

# ECLIPSE TOMCAT

## In dieser Starthilfe

*Installation des Tomcat Plugins  
Anpassen der Installation  
Das Tomcat HelloWorld (Servlet) Projekt  
Hot Code Replacement*

# Eclipse Tomcat Plugin Starthilfe

## 1.1. Installation

Folgende Software sollten Sie bereits installiert haben:

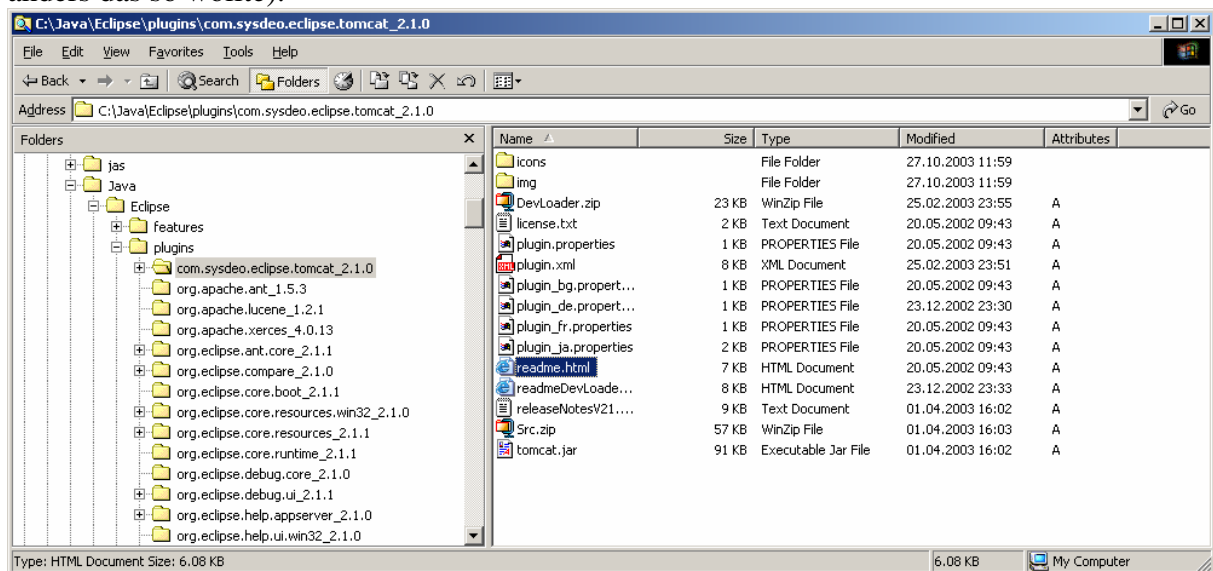
- J2SDK1.4 oder neuer (ich habe 1.4.2\_02)
- Tomcat 4 oder neuer (ich habe 4.1.27)

Falls Sie wissen möchten, wie man Tomcat installiert und konfiguriert, können Sie dies in der entsprechenden Starthilfe nachschauen.

Folgend Software brauchen Sie als Plugin:

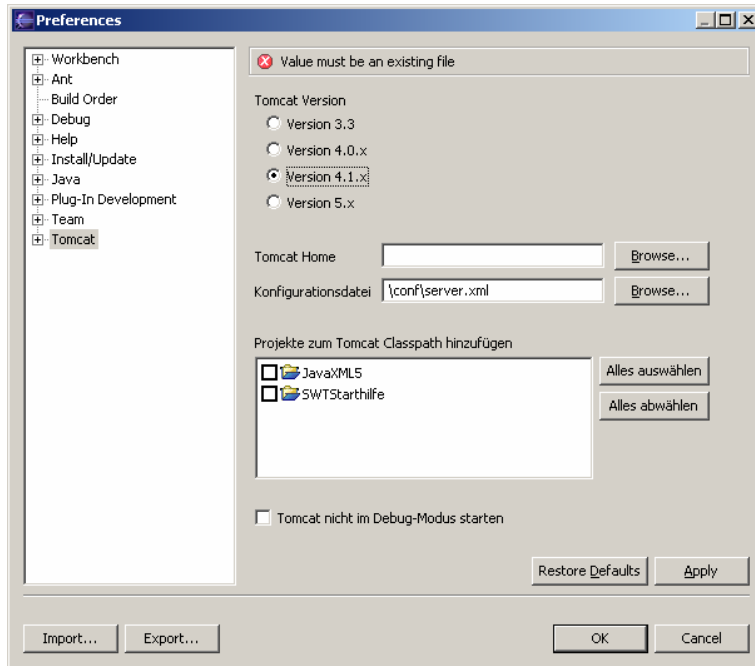
- Sysdeo Tomcat Plugin  
<http://www.sysdeo.com/eclipse/tomcatPlugin.html>

Laden Sie diese Plugin herunter (tomcatPluginV21.zip) und installieren sie es als Plugin, indem Sie es ins Verzeichnis ECLIPSE\_HOME\plugins entzippen. ECLIPSE\_HOME ist typischerweise C:\Eclipse. (bei mir ist es C:\Java\Eclipse, weil jemand anders das so wollte).



# ECLIPSE TOMCAT

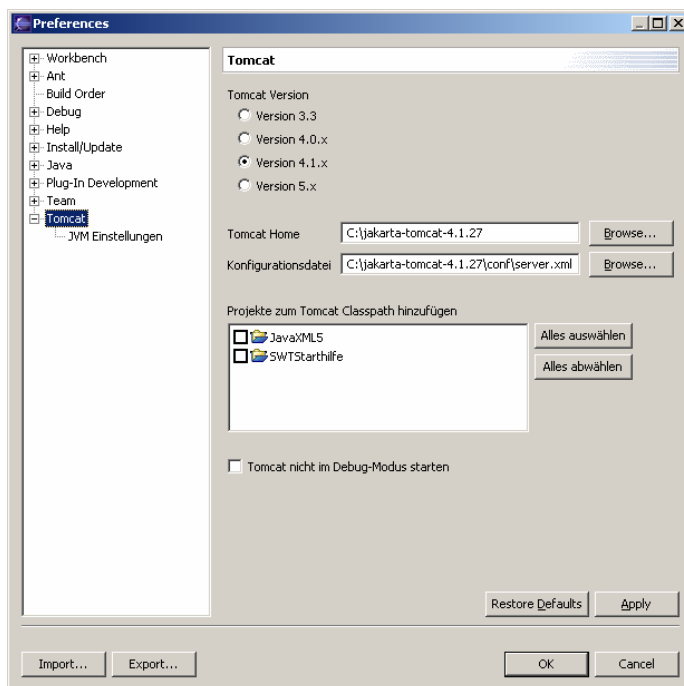
Wenn Sie jetzt Eclipse starten, so sehen Sie unter „Window->Preferences“ folgendes Fenster:



## 1.2. Anpassen der Installation

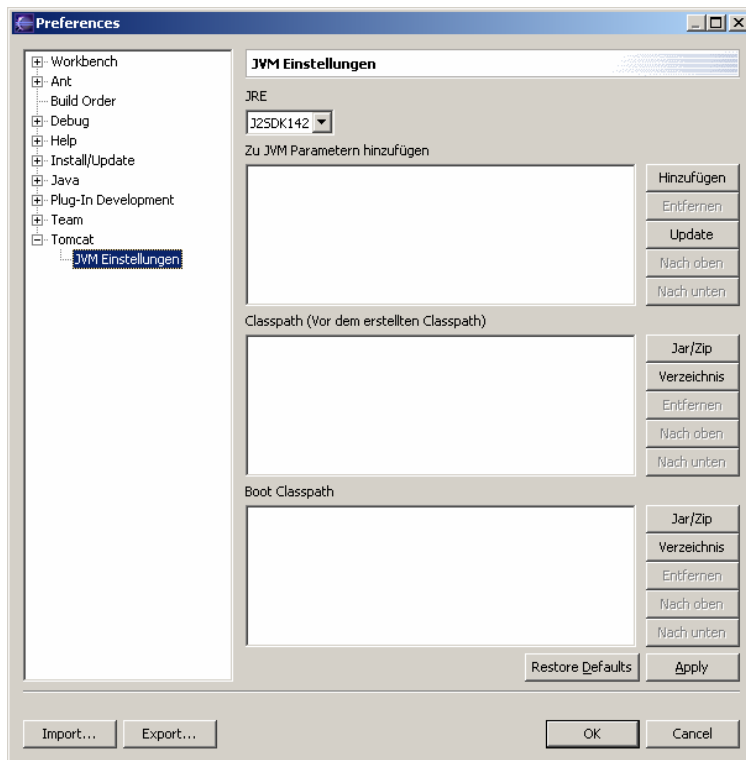
Starten Sie Eclipse und öffnen Sie das „Window->Preferences“ Fenster (wie oben).

- wählen Sie den „Tomcat“ Knoten.
- wählen Sie die Version, die Sie installiert haben (bei mir: „Version 4.1.x“)
- geben Sie an, wo sich Jakarta Tomcat befindet (bei mir: C:\jakarta-tomcat-4.1.27)
- bestätigen Sie Ihre Auswahl mit „Apply“
- verlassen Sie das Auswahlfenster mit „OK“

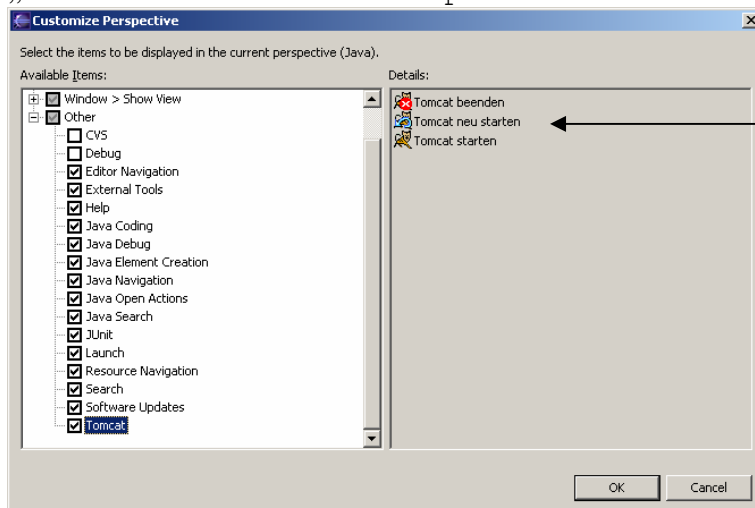


# ECLIPSE TOMCAT

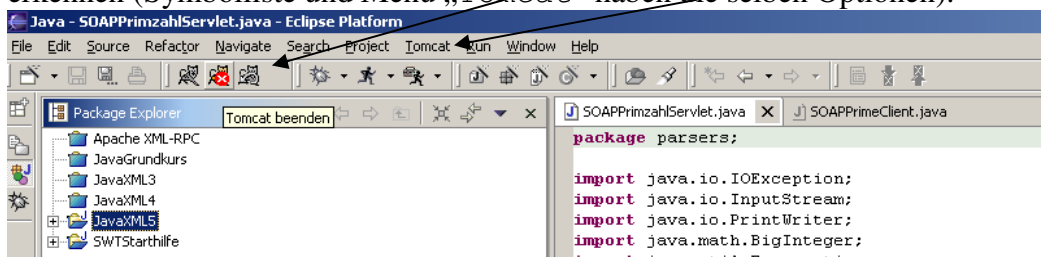
Beim JVM Submenü habe ich das Java Runtime auf 1.4 eingestellt:



Nun können Sie in der Eclipse Perspektive Tomcat sichtbar machen unter „Window->Customize Perspective“



Aktivieren Sie unter „Others->Tomcat“ und bestätigen Sie Ihre Auswahl mit „OK“. Jetzt können Sie in der angepassten Perspektive verschiedene Tomcat Symbole und Optionen erkennen (Symbolleiste und Menü „Tomcat“ haben die selben Optionen):

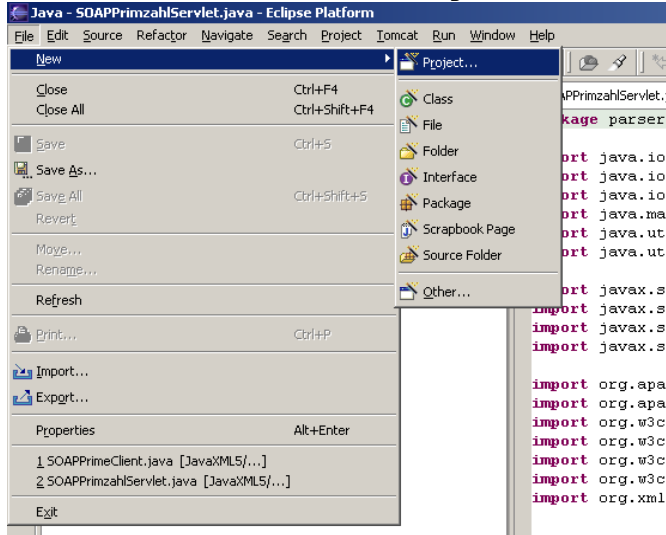


# ECLIPSE TOMCAT

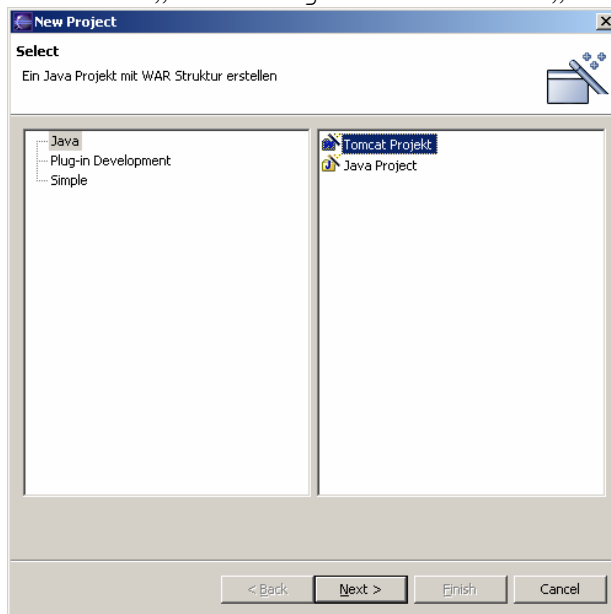
## 1.3. Tomcat Hello World Projekt

Wir illustrieren das Zusammenspiel zwischen Eclipse und Tomcat mithilfe eines einfachen Hello World Servlets. Anschliessend schauen wir uns an, wie JSP's entwickelt und getestet werden können.

### 1. Wählen Sie „File->New->Project“

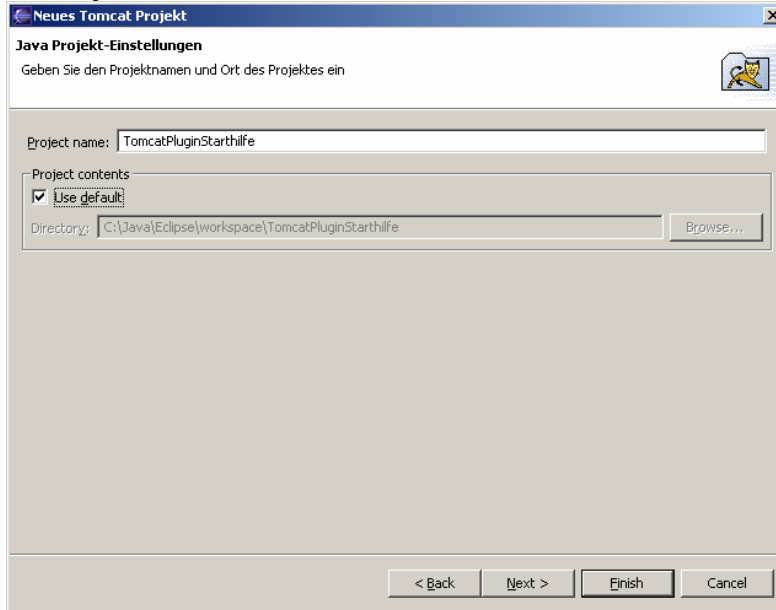


### 2. Im Fenster „New Project“ wählen Sie „Java->Tomcat Project“



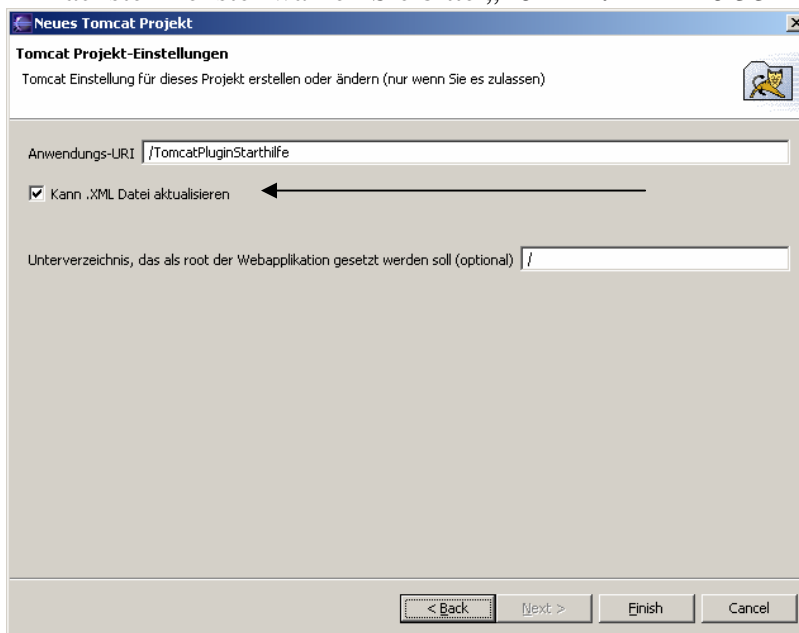
# ECLIPSE TOMCAT

3. als Projektnamen wählen wir „TomcatPluginStarhilfe“



Als „Project Content“ verwenden wir die Default Einstellung, also das Verzeichnis „%ECLIPSE\_HOME%\workspace\TomcatPluginStarhilfe“

4. im nächsten Fenster wählen Sie bitte „Kann .XML Datei aktualisieren“



Dabei handelt es sich um die Datei `server.xml` im Tomcat `conf` Verzeichnis. Indem Sie diese Option aktivieren, brauchen Sie jeweils keine projektspezifischen Anpassungen an die Server-Konfiguration manuell durchzuführen: der passende Kontext (Context) wird automatisch eingetragen und ein Link in den Workspace hergestellt.

Die Anwendungs-URI entspricht der Relativadresse auf dem Server (siehe unten)

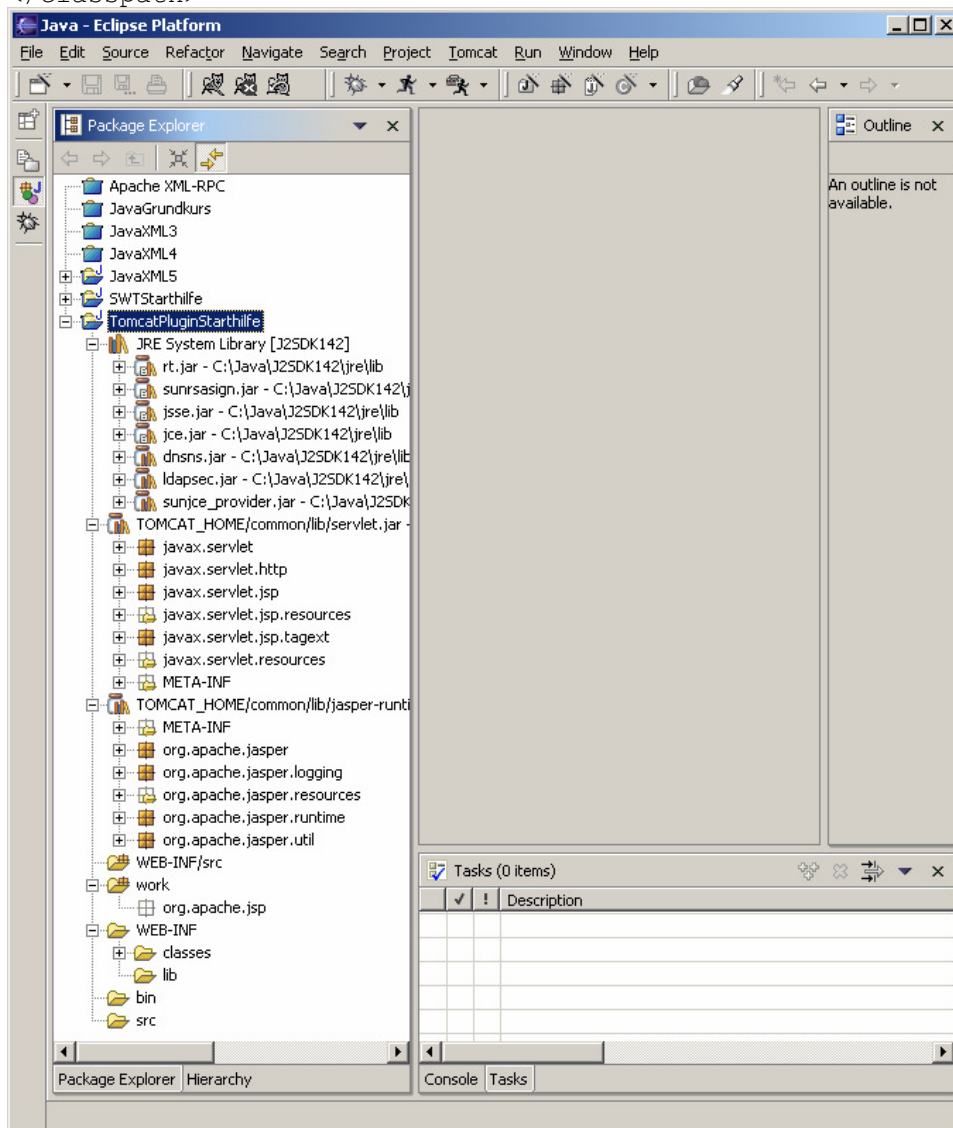
5. klicken Sie „Finish“. Das Projekt ist gestartet!

# ECLIPSE TOMCAT

## 6. Schauen wir uns die Struktur des Projekts an:

In der Datei `.classpath` finden wir die oben durchgeführten Einstellungen wieder:

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
  <classpathentry kind="con"
    path="org.eclipse.jdt.launching.JRE_CONTAINER"/>
  <classpathentry kind="var"
    path="TOMCAT_HOME/common/lib/servlet.jar"/>
  <classpathentry kind="var" path="TOMCAT_HOME/common/lib/jasper-
    runtime.jar"/>
  <classpathentry kind="var" path="TOMCAT_HOME"/>
  <classpathentry kind="src" path="WEB-INF/src"/>
  <classpathentry kind="src" path="work"/>
  <classpathentry kind="output" path="WEB-INF/classes"/>
</classpath>
```

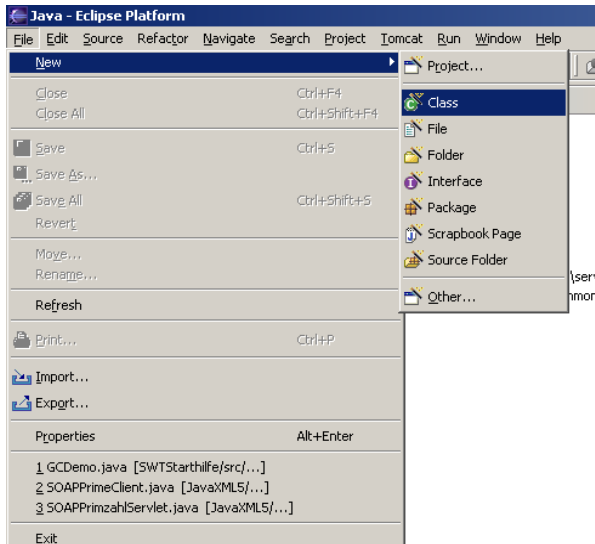


Die Verzeichnisse "bin" und "src" werden nicht weiter benötigt.

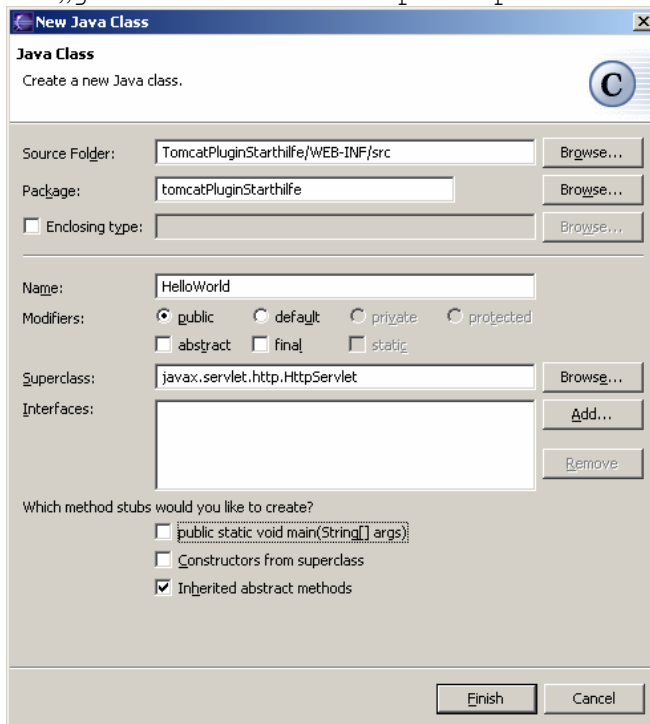
# ECLIPSE TOMCAT

- Jetzt schreiben wir das Hello World Servlet:  
die Datei muss im Verzeichnis  
„TomcatPluginStarthilfe/WEB-INF/src“ angelegt werden.  
Selektieren Sie dieses Verzeichnis.

- Jetzt legen wir die Klasse an:  
“File->New->Class“



- Geben Sie einen Package und einen Klassennamen; dabei müssen wir unsere Klasse aus „javax.servlet.http.HttpServlet“ ableiten.

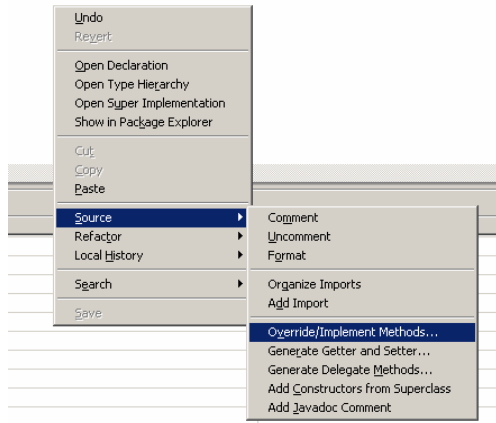


Die „public static void main(String[] args)“ Methode brauchen wir nicht, obschon ich diese oft stehen lasse, um das Servlet als Applikation testen zu können.

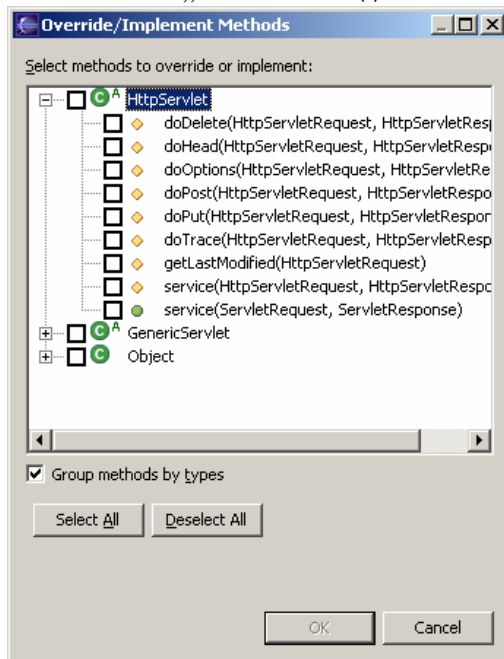
Klicken Sie auf „Finish“, um die Klasse zu generieren.

# ECLIPSE TOMCAT

10. Ergänzen Sie oder überschreiben Sie nun die `doGet ()` Methode Ihres `HelloWorld` Servlets. Wählen Sie dazu im Source Code mit der rechten Maustaste das Popup Menü „Source->Organize/Implement Methods...“.



11. Es öffnet sich ein Popup Fenster und Sie können die gewünschten Methoden (in unserem Fall „...doGet ()“ auswählen (sofern Sie diese finden!).



## Hinweis:

falls Sie eine Klasse mit vielen Methoden haben, dann können Sie in der „Outline View“ die gewünschte Methode leicht finden, da Sie in dieser View die Methoden aufsteigend oder absteigend sortieren können, bzw. Methoden und Attribute selektiv verbergen können.

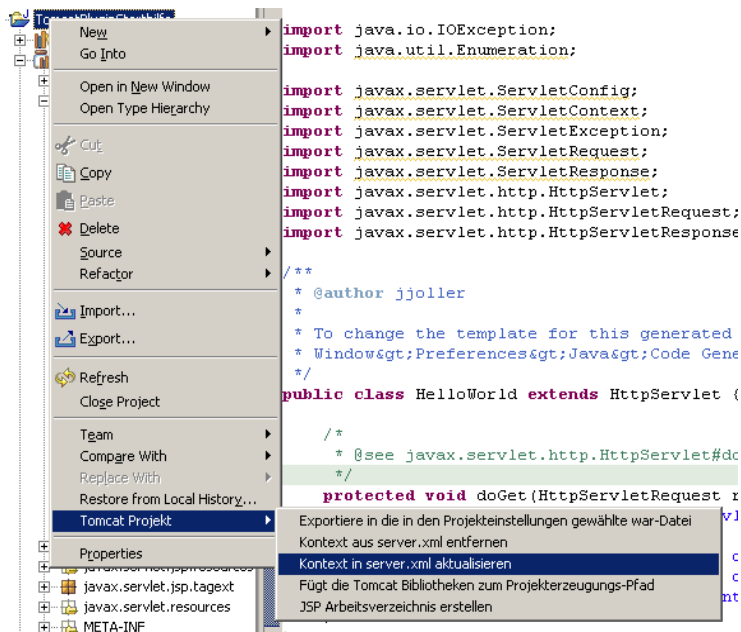
12. Speichern Sie Ihr Servlet!



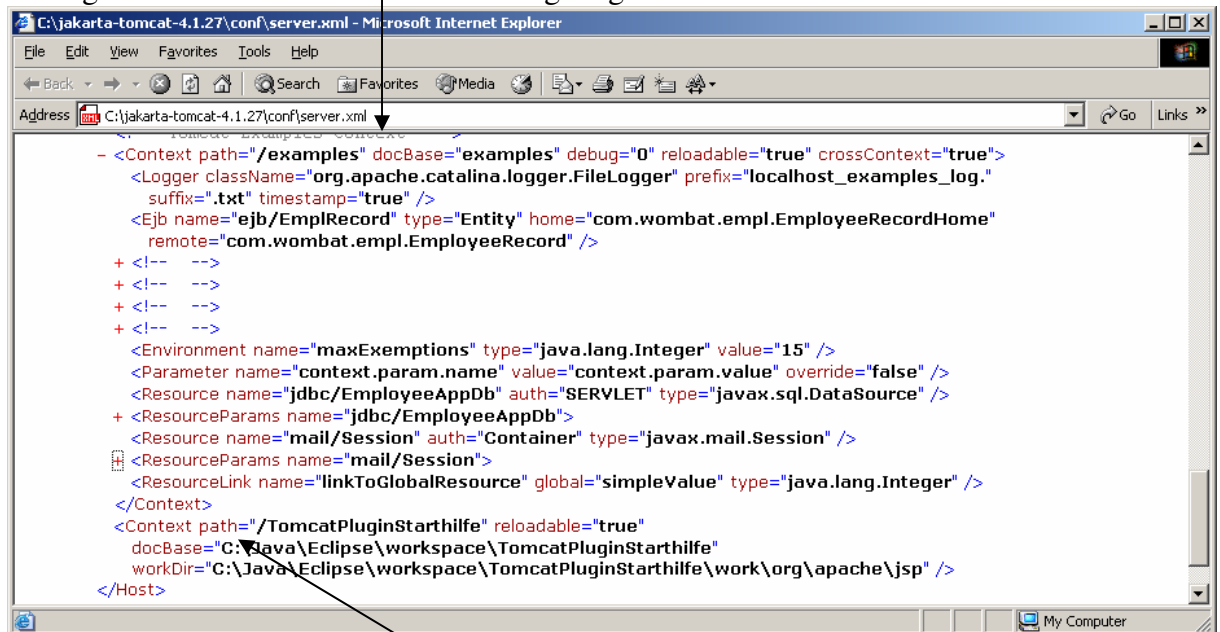
# ECLIPSE TOMCAT

13. Jetzt folgt der eigentliche Tomcat Teil:

öffnen Sie das Kontextmenü Ihres Projekts d.h. selektieren Sie das Projekt und klicken Sie auf die rechte Maustaste. Im Popup Menü, selektieren Sie die „Tomcat“ Option.



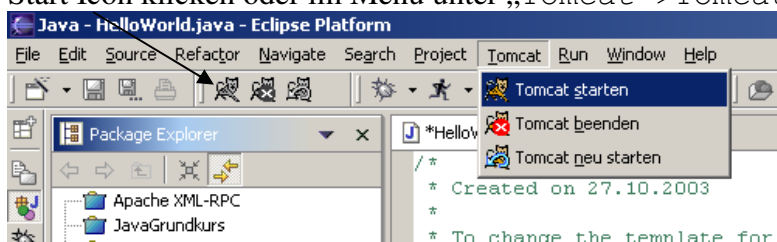
Aktualisieren Sie die Datei `server.xml`. Damit wird in der Tomcat Server Konfiguration ein Link auf Ihr Servlet eingetragen:



Hier sehen Sie auch, was es mit der Pfadangabe auf sich hat.

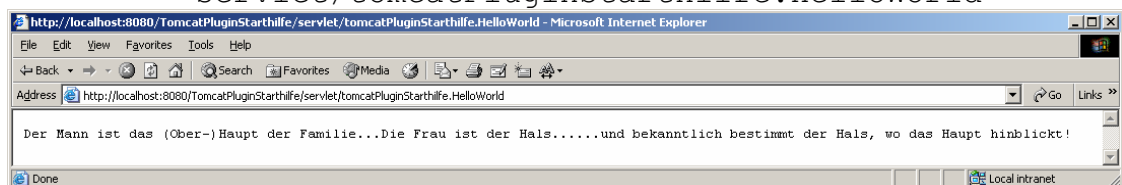
# ECLIPSE TOMCAT

14. Jetzt wird's spannend. Starten Sie den Tomcat Server, indem Sie entweder auf das Start Icon klicken oder im Menü unter „Tomcat->Tomcat starten“:



15. Unser Servlet befindet sich an der URL

“<http://localhost:8080/TomcatPluginStarthilfe/servlet/tomcatPluginStarthilfe.HelloWorld>“



Die Zusammensetzung der URL ergibt sich wie folgt:

- Protokoll und Host-Name : `http://localhost:8080`
- URI unserer Anwendung : `/TomcatPluginStarthilfe`
- Servlet-URI : `/servlet/tomcatPluginStarthilfe.HelloWorld`  
(`/servlet + Packagename.Klasse`)

### Hinweis:

falls Sie jetzt eine Fehlermeldung statt der obigen Ausgabe erhalten haben, dann haben Sie eventuell Tomcat zwar installiert, aber das Invoker-Servlet nicht aktiviert. Wie Sie dieses Problem lösen können, finden Sie in der Tomcat Starthilfe.

### Hinweis:

falls Sie ein Crack sind und einen Deployment Descriptor „web.xml“ im WEB-INF Verzeichnis erstellt haben, wird Tomcat diesen zusätzlich berücksichtigen.

### Hinweis:

falls Sie Ihr Programm verändern und abspeichern, erkennt dies Tomcat und meldet, dass eine neue Version erkannt und geladen wurde, wie im folgenden Protokoll ersichtlich:

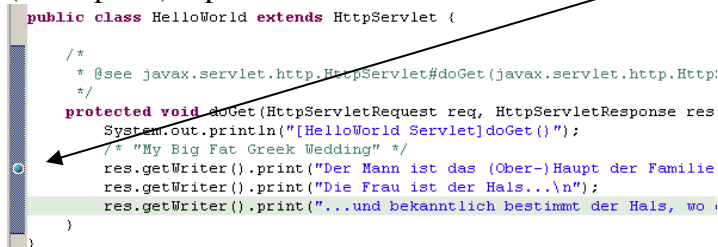
```
INFO: Initializing Coyote HTTP/1.1 on port 8080
Starting service Tomcat-Standalone Apache Tomcat/4.1.27
27.10.2003 15:53:00 org.apache.struts.util.PropertyMessageResources <init>
INFO: Initializing, config='org.apache.struts.util.LocalStrings', returnNull=true
...
INFO: JK2: ajp13 listening on /0.0.0.0:8009
27.10.2003 15:53:05 org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=20/130 config=C:\jakarta-tomcat-4.1.27\conf\jk2.properties
[HelloWorld Servlet]doGet()
WebappClassLoader: Resource '/WEB-INF/classes/TomcatPluginStarthilfe/HelloWorld.class' was modified; Date is now: Mon
Oct 27 15:56:12 CET 2003 Was: Mon Oct 27 15:15:48 CET 2003
```

# ECLIPSE TOMCAT

## 1.4. Hot Code Replacement

Jetzt wollen wir uns dieses Verhalten zunutze machen und „Hot Code Replacement“ testen, also das Ersetzen von Programmcode in einem laufenden Programm.

1. Für das Debuggen Ihres Hello World Programms setzen Sie einen *Breakpoint* in der `doGet()` Methode:  
doppelklicken Sie auf der linken Rand; es sollte ein *blauer Punkt* sichtbar werden (Breakpoint). Speichern Sie Ihr Servlet.

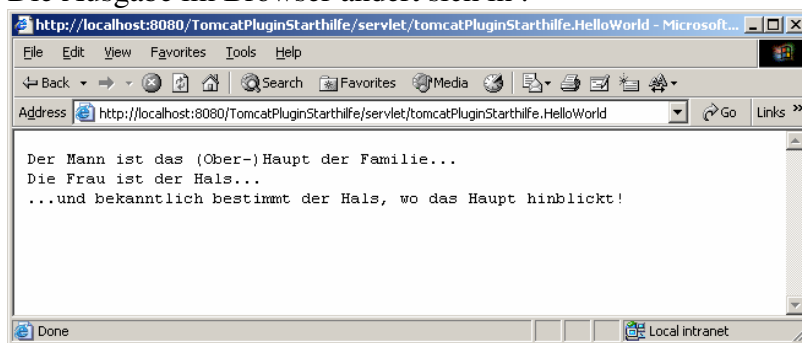


```
public class HelloWorld extends HttpServlet {  
    /*  
     * @see javax.servlet.http.HttpServlet#doGet(javax.servlet.http.Http:  
     */  
    protected void doGet(HttpServletRequest req, HttpServletResponse res  
        System.out.println("[HelloWorld Servlet]doGet()");  
        /* "My Big Fat Greek Wedding" */  
        res.getWriter().print("Der Mann ist das (Ober-)Haupt der Familie  
        res.getWriter().print("Die Frau ist der Hals...\n");  
        res.getWriter().print("...und bekanntlich bestimmt der Hals, wo  
    }  
}
```

2. Wenn Sie jetzt die Servlet URL neu laden (Refresh), startet das Servlet und bleibt beim Breakpoint stehen.

Ändern Sie die Methoden von `...print(...)` in `..println(...)` ...  
Speichern Sie die Änderungen.

Die Ausgabe im Browser ändert sich in :

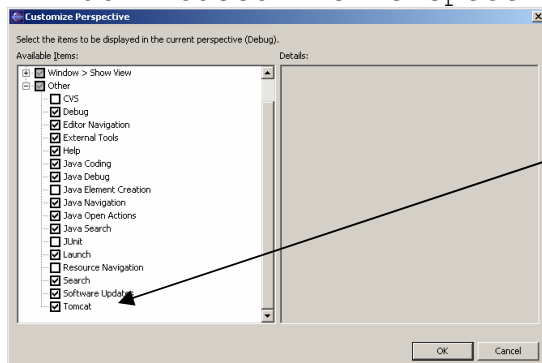


Tomcat hat also die Änderung erkannt und gleich berücksichtigt, aus dem Debuggen heraus!

### Hinweis:

Damit Sie die Tomcat Steuerelemente auch in der Debug Perspektive zur Verfügung haben, müssen Sie die Perspektive entsprechend anpassen:

“Window->Customize Perspective“



ECLIPSE TOMCAT PLUGIN STARTHILFE.....	1
1.1. INSTALLATION .....	1
1.2. ANPASSEN DER INSTALLATION .....	2
1.3. TOMCAT HELLO WORLD PROJEKT.....	4
1.4. HOT CODE REPLACEMENT.....	11