Userguide Caesar Development Tools

Software Technology Group
Version 0.1

Jochen Unger Daniel Zwicker

September 22, 2004

Contents

1 Introduction **3**

2 Getting Started with CJDT **4**
2.1 CJDT Highlights: 4

3 CJDT Installation **7**
3.1 Clean Install 7
3.1.1 7
3.1.2 8
3.2 Updating an Existing Installation 8
3.3 Is Everything OK? 9

4 Features **10**
4.1 Creating an CaesarJ Project 10
4.2 Creating a new CaesarJ project 10
4.3 Adding a Class to Your Project 12
4.4 Adding a New Aspect to Your Project 14
4.5 Running an Caesar Program 15
4.6 Debugging Caesar Programs 16

5 Propertie and Shortcuts **19**

6 Using the Visualiser **21**

1 Introduction

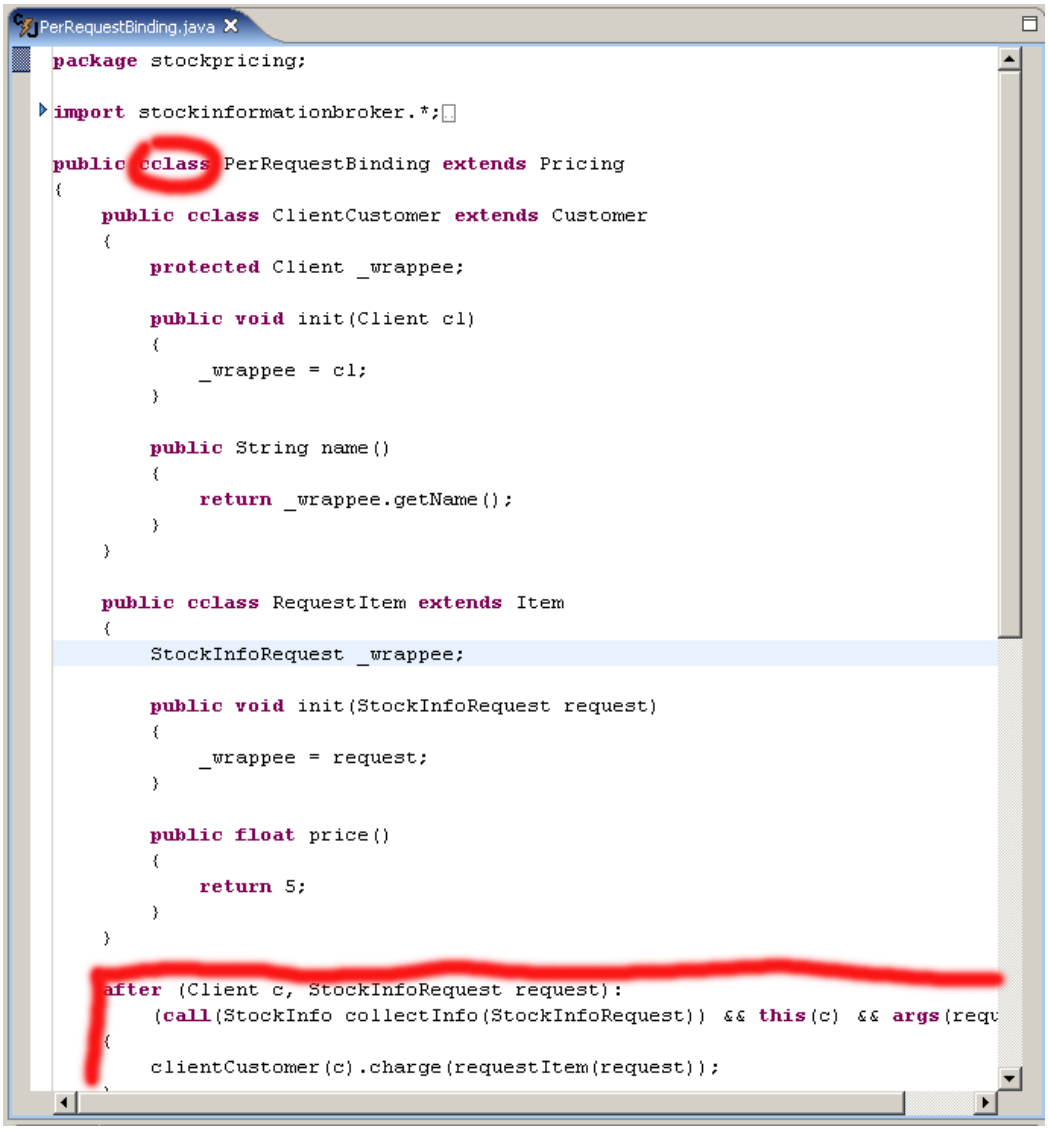
Here the introduction!

2 Getting Started with CJDT

This document describes how to get started with the CaesarJ Development Tools in Eclipse. CJDT provides a rich set of features for working with Caesar programs inside Eclipse.

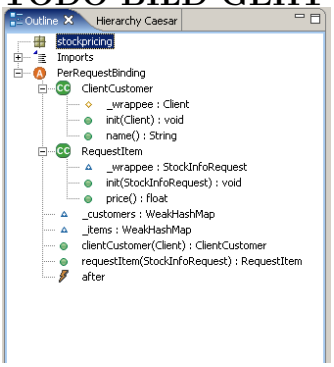
2.1 CJDT Highlights:

- Editor support with keyword highlighting

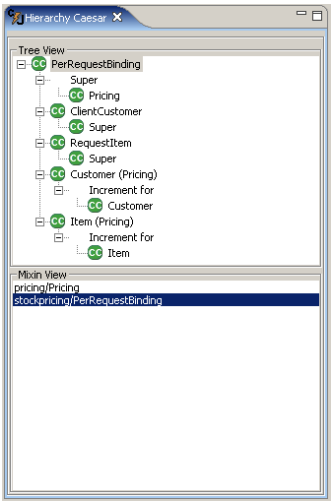


- Outline view showing structural members and crosscutting relationships (e.g. from an advice declaration to the places it advises).

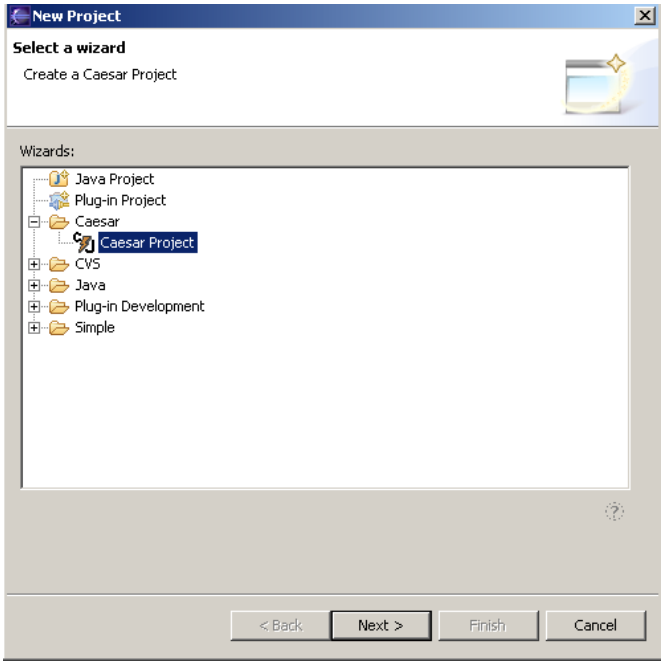
TODO BILD GEHT NICHT RICHTIG ASPECTE FEHLEN



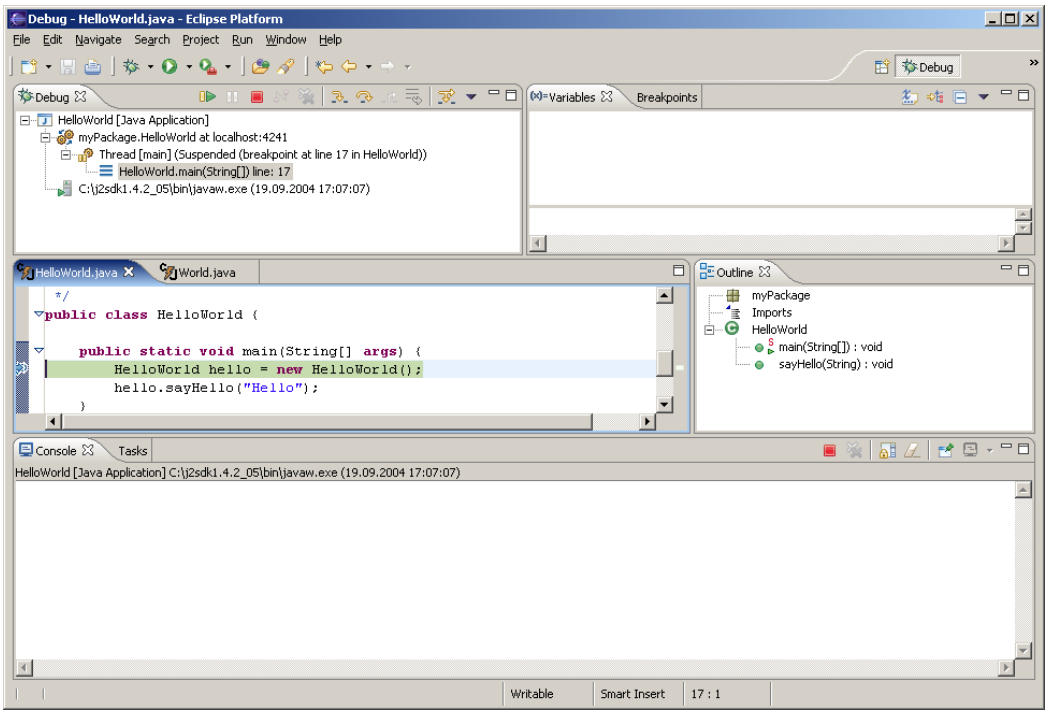
- Caesar-Hierarchy



- New Caesar Project wizard



- Debugging support



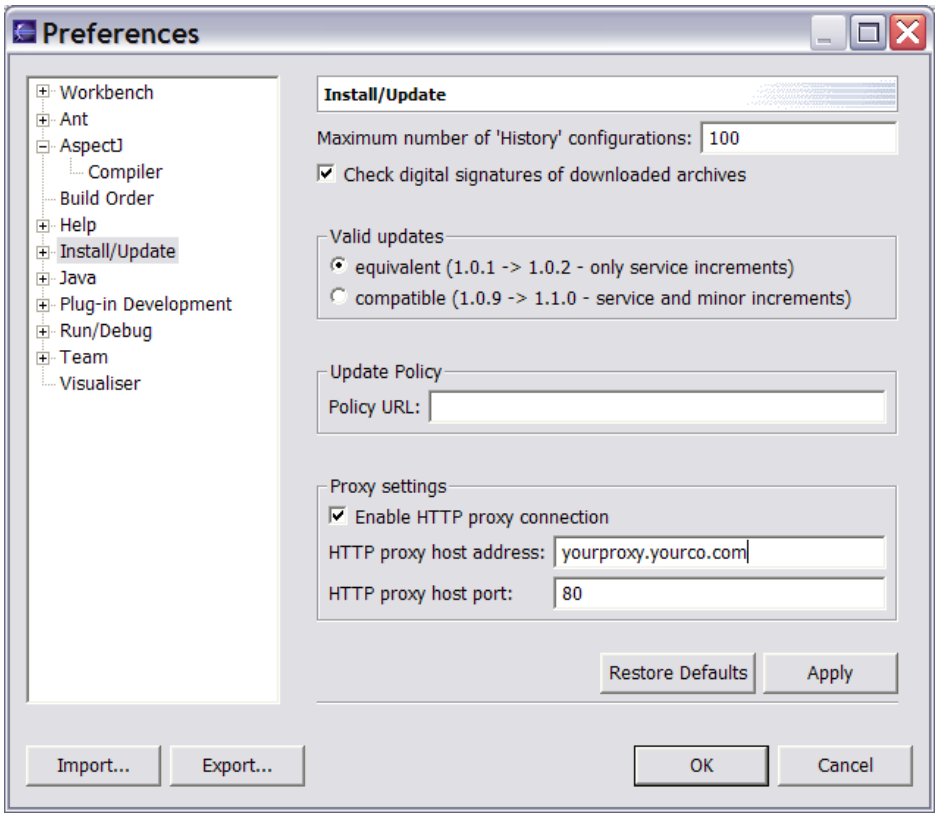
3 CJDT Installation

3.1 Clean Install

CJDT is installed using the Eclipse Update Manager. We recommend you use Eclipse 3.0.

3.1.1

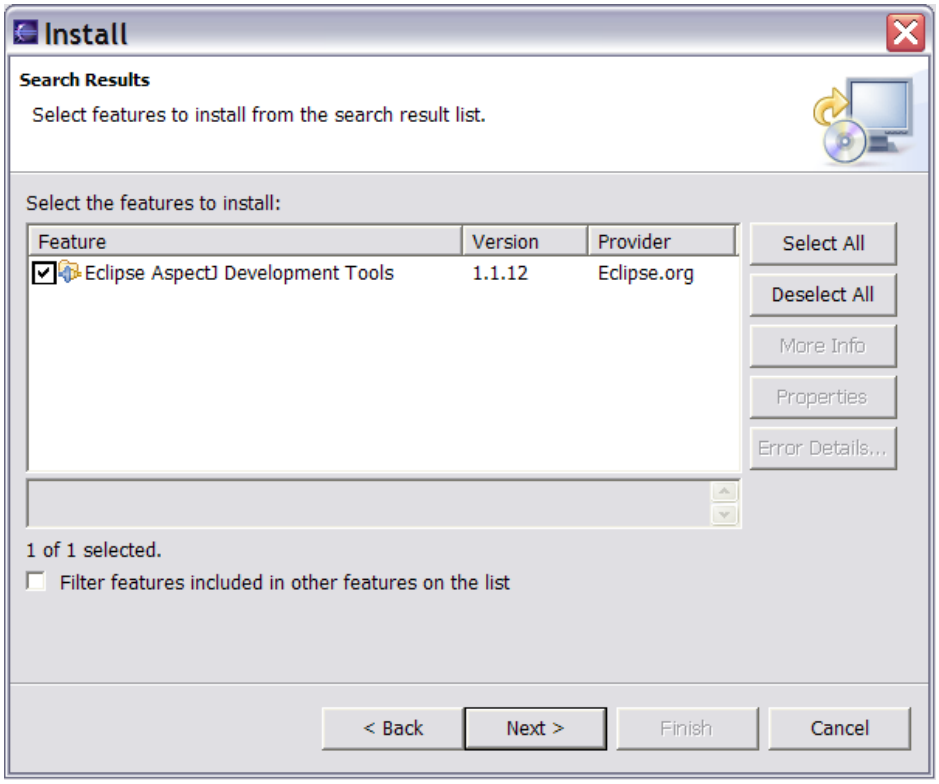
If you need to use a proxy server to access the internet, the first thing to do is configure the proxy details so that the update manager can contact the CJDT Update Site. From the Window menu select preferences, and then the Install/Update tab.



3.1.2

Create an update site bookmark for the CJDT update site, and start the install procedure.

In Eclipse 3.0: From the help menu, select Software Updates -> Find and Install... . Select "Search for new features to install" and click next. Click "Add Update Site" Enter the name "AJDT update site"
and the URL: ***http://download.eclipse.org/technology/ajdt/30/update***.
Click ok.
Fully expand the CJDT Update Site node that appears, and select "'CaesarJ"'.
Click next. Select "Eclipse CaesarJ Development Tools 0.X"



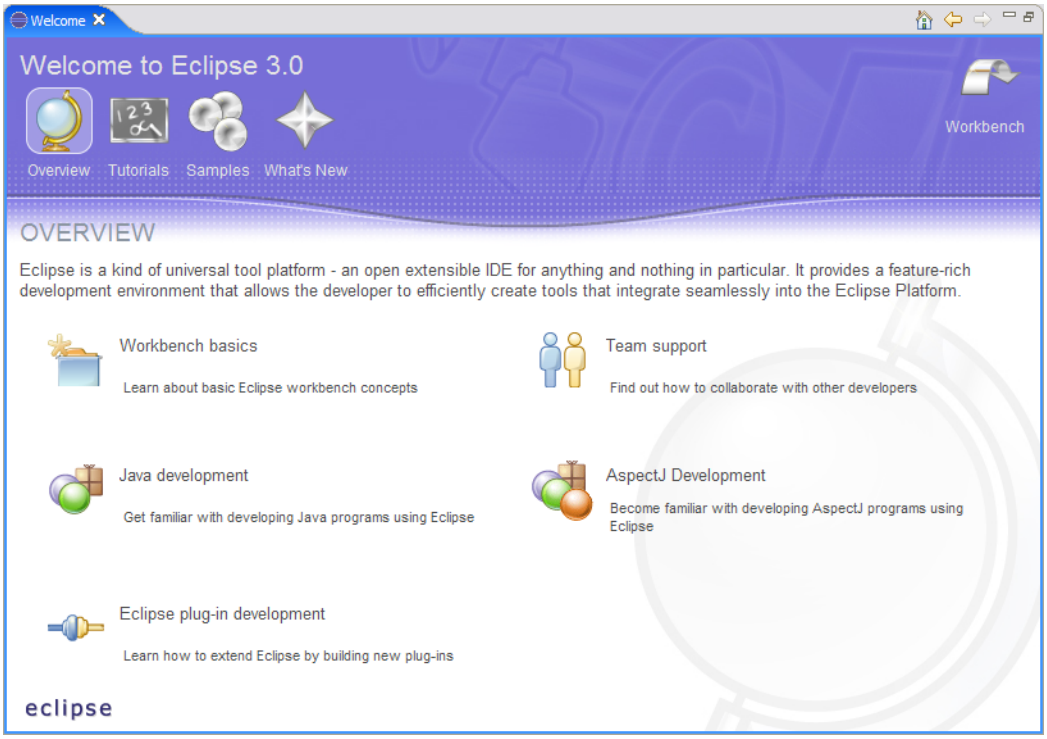
Select Next, accept the license agreement, and proceed to the installation.

3.2 Updating an Existing Installation

Proceed as for a clean install, except that the CJDT Update Site bookmark should already exist – so all you need to do is expand it and go. If the version you have installed is the same as the version on the update site (or more recent even), then you will not be presented with any install options.

3.3 Is Everything OK?

In Eclipse 3.0 the CJDT welcome pages have been integrated into the Eclipse Platform welcome pages. You can view these by selecting "Welcome" from the help menu and then following the Overview, Tutorials, Samples or the What's New links.



4 Features

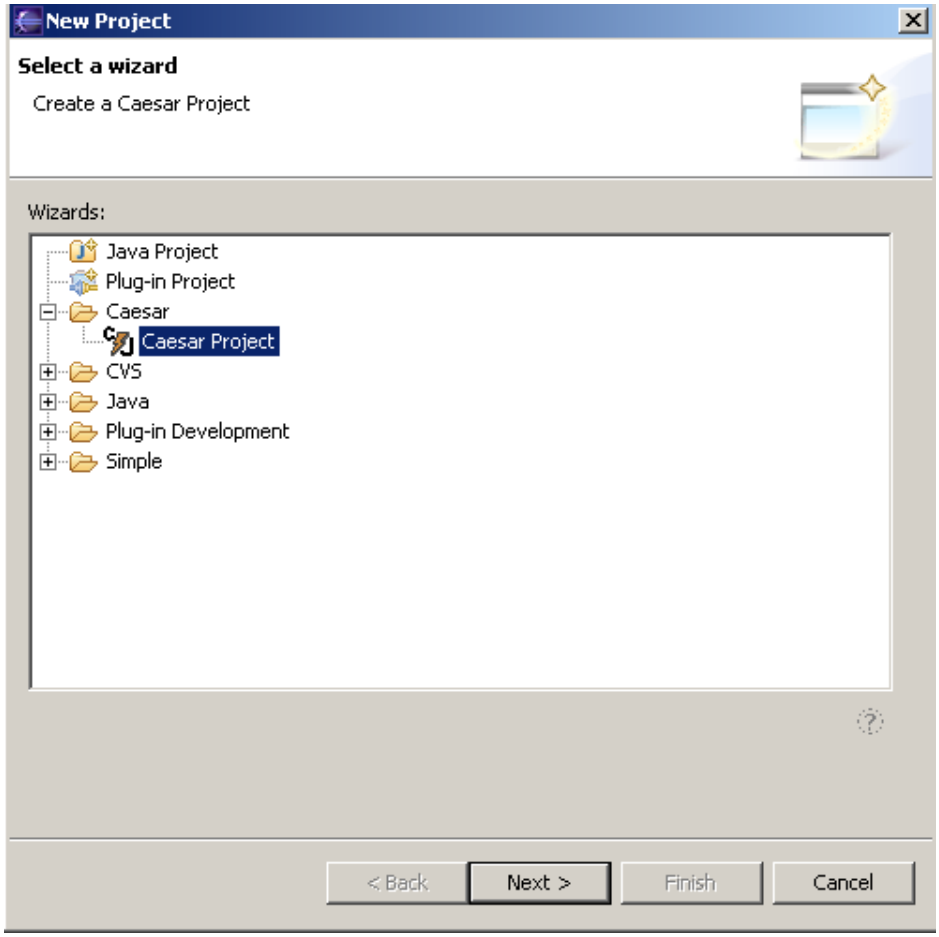
The following section describes the extending features of the CaesarJ Plugin.

4.1 Creating an CaesarJ Project

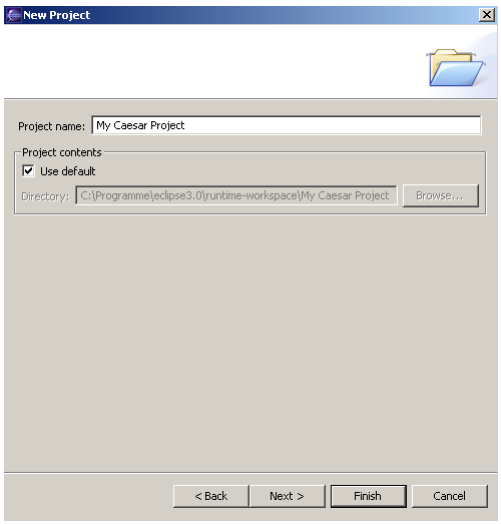
4.2 Creating a new CaesarJ project

From the File menu select new... . If "Caesar Project" appears in the list, select it.

If it doesn't, this is probably the first time you've used the plugin - select "Other" and then "Caesar" and "Caesar Project".

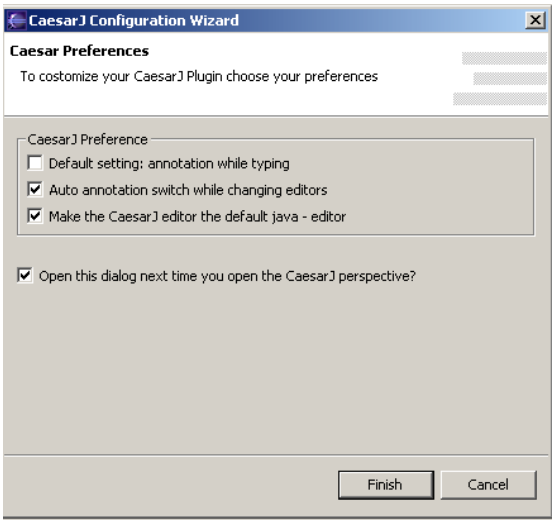


The new project wizard appears:



This wizard has identical behaviour to the new Java project wizard (except of course that it creates a project with the Caesar nature).
When you click "Finish" on the new project wizard, your project will be created.

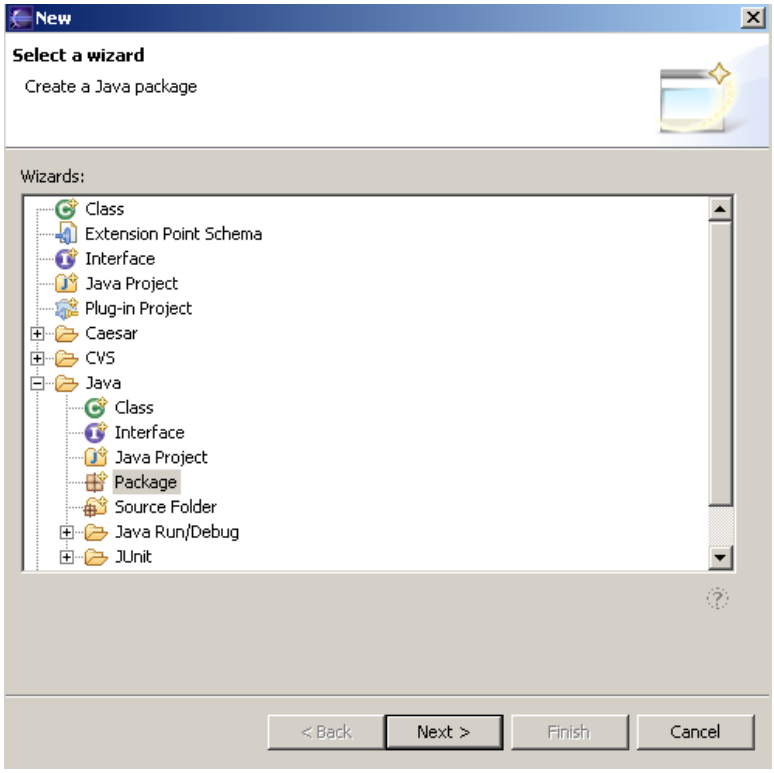
Now you propperly want to open the "Caesar Perspective". Select Window -> Open Perspective -> Others -> CaesarJDT Perspective. If this is the first time you've used CJDTP, you will see the following dialog pop-up:



This dialog configures some Eclipse settings that will make your life much easier when working with CaesarJ projects. Leave everything selected and click "Finish".

4.3 Adding a Class to Your Project

First, creat a package for your class. Select the project you created in the package explorer, right click, and select "New" -> "Package" -> "Other" from the context menu. You have to look for "Package" in the "Java" folder.



Name the package "myPackage" then click finish.

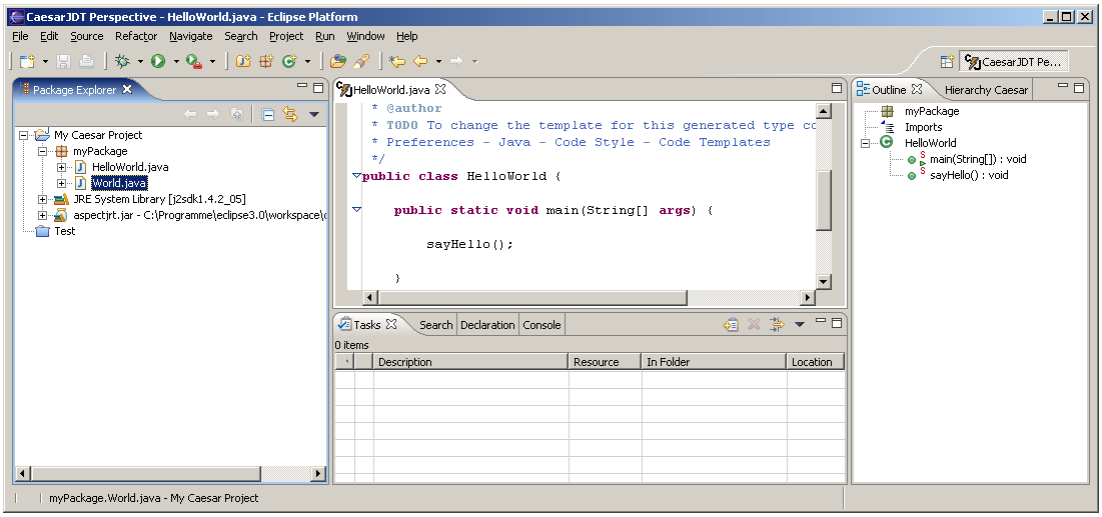
Select the package you just created, and from the context menu select "New" -> "Class". Name the class "HelloWorld" and select the option to let Eclipse create a new main method for you. Click "Finish".

Edit the text in the editor so that it looks something like this:

```
/*
 * Created on 19.09.2004
 *
 * TODO To change the template for this generated file go to
 * Window - Preferences - Java - Code Style - Code Templates
 */
package myPackage;

/**
 * @author
 * TODO To change the template for this generated type comment go to Window -
 * Preferences - Java - Code Style - Code Templates
 */
public class HelloWorld {

    public static void main(String[] args) {
        HelloWorld hello = new HelloWorld();
        hello.sayHello("Hello");
    }

    public void sayHello(String arg) {
        System.out.println(arg);
    }
}
```

Save the file.

Notice that unlike in a Java project, there was no eager parsing of the buffer as you typed (the outline view didn't update). Your Eclipse workbench should be looking something like this:

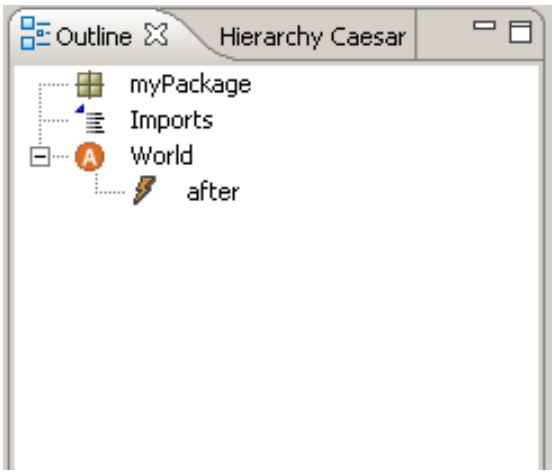


4.4 Adding a New Aspect to Your Project

Create a new Class and name it "'**World**"'. Edit the buffer so it looks as follows and then save it:

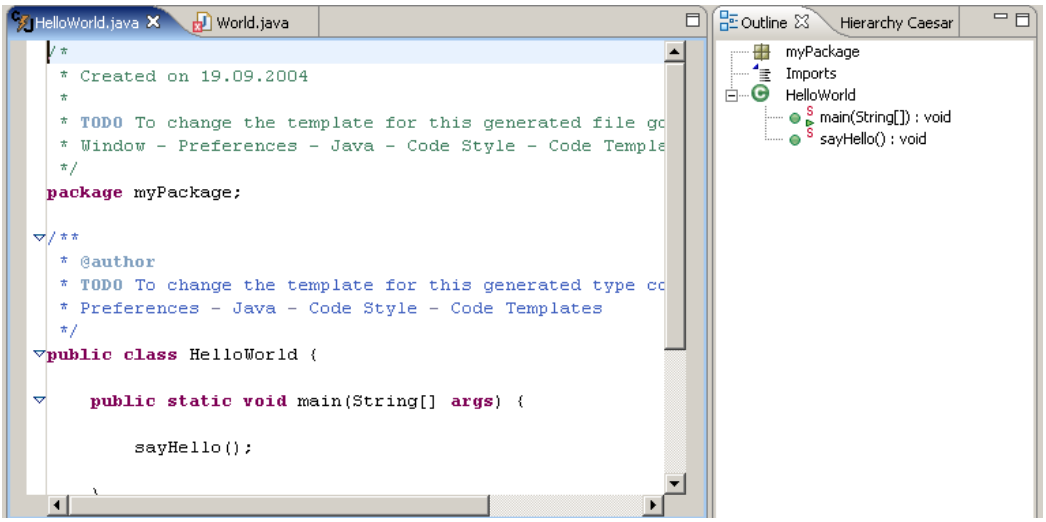
```
Hier kommt dann der Code hin, der in aspect.png abgelegt ist.
private RootNode buildTreeModel(String path)
{
try {
      //TODO BUG: bei Zwei geöffneten Projekten wird nicht die Auswahl
      //          des aktuellen Projekts genommen, sondern das, was als
//          letztes gebuildet wurde!!!
ProjectProperties projectProperties = new ProjectProperties(Builder.getLastBuild
log.debug(projectProperties.getProjectLocation()+projectProperties.getOutputPath
globalPathForInformationAdapter = projectProperties.getProjectLocation()+project
} catch(Exception e)
{

}
```

Make a clean Build of the project, and the outline view populates. Expand the "'**after()**"' node.



You can see that this advice is affecting the HelloWorld.sayHello() method. Clicking on the **"'HelloWorld.sayHello()"'** node in the outline takes you to the declaration of HelloWorld.sayHello().

Notice the **"'advice annotation"'** in the editor buffer (highlighted) and that the **"'sayHello"'** method in the outline view shows that it is advised by the World aspect.

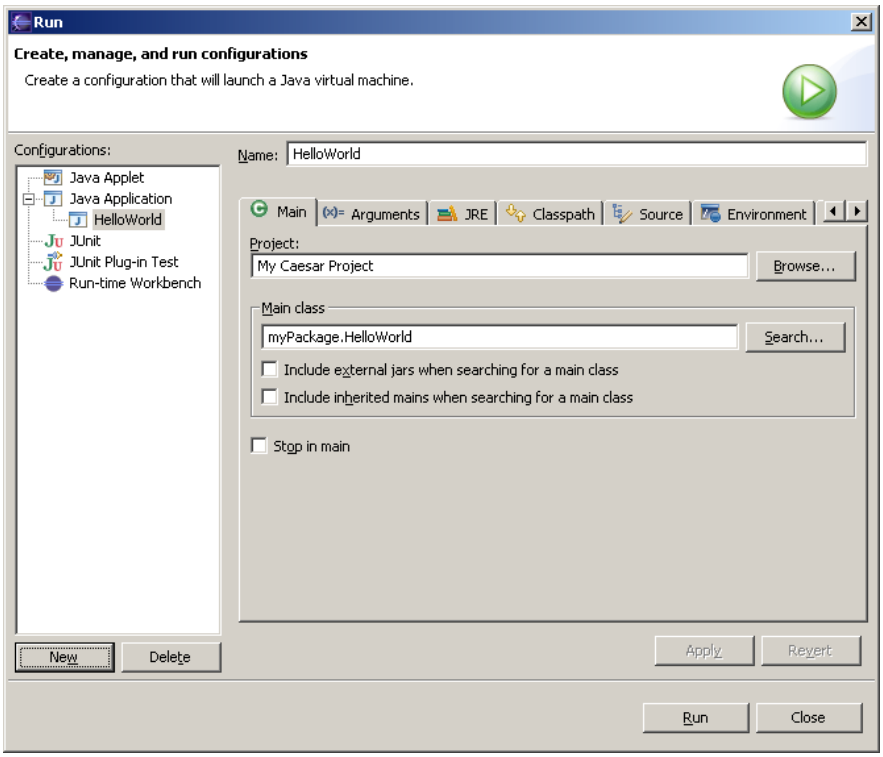


Selecting the **"'World.after()"'** node in the outline view takes you back to the advice declaration. Right-clicking on the advice annotation brings up a context menu that also allows you to navigate to the advice.

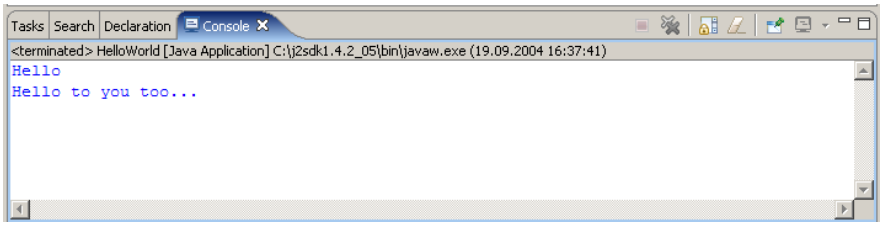
TODO BILD FEHLT KONNTE ICH NICHT SCREENSHOOT MACHEN Weil es nicht geht EDITOR mit ADVICE CONTEXT

4.5 Running an Caesar Program

Select your Caesar project in the Package Explorer. Drop-down the "run" icon on the toolbar and click "Run..."
Select "Java Application" in the left-hand tab and click "New" Name this configuration "HelloWorld" and then click "Search" to find the main class. Select "HelloWorld".



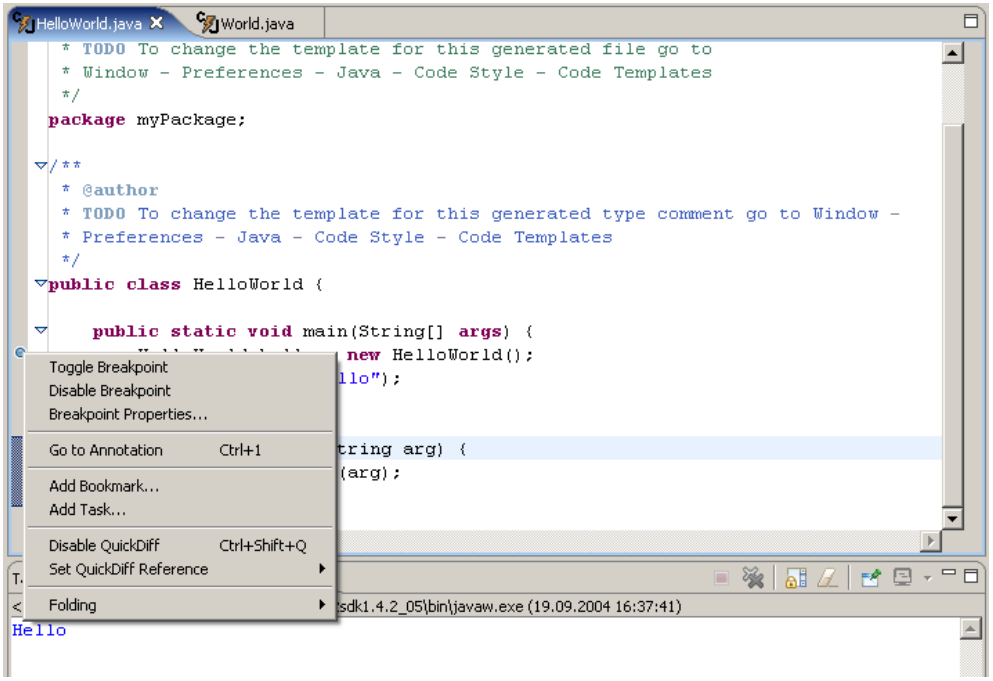
Click "Apply" and then "Run".
You should see the output of the HelloWorld class and the World aspect in the console.



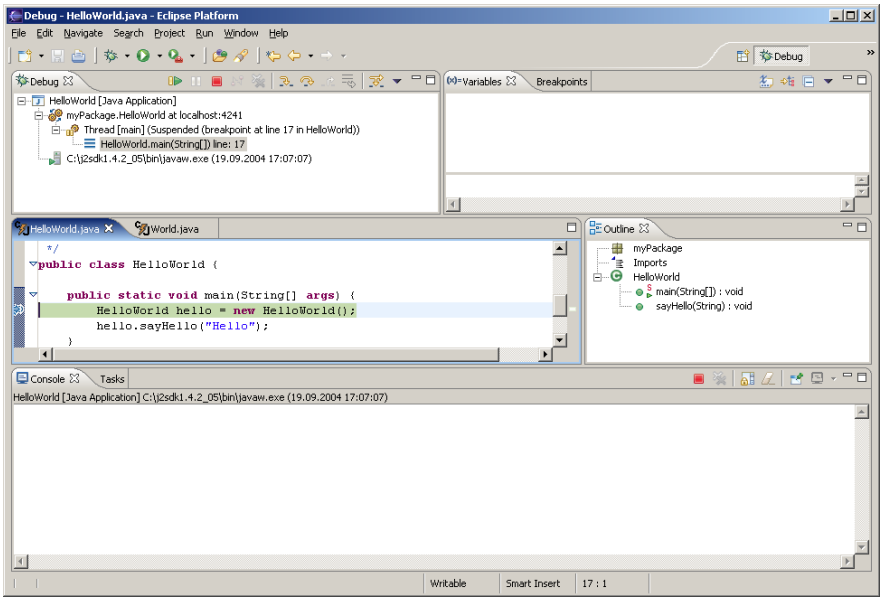
To run this configuration again, just click on the "run" icon on the toolbar.

4.6 Debugging Caesar Programs

You can debug Caesar programs using the normal Java debugger. To set a breakpoint, right-click in the gutter of the editor and choose "Toggle Breakpoint", or simply double-click in the gutter.



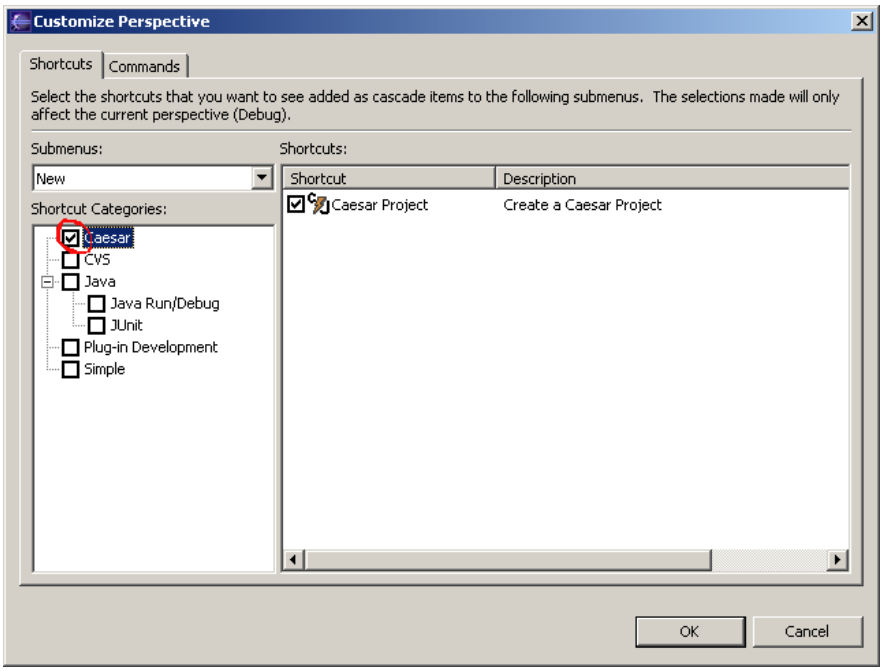
With one or more breakpoints set, you launch the Eclipse debugger in the normal way by clicking on the debug icon in the toolbar.



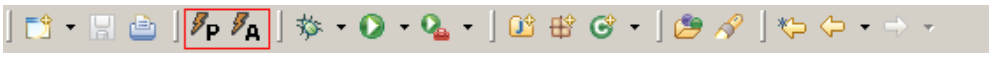
You can use the Java Debug step filters (Window -> Preferences -> Java -> Debug -> Step Filtering) to make this process a little easier. Note: A current limitation is that you cannot step into advices.

5 Propertie and Shortcuts

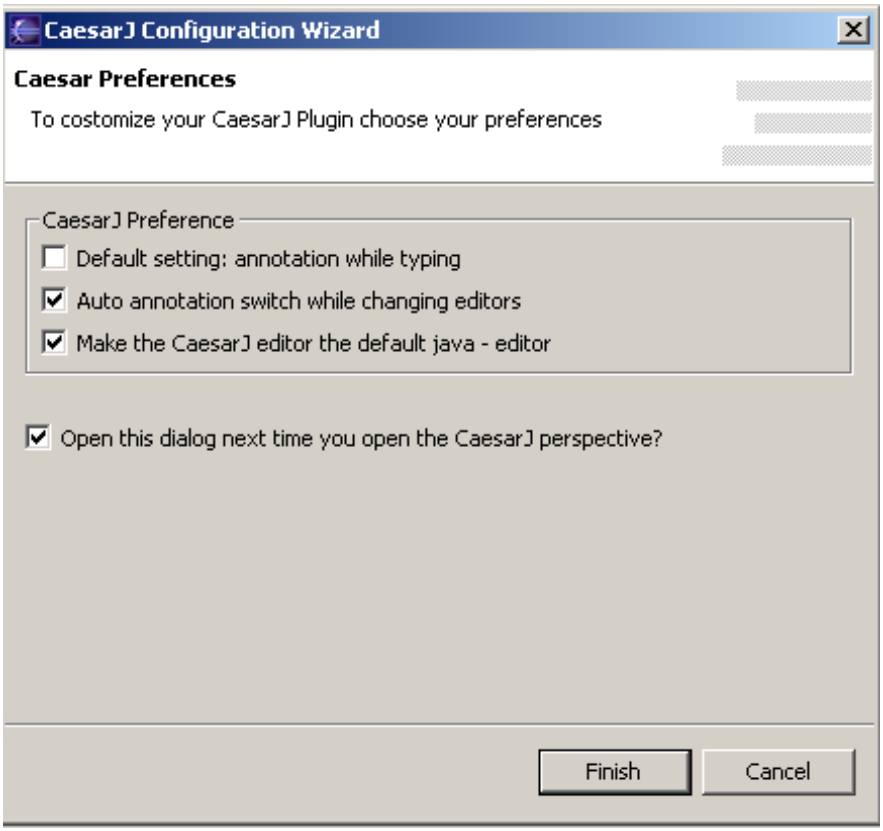
If you have opened the Caesar Perspective, there are some configurations left. Open Window->Customise Perspective... .
Check the Caesar-Checkbox as shown below.



If you have done this two new Buttons will appear in the Toolbar.



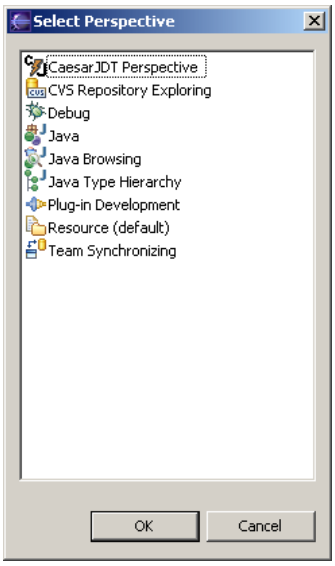
With the "P"-Button the "Caesar-Configuration-Wizard" will be shown.



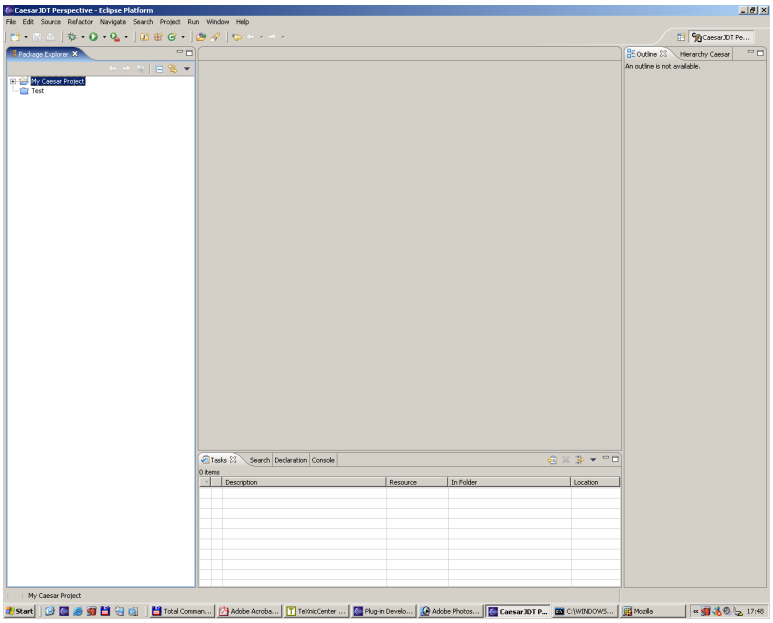
The "A"-Button toggels the "Annotation-While-Typing" option on or off. Even for the Java-Editor!

6 Using the Visualiser

If this is the first time you have used the Visualiser, switch to the visualization perspective by selection Window -> Open Perspective -> Other... and then "CaesarJDT Perspective".



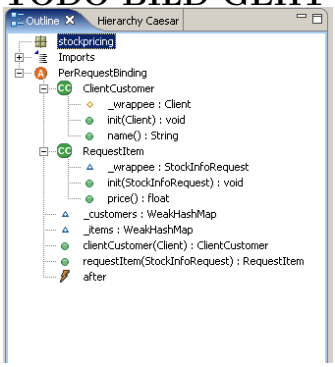
The visualization perspective opens:



You can switch between the Java and Caesar Visualization perspectives using the perspective icons in the top right of the menu bar.

Outline view showing structural members and crosscutting relationships (e.g. from an advice declaration to the places it advises):

TODO BILD GEHT NICHT RICHTIG ASPECTE FEHLEN



A Caesar Hierarchy view showing the Typhierachy and the Linearhierarchy of an Caesar-CClass: