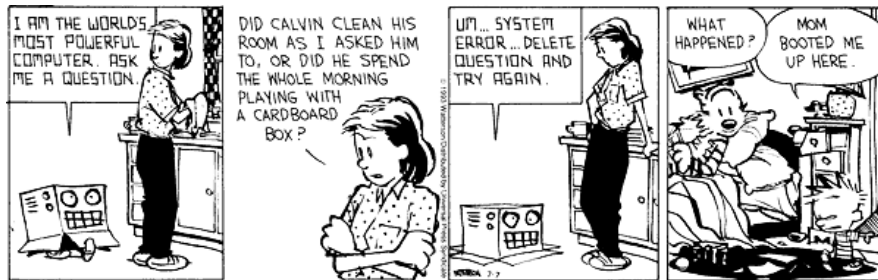


Assignment 1: Getting started

ETH Zurich

Hand-out: Monday, 22 September 2008



Calvin and Hobbes© Bill Watterson

1 Welcome to ETH!

Goal

The goal of this task is to familiarize you with the infrastructure provided by ETH Zurich and to make sure that you have subscribed to this course.

To do

- Log in to a computer in one of the computer rooms. The location of the computer rooms is given in the information sheet that you received.
- Log in to the n.ethz.ch account administration page (<http://www.passwort.ethz.ch/>). After logging in you should see the welcome screen (Figure 1). Click on “Passwort ändern”(top of screen), change your password¹ (Figure 2) and log out. After this you can log on to the ETH student web-mail interface (<https://mail.ethz.ch/exchange>) with your user name and your newly set password.
- Try to find the Eiffel development environment - EiffelStudio - on the machine that you are using.
- Bookmark the Forum der Informatik Studierenden: <http://forum.vis.ethz.ch>. The forum provides a platform for questions and discussions with your peers.
- **IMPORTANT:** Make sure that you have subscribed to this course (and all the other courses you attend) on <https://www.mystudies.ethz.ch>. Otherwise it is difficult to give you your testat at the end of the semester.

¹Your password should be simple enough for you to remember; yet it should be complex enough so that other people cannot guess it. See password recommendations at CERN: <http://security.web.cern.ch/security/passwords/>

Hint

You will find that your n.ethz login and password are very useful on many other ETH web pages, e.g. the <https://www.mystudies.ethz.ch> page mentioned above.

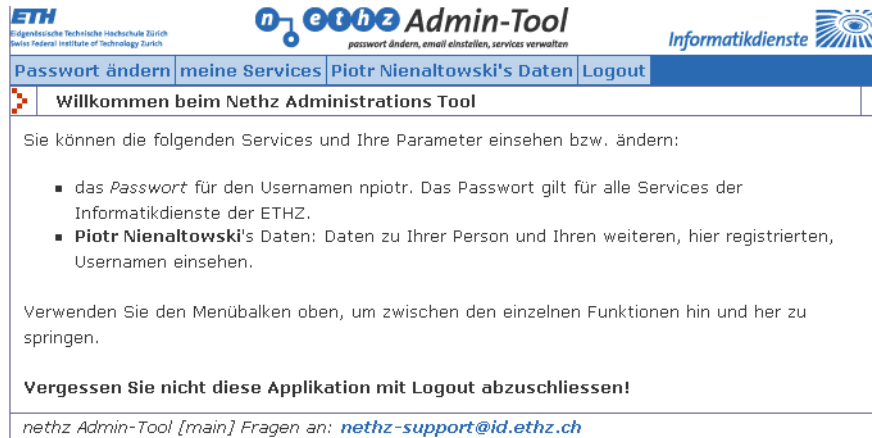


Figure 1: Welcome screen

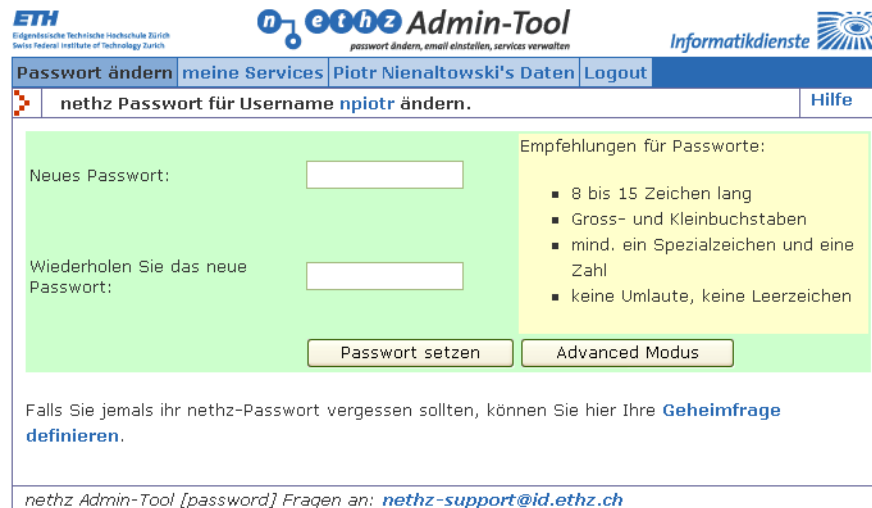


Figure 2: Change password screen

To hand in

There is nothing to hand in; the task is accomplished as soon as you have subscribed to the course.

2 EiffelStudio installation

Goal

In this task you will install the software development platform EiffelStudio. If you intend to work in a computer lab of ETH you will not need to install EiffelStudio. Proceed with task 3. We support the operating systems Linux and Windows (Microsoft C compiler only!); if you are a Mac user please talk to your assistant.

To do

For Windows

1. As a first step you need to get the Microsoft C compiler. If you already have it installed (e.g. as part of Microsoft VisualStudio), skip this point. To install Microsoft C compiler, follow the instructions at http://dev.eiffel.com/index.php/Installing_Microsoft_C_compiler.
2. Download EiffelStudio 6.2 (not 6.3!) from <http://eiffelstudio.origo.ethz.ch/download>. Make sure that the version you download is 73753 (e.g. Eiffel62_gpl_73753-windows.msi).
3. Start the installer and follow the instructions. Choose the option *EiffelBase, WEL and EiffelVision2* in the step "Precompiled Libraries". The precompilation takes a lot of time during installation, but significantly speeds up later compilations.

For Linux

Please note that we assume a basic level of preinstalled tools, such as tar, gcc, headerfiles, etc. These instructions have been tested under debian. We recommend to install EiffelStudio with the default locale (en_US).

1. Check your GTK+ version: EiffelStudio requires version 2.4.0 or above.
`pkg-config --modversion gtk+-2.0`
2. Pick a directory of your choice without any spaces in the path, for example /opt:
`cd /opt`
3. Download EiffelStudio 6.2 (not 6.3!) version 73753 at <http://eiffelstudio.origo.ethz.ch/download> and extract it.
`tar -xvjf Eiffel62_gpl_73753-linux-x86.tar.bz2`
4. Set the following environment variables (note that how and where to change environment variables depends on which distribution you use). Make sure you set these permanently and systemwide! Note: If you download the 64 bit version of EiffelStudio make sure to change the environment variable ISE_PLATFORM accordingly (as explained in the installation instructions of EiffelStudio).
(Type as root:)
`echo ISE_EIFFEL=/opt/Eiffel62 >> /etc/environment`
`echo ISE_PLATFORM=linux-x86 >> /etc/environment`
`echo PATH=$PATH:$ISE_EIFFEL/studio/spec/$ISE_PLATFORM/bin >> /etc/environment`
`echo source /etc/environment >> /etc/profile`
5. Restart the X server. You should now have the eiffel compiler (ec) in your path, as well as all environment variables set.

6. To speed things up later, we need to precompile two libraries. Note that this may take some time.

```
cd /opt/Eiffel62
./make_install
```

Answer 'y' to both questions: this will build the precompiled libraries. Note: if you get a linker error about "-lxtst not found", be sure you have the dev version of libxtst installed [debian: package libxtst-dev].

To hand in

There is nothing to hand in.

3 Your first Traffic program

In this task you will download Traffic and write your first feature calls. You need to have EiffelStudio installed (see task 2). Note to the repeating students: Traffic does **NOT** require EiffelMedia any more.

Todo

1. Download Traffic from <http://traffic.origo.ethz.ch/download> and unzip it to a folder of your choice (it's recommended to use a path without any spaces). Make sure that you download the release traffic_ev.3.3-1079.zip!
2. Figure 3 shows the directory structure of Traffic. The top-level directory `library` contains the core of Traffic: Eiffel class files that model the transportation system in a city and support its visualization. The other top-level directory `example` contains the existing Traffic projects such as `city_vision2` (an application that allows testing most of the functionality of Traffic), and the applications that implement the examples of the book Touch of Class (e.g. `02_objects` implementing the Preview example found in Chapter 2 of Touch of Class). Note that the implementation of the examples found in Traffic may differ slightly from the written text in Touch of Class. The section "Supported Examples from Touch of Class and Traffic 3.3" under <http://traffic.origo.ethz.ch/wiki/doc> provides more information.
3. Start EiffelStudio. **For Linux users:** always start ES from a console! Don't use any shortcuts on your desktop, or the "Run Command" menu or the like. Type `estudio` in a console.
4. EiffelStudio will show the dialog of figure 4. If this does not happen automatically, go to `File > Open project` and it should appear.
5. Click on `Add project` and choose the file found under the directory where you unpacked Traffic, at `example/02_objects/objects.ecf`. Click `Open`. This will compile the test application.
6. Launch the application (see figure 5).
7. Open the class `PREVIEW`. Without changing anything, launch the program. Click on `Run example`. After some loading time you will see a map. Note that the loading will take quite long this first time, but will be much faster the next time you do it. Close the application by clicking on the stop button (the red square) in EiffelStudio.

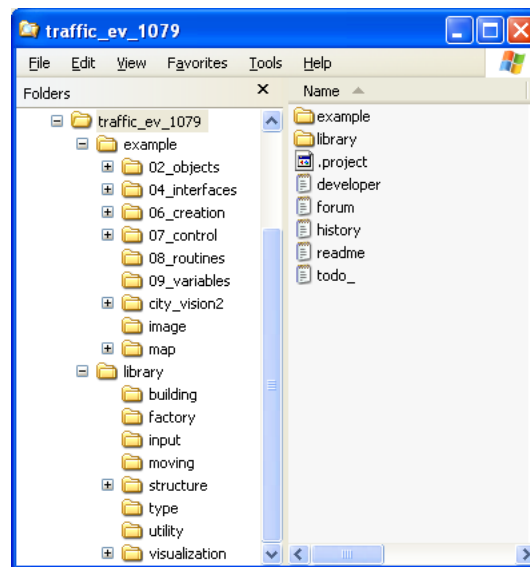


Figure 3: Directory structure of Traffic.

8. In the feature *explore*, between the **do** and the **end** and after *Paris.display*, fill in the following text:

```
1  Lowvre.spotlight  
   Line8.highlight  
3  Route1.animate  
   Console.show (Route1.origin)
```

9. Recompile the project (by clicking on the compile button in EiffelStudio) and launch it again. Click again on the *Run example* button and you will see the above feature calls executing.

To hand in

There is nothing to hand in.

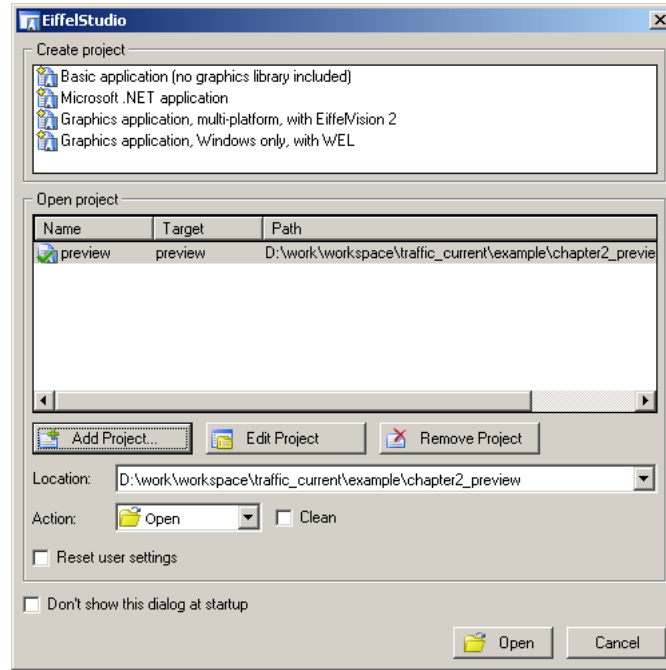


Figure 4: Open an Eiffel project

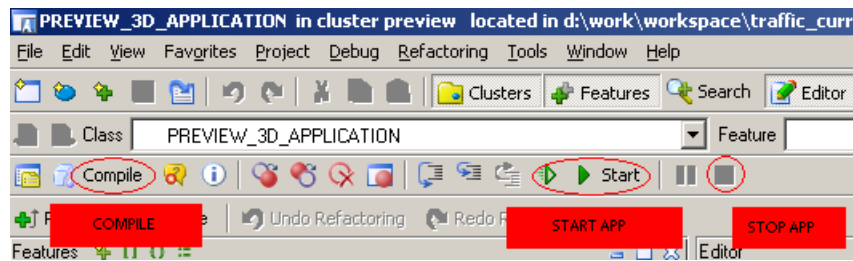


Figure 5: Compiling a project, starting and stopping an application