controls are just fakes, the split views might cause your window to turn into an ugly piece of GUI when used to their limits and the drag&drop stuff might crash the application.

What it should

Of course this should become the killer application of the nineties. But I am lazy and I *do* believe in code reuse. The guys at NeXT have already done all the nasty work. So I only want to build a minimal tool to fill the gap until NeXTSTEP 4.0 arrives.

Perhaps this app will only lead to a simple font service application (btw. too many people use the wrong font for source code inside the docus..it is Courier 12pt¼not 14pt) or it will migrate to a real destributed class editor (the buzzword here is: groupware).
Who knows.

Future plans

My plans are to enjoy NeXT's upcoming ProjectBuilder *if* it comes close to the ClassEditors idea. As far as I can tell from all the published material I'm not sure that it will really cover all the fruits of desire.

The aim is to build a combination of a class browser with the freedom to write ugly and nasty C-style code while providing all the help an editor should give. This includes concurrent editing and versioning.

ClassEditor should split into the real editor and a ResourceServer. This would free the editor from many low-level issues which might be implemented quite individually (e.g. taking sources from a database and not a filesystem).

So lets see what this application is still missing. I do consider to add some of these features:

- Online source sanity checks. Checking braces is only a primitive step into that direction.
- Creating new classes, methods, categories and protocols on the fly. Specifying dependancies and relations should be as simple as a drag&drop. Grouping methods and adding descriptions to instance variables.
- The editor should be able to dump templates for get/set'er methods and other ever returning tasks.
- The special categories called (private) and (protected) might be used in the headerfiles and other sections by default. Every category should follow the *ClassName+categoryName.h* (*.m* or even *.rtf*) convention.

The editor should create those files for you if you don't want to organize it in your private style.
This would keep the code clean for fast and easy distribution.

- Prepare sources for public distribution by stripping protected methods from the headers and such.
- Support for concurrent editing over a network. This should not necessariy mean sharing the real source files. And DO approach would be much better.
- Maintaining a revision history. This could be done by including CVS support.
- Storing bug infos and idea notes together with the code and documentation.
- More support for "See also:" references and other logical links.

Some features can only be implemented if the whole development environment works together. This includes apps like InterfaceBuilder and therefore requires some open APIs provided by NeXT:

- Being sure that classes inside NIBs are always up to date with the code. I hate to have to remember all those dependancies.
- Speaking about IB; one of its nasty *features* is that it won't drop back to palettalized classes if it can't find the code for a certain subclass of them. In testmode this only works for subclasses of window¼why ??

It is true that many things can already be done right now¼but you have do them by hand. There often is no real support from the tools. It is time to change that.

About this Project

I don't see the need to rush for a much better version of this app until it is clear what NeXT will include with their NeXTSTEP 4.0 release. There are other projects I want to spend more time on.

The latest version of this editor will be available at the ftp.nmr.embl-heidelberg.de, ftp.informatik.uni-muenchen.de or ftp.cs.orst.edu anonymous ftp servers. There is a WWW page were you can find out more about the status of this¼and all the other projects I'm working on:

```
http://wwwcip.informatik.uni-erlangen.de/user/tsengel/Projects/Projects.html
```

Compiling

This version includes all the source code necessary to recompile the program from scratch. It does **not** come with all the libraries! You will need to get the MiscKit project (Version 1.5.0 or higher) from the archives.

For more details on the MiscKit collection see the common NeXT ftp servers, get the Objective-C or NeXT FAQs or take a look at the EduSteps developer area:

```
http://www.nmr.embl-heidelberg.de/eduStep/Developer/Kits/MiscKit/MiscKit.html
```

If you would like to submit something to this collection you should contact the current maintainer: `Don Yacktman <don@darth.byu.edu>`

In Case of Trouble

If you have any questions you can contact me.


```
Thomas Engel
Netpunstr. 9
D-90522 Oberasbach
Germany

E-mail:   tsengel@cip.informatik.uni-erlangen.de
          tomi@shinto.nbg.sub.org
```

(NeXTMail welcome)

Warranty and copyright

Copyleft

world.

Anyway¼comments are *highly* appreciated.

No Warranty

This software is provided '*as is*' and the programmer is not responsible for any harm this program may cause.
You Ð the user Ð are responsible for everything that may happen to your business, hardware, software, car, CD collection or what ever may be worth your attention or money.
Using this product is at your own risk and your private fun.

There should be no serious bugs inside but remember that a carbon-based unit did the coding.

Travelware

Like all my projects this app is free and should be considered as travelware.

166998_12ptHevBlk.tiff ¬ Let it travel to as many people as you know.

02_12ptHevBlk.tiff ¬ Send me an E-mail or postcard if you use it. I will try to keep you informed about new releases.

03_12ptHevBlk.tiff ¬ If you have a free bed or some free space on your floor¼give me a hint. I might come and visit you on my next trip to the US¼locations on

Hawaii, near Seattle or WhistlerMnt. (Canada) preferred.

Enjoy it.

.Unterschrift.tiff ¬