

## IN DIESER KURSEINHEIT

- *Einleitung*
  - Um was geht's eigentlich?
- *Hello World*
  - Das Ant Skript
- *Mehrere Targets und Properties*
  - Hello World Ausgabe
- *Ant Launch Configurations*
  - Definition
  - Modifikation
  - Nutzung
- *Ant View*
- *Project Builder Ant Skripte*
  - JAR Builder

## *Ant in Eclipse Starthilfe*

### **1.1. Einleitung**

In der Ant Starthilfe haben wir uns an Ant herangewagt und erste einfache Ant Skripte geschrieben. Jetzt wir Ant in Eclipse nutzen, beispielsweise um SOAP Services bei Tomcat anzumelden oder verfügbare Web Services aufzulisten.

### **1.2. Das erste Ant Skript in Eclipse**

... ist auch wieder eine reine Textausgabe („Hello World“ oder so etwas ähnliches). Die Ant Task ist also eigentlich nur `@echo <Text>`.

Als erstes müssen wir eine Datei kreieren, mit irgend einem Namen! Aber der Dateityp muss .XML sein, denn Eclipse geht davon aus, dass eine XML Datei zu Ant gehört.

Für diese Starthilfe habe ich ein *Java Projekt* in Eclipse eröffnet (*ApacheAnt*); Java, weil wir anschliessend noch Ant Skripte bauen, welche Java Klassen verwenden.

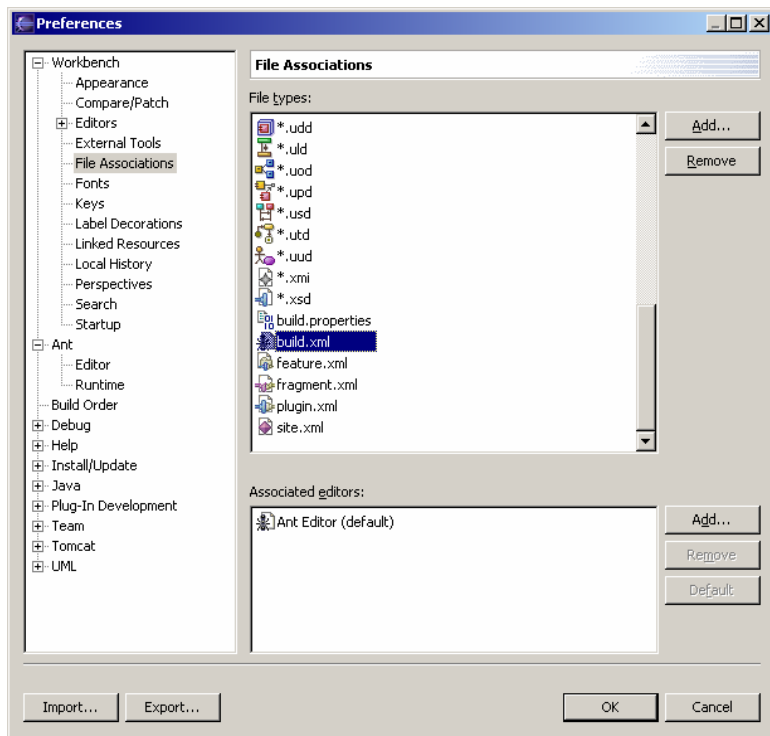
Im Java Projekt habe ich ein Package `HelloWorld` angelegt und darin die Datei `build.xml`.

Jetzt wird's bereits spannend: Eclipse verfügt über einen Ant Editor mit :

- Syntax Coloring
- Content Assistent
- Outline View

# ANT IN ECLIPSE

Falls dies nicht der Fall ist, dann wurde vermutlich der Text-Editor, nicht der Ant Editor aufgemacht. In diesem Fall müssen Sie den Dateityp XML mit dem Ant Editor assoziieren:  
Window -> Preferences -> Workbench -> File Association.



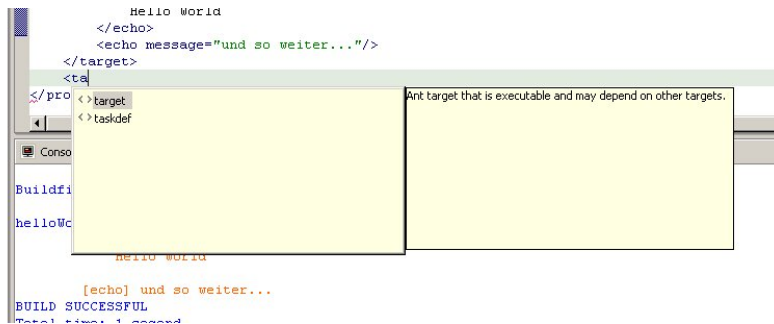
Die Farbgebung sehen Sie sofort, wenn Sie beispielsweise das Hello World Skript aus der Ant Starthilfe in die oben eröffnete Datei kopieren:

```
<?xml version="1.0" ?>
<!-- HelloWorld als ant Build.xml //-->

<!-- jedes Ant Skript besteht aus einem Projekt
      Pro Projekt muss ein Default Target definiert sein!
-->
<project default="helloWorld">
<!-- jedes Ant Project besteht aus einer oder mehreren Targets, in denen
Tasks ausgefuehrt werden
      Ant definiert viele Tasks : sie koennen diese einfach einsetzen
      Welche Tasks existieren sehen Sie im Manual. Wir gehen auf einige
ein!
-->
    <target name="helloWorld">
        <echo>
            Hello World
        </echo>
        <echo message="und so weiter..."/>
    </target>
</project>
```

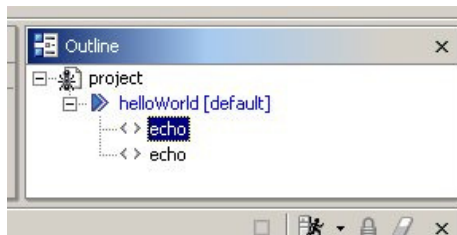
# ANT IN ECLIPSE

Den Inhaltsassistenten können Sie provozieren, indem Sie in der obigen Datei anfangen, eine neue Task zu spezifizieren:



- Tippen Sie <ta ein: es erscheinen die zwei obigen Fenster mit Ergänzungshinweisen.
- Falls der Assistent nicht erscheinen will, dann klicken Sie CTL+Leerzeichen, um ihn aufzuwecken.
- Wählen Sie die passende Vervollständigung aus.
- Ergänzen Sie Ihr Skript entsprechend Ihren Anforderungen.

Sie können eine gute Übersicht über den Inhalt Ihres Skripts erhalten, wenn Sie die Outline Sicht öffnen:



- jedes Target ist sichtbar.

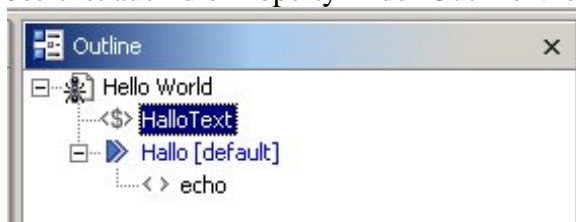
Jetzt bauen wir unser Hello World Skript so um, dass wir auch noch eine Property einsetzen können:

- wir definieren den Text in einer Property und verwenden diese im Echo :

```
<?xml version="1.0" ?>
<project name="Hello World" default="Hallo" basedir=".">
  <property name="HalloText" value="Hallo Du da!"/>

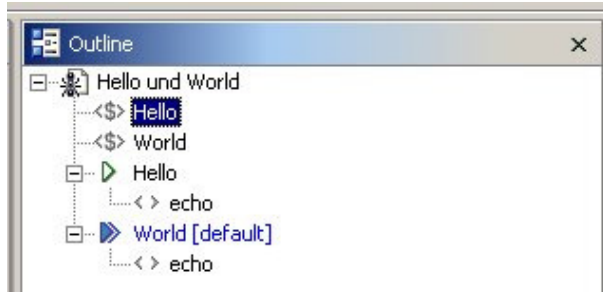
  <target name="Hallo">
    <echo>${HalloText}</echo>
  </target>
</project>
```

Jetzt ist auch die Property in der Outline View sichtbar:



## 1.3. Skripts mit mehreren Targets und Properties

Jetzt können wir den Text auch auf zwei Targets und zwei Properties aufteilen: Hello und World. Wir haben damit zwei Targets und in der Outline View wird's ein bisschen interessanter:



Wenn Sie jetzt noch „World“ zum Default Target machen und diese durch `depends="Hello"` ergänzen, erhalten Sie folgendes Skript:

```
<?xml version="1.0" ?>
<project name="Hello und World" default="World" basedir=".">
  <property name="Hello" value="Hallo"/>
  <property name="World" value=" Du da!"/>

  <target name="Hello">
    <echo>${Hello}</echo>
  </target>
  <target name="World" depends="Hello">
    <echo>${World}</echo>
  </target>
</project>
```

und folgende Ausgabe:

```
Buildfile:
C:\Java\Eclipse\workspace\ApacheAnt\src\HelloWorld\echoHelloWorld.xml

Hello:
    [echo] Hallo

World:
    [echo] Du da!
BUILD SUCCESSFUL
Total time: 180 milliseconds
```

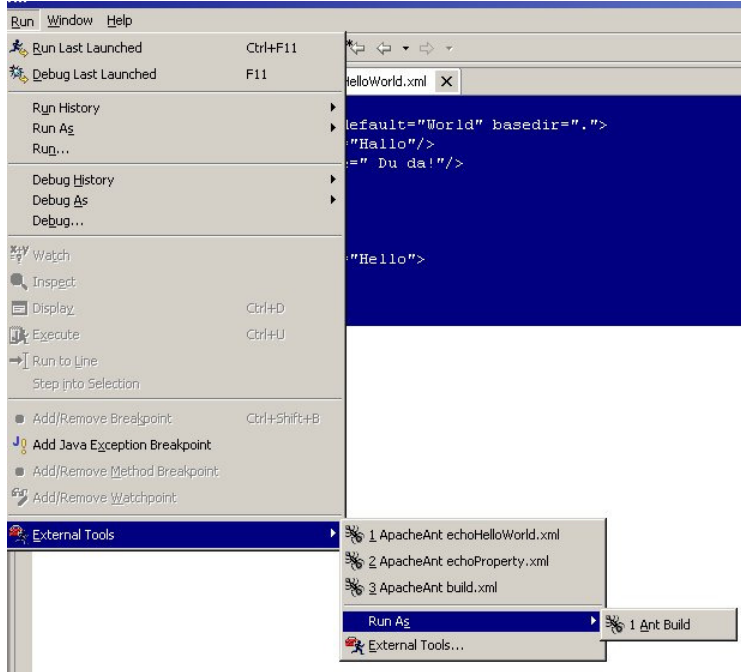
Jetzt stellen Sie sich sicher die Frage: wie führt man das Skript aus.

# ANT IN ECLIPSE

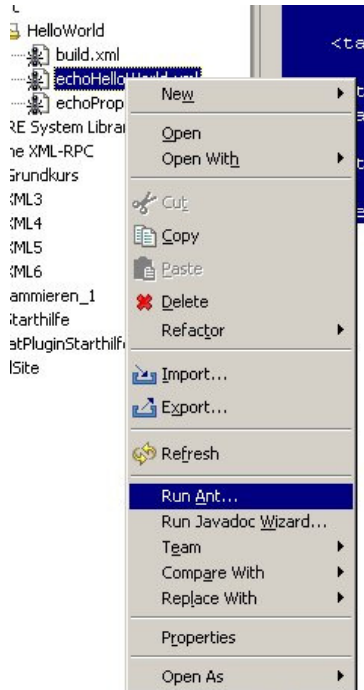
## 1.4. Die Ant Launch Configurations

Sie haben's sicher schon herausgefunden:

- entweder mit Run -> External Tools -> Run As -> Ant Build oder

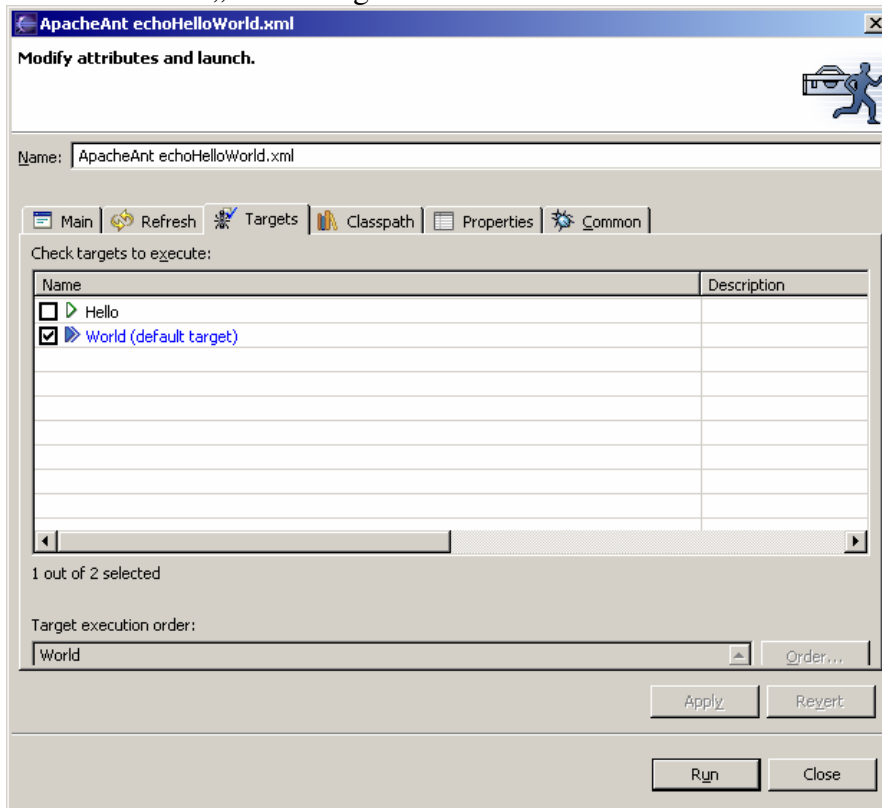


- aus dem Kontextmenü in der Package Explorer View (rechte Maustaste)



# ANT IN ECLIPSE

- es erscheint ein „Ant Dialog“:



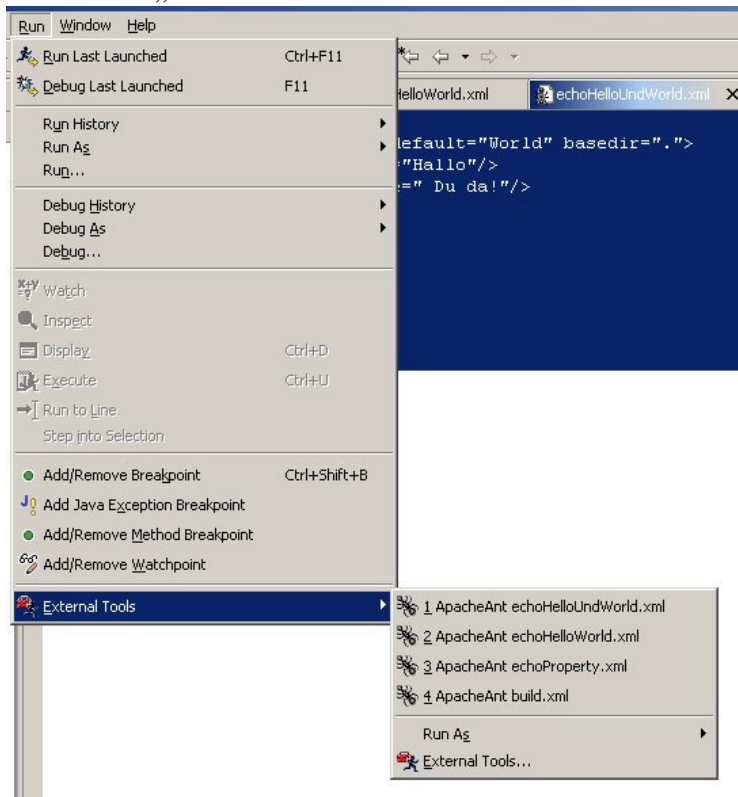
Als Testskript verwende ich das obige ohne die depends Option (echoHelloUndWorld.xml).

- mit „Run“ wird das Skript ausgeführt.
- Falls ein anderes als das Default Target ausgeführt werden soll, dann kann dies einfach angeklickt werden. Und schon hat mein ein neues Target. Die Änderung wird nicht im XML, sondern in einer sogenannten „**Ant Launch Configuration**“ abgespeichert.

# ANT IN ECLIPSE

Angenommen, Sie wollen mehrere Konfigurationen nebeneinander benutzen, aber nicht jedes Mal den Ant Dialog ausfüllen oder mutieren, dann bietet Ihnen Eclipse die Möglichkeit Ant Optionen zu speichern und wieder zu verwenden:

- Öffnen Sie „Run -> External Tools“:

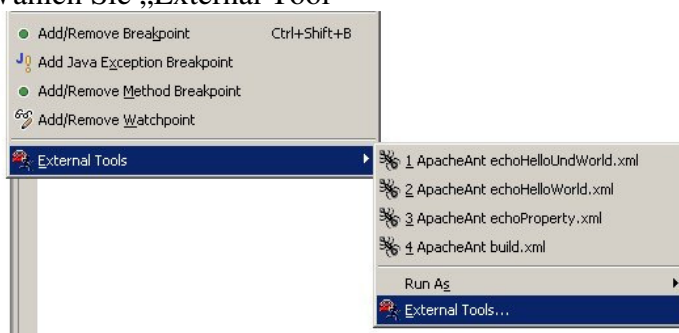


Alle Konfigurationen, die Sie benutzt haben, finden Sie hier wieder.

- Klicken Sie auf die von Ihnen gewünschte, und das entsprechende Skript wird in der gespeicherten Konfiguration ausgeführt.

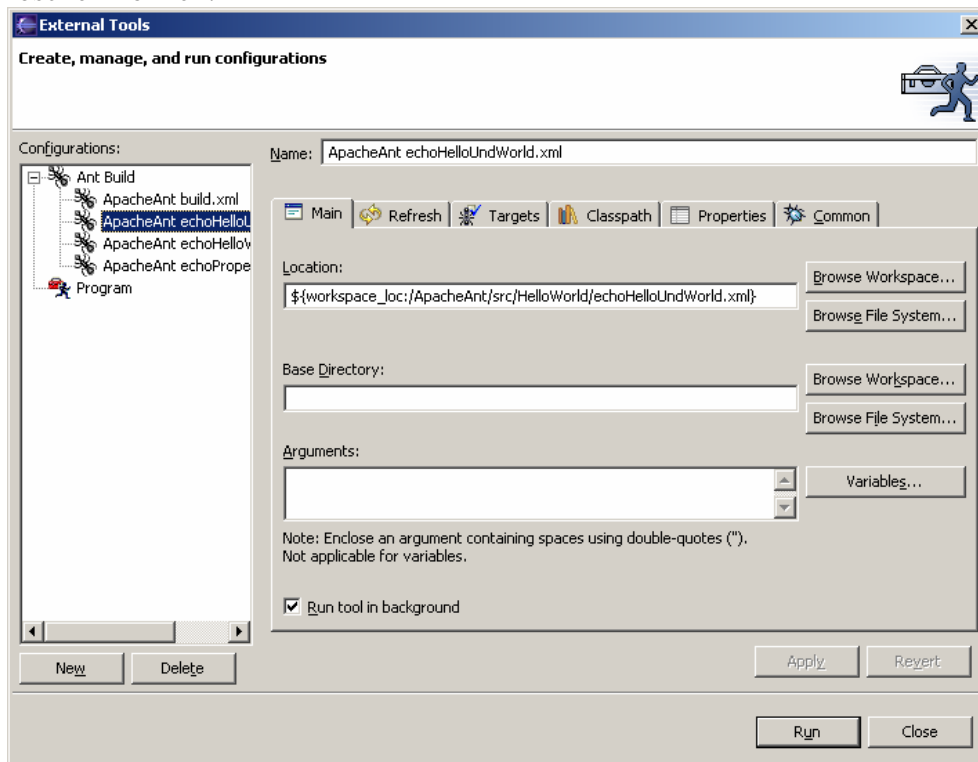
Sie können die Konfigurationen auch ändern oder löschen:

- Wählen Sie „External Tool“



# ANT IN ECLIPSE

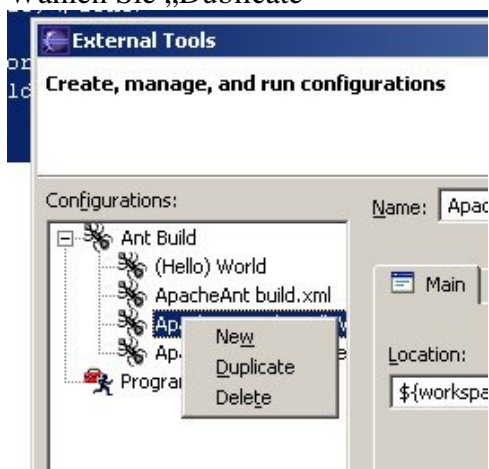
- Es öffnet sich ein Fenster, indem Sie die Launching Configurations editieren oder löschen können:



- Sie können beispielsweise den Namen (ganz oben) etwas sinnvoller und bedeutungsvoller machen (aussagekräftiger).

Wenn Ihnen die Konfiguration als Basis für Varianten dienen soll, dann können Sie auch gleich Duplikate herstellen:

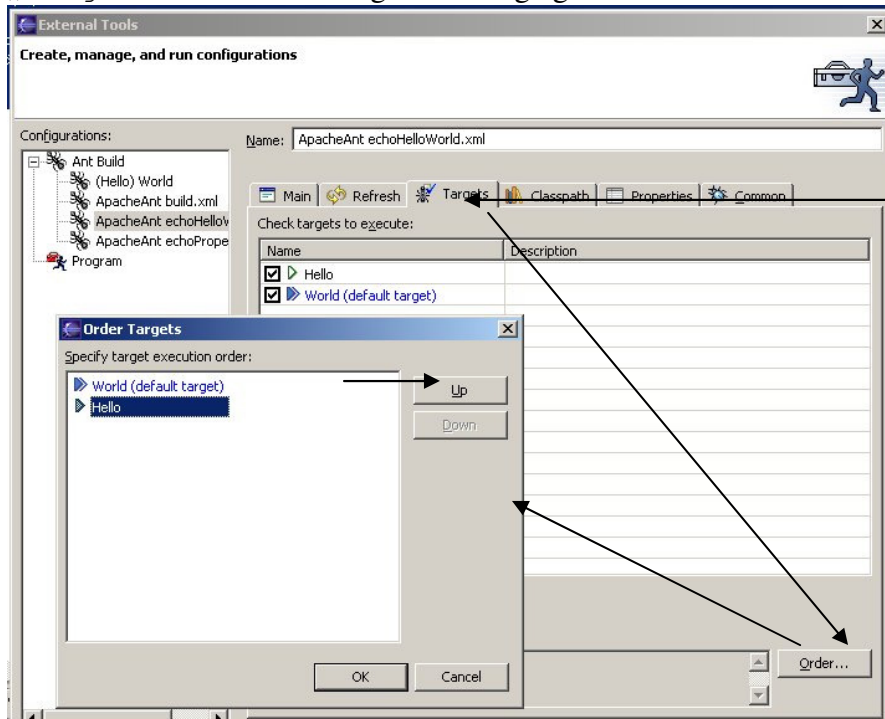
- Wählen Sie im „External Tool“ Fenster auf der linken Seite das Skript aus, welches Sie kopieren möchten und klicken Sie auf die rechte Maustaste (Kontextmenü).
- Wählen Sie „Duplicate“





# ANT IN ECLIPSE

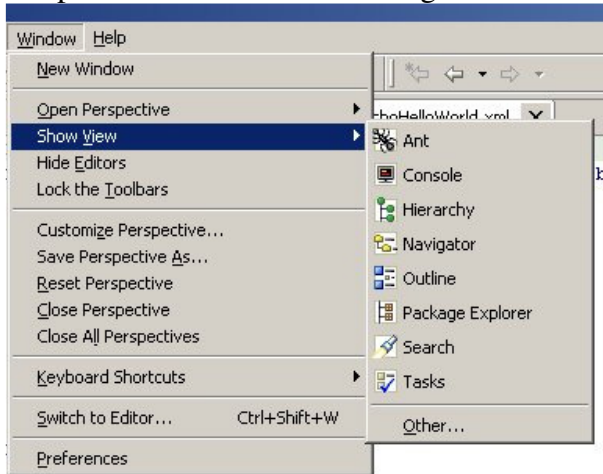
- Passen Sie die duplizierte Launch Configuration gemäss Ihren Anforderungen an. Beispiel: passen Sie die Reihenfolge der Target Ausführung an, indem Sie unter „Targets“ die Ausführungsreihenfolge gemäss Ihren Wünschen verändern:



# ANT IN ECLIPSE

## 1.5. Die Ant View

Eclipse bietet Ihnen auch eine eigene Ant View (Windows -> Show View -> Ant):

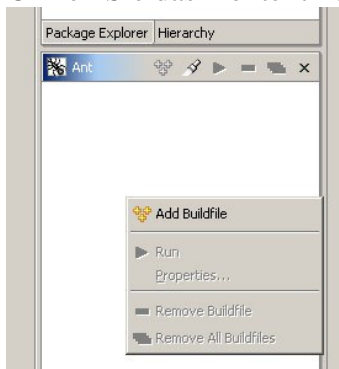


Sie können diese auch in Ihre Arbeitsumgebung einbauen:

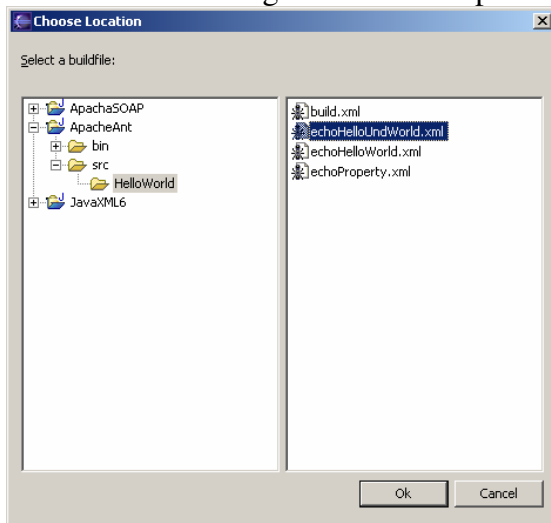
- Öffnen Sie die Sicht
- Verschieben Sie die Sicht an die von Ihnen gewünschte Stelle (mit der Maus)

Jetzt können Sie in der Ant View Buildskripts platzieren:

- Öffnen Sie das Kontextmenü (rechte Maustaste)



- und wählen Sie das gewünschte Skript aus



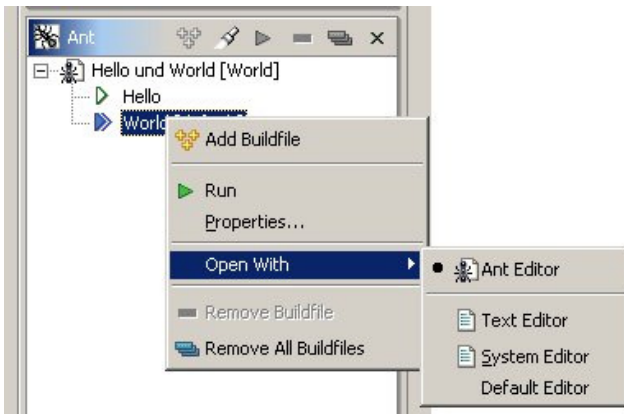
# ANT IN ECLIPSE

- in der Ant View erscheint das Skript:



Oben sehen Sie Controls zum Hinzufügen, Suchen und Hinzufügen, Starten, Löschen und Alle Löschen (von links nach rechts).

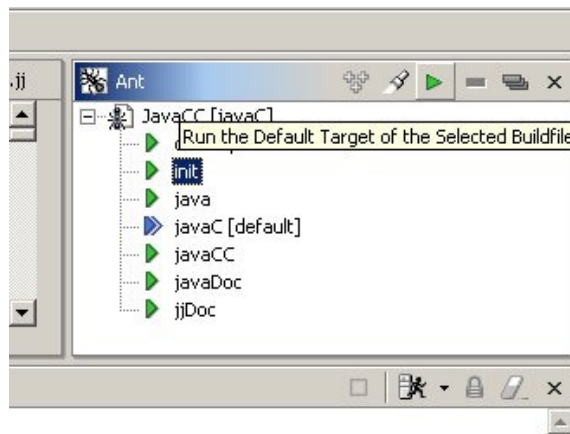
- Das Kontextmenü erlaubt Ihnen, weitere Skripte hinzuzufügen, auszuführen, das Launcher Fenster zu öffnen (Properties) oder zu editieren:



Remove entfernt lediglich den Eintrag in der Ant View, nicht das Ant Skript!

Das Starten einer beliebigen Task ist in der Ant View sehr einfach:

- klicken Sie auf die Task; diese wird dann ausgeführt
- weitere Manipulationen erkennen Sie aus dem Menü oben in der View



## 1.6. Ant Skripte als Erweiterung des Projekt Builders

Sie haben die Möglichkeit Ant Skripte zu definieren, welche beim Aktualisieren des Projekts ausgeführt werden. Dies erlaubt Ihnen beispielsweise:

- Java Archive zu erstellen
- Archive zu deployen
- Dateien an die passende Stelle zu kopieren.
- Generell:
  - Pre- und Post-Processing Aufgaben zu übernehmen.

### 1.6.1. Jar Builder Ant Projekt Builder Skript

Der Einfachheit halber starten wir mit einem Skript, welches eine jar Datei bildet. Zuerst legen wir ein neues Package an, in dem wir ein einfaches Hello World Java Programm schreiben.

- new -> Package -> projectBuilder
- new -> Class -> HelloWorld
- `System.out.println("Hello World");` im main(...) einfügen

Nun definieren wir unser Projekt Builder Ant Skript:

- new -> File -> projectBuilder.xml

Kontextmenü: open/(open with -> Ant Editor)

- tippen Sie das folgend Skript ein

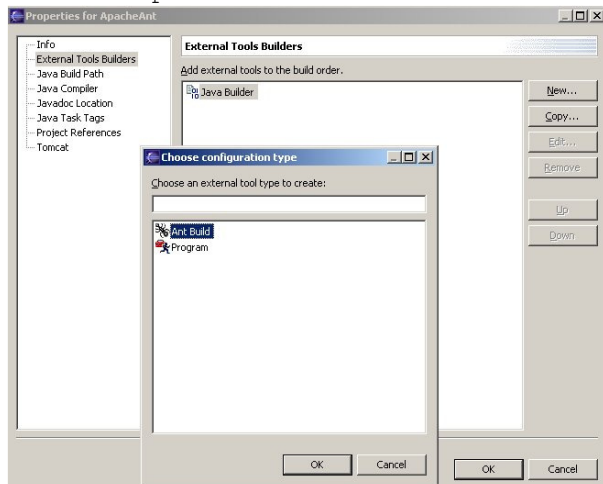
```
<?xml version="1.0" encoding="UTF-8"?>
<project name="JAR Builder" default="jarBuilder" basedir=".">
  <target name="jarBuilder" description="Kreieren eines JAR Archives
                                     (Project Builder)">
    <jar jarfile="HelloWorld.jar" includes="*.class"
        basedir="."></jar>
  </target>
</project>
```

Jetzt müssen wir das Skript als Project Builder Skript in unser Projekt einbinden.

In der Package Explorer View :

- Kontextmenü

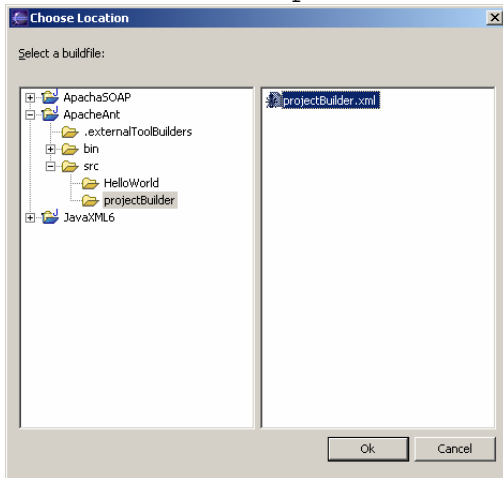
Properties -> External Tools Builder -> new -> Ant Build -> OK



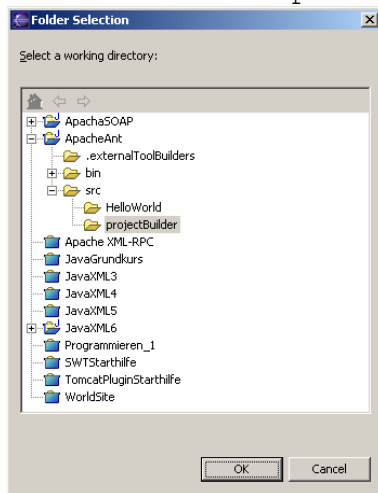
# ANT IN ECLIPSE

Ein neues Fenster öffnet sich und wir müssen die Properties für den neuen Builder angeben:

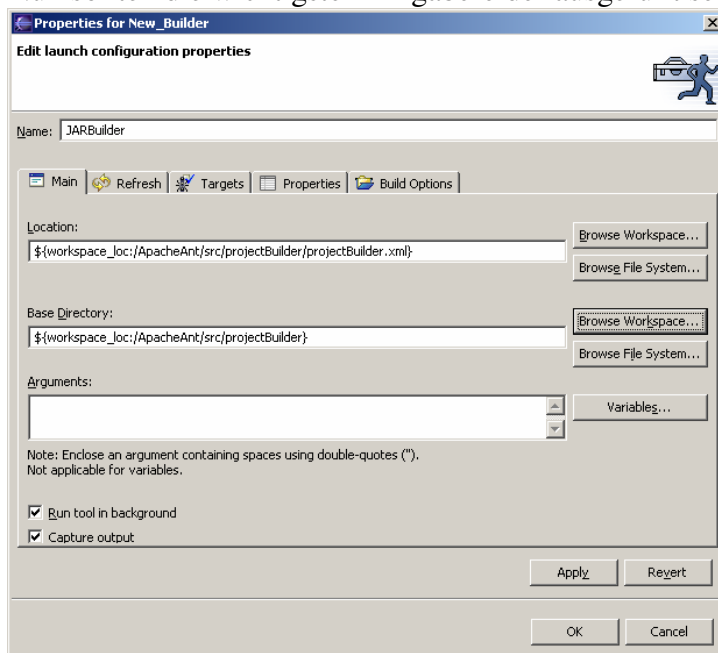
- Geben Sie einen sinnvollen Namen
- Mit Browse Workspace finden Sie das Skript in Ihrem Projekt



- Mit Browse Workspace finden Sie das Base Directory in Ihrem Projekt



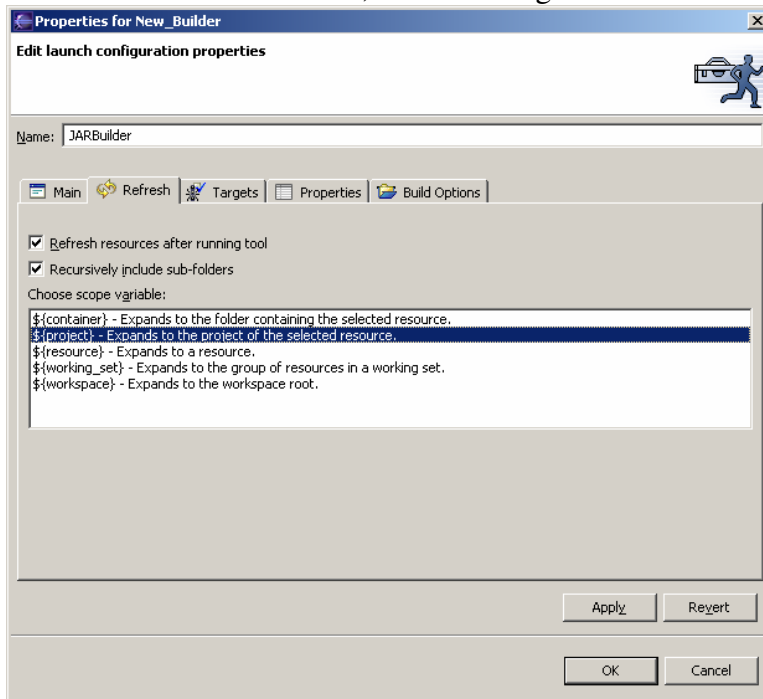
Nun sollten die wichtigsten Eingabefelder ausgefüllt sein:



# ANT IN ECLIPSE

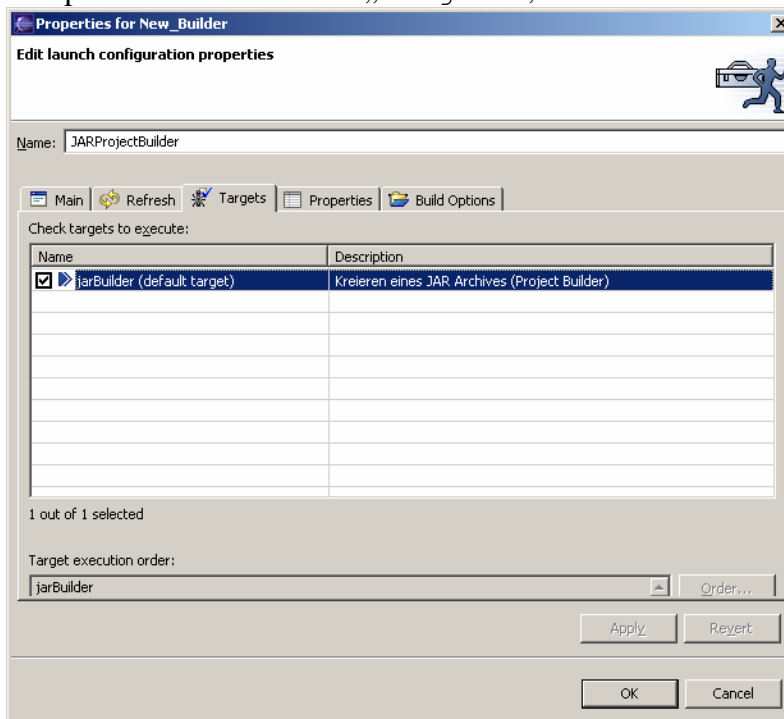
Zusätzlich legen wir das Verhalten fest.

- Im Tab „Refresh“ des Properties Fensters selektieren wir zweite Option, damit wir das JAR Archiv auch sehen, nachdem es gebildet wurde:



Stellen Sie sicher, dass Sie das korrekte Target aktiviert haben.

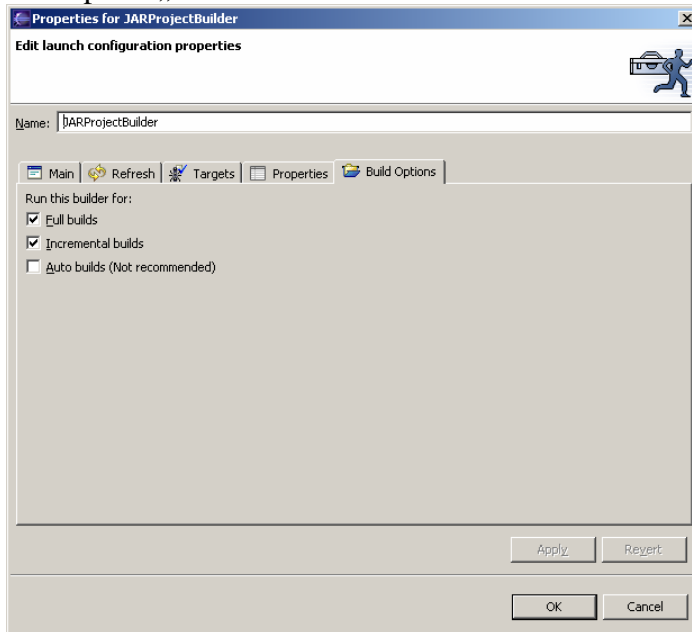
- überprüfen Sie dies im Tab „Targets“,



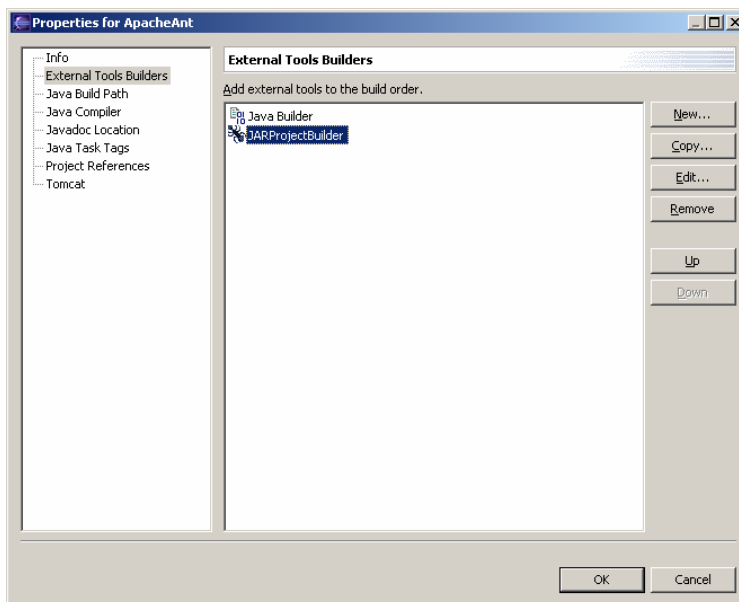
# ANT IN ECLIPSE

Als letztes überprüfen Sie die Einstellungen im „Build Options“ Tab.

- Die Option „Auto Build“ wird aus Performance-Gründen nicht empfohlen



Schliessen Sie die Definition Ihres Builders mit „OK“ ab. Falls Sie später Änderungen vornehmen wollen, so können sie jederzeit über „Properties -> External Tool Builders -> <Ihr Builder> -> Edit“ wieder in diese Fenster zurück kehren und Anpassungen vornehmen.



Der Java Builder wird immer ausgeführt, falls Sie ein Java Projekt kreiert haben. Sie können den JavaBuilder nicht editieren. Er wird für viele Aufgaben verwendet:

- Refactoring
- Suchen
- ...

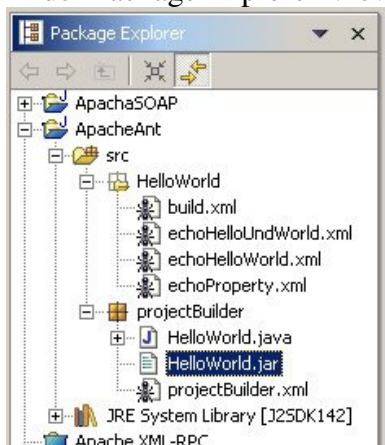
# ANT IN ECLIPSE

## 1.6.2. Ausführen des Projekt Builders

Projekt Builder werden nicht von Ihnen manuell gestartet. Die Idee hinter den Builders ist das automatische Ausführen des Build Skripts, immer wenn die Bedingungen zutreffen, die Sie in den obigen Fenstern ausgewählt haben.

Zum Testen Ihres Projekt Builders

- Aktualisieren Sie das Hello World Projekt  
Project -> Rebuild All
- In der Console View sollten Sie in etwa folgende Ausgabe sehen:  
Buildfile:  
C:\Java\Eclipse\workspace\ApacheAnt\src\projectBuilder\projectBuilder.xml  
  
jarBuilder:  
[jar] Building MANIFEST-only jar:  
:\Java\Eclipse\workspace\ApacheAnt\src\projectBuilder\HelloWorld.jar  
[jar] Building jar:  
:\Java\Eclipse\workspace\ApacheAnt\src\projectBuilder\HelloWorld.jar  
BUILD SUCCESSFUL  
Total time: 1 second
- In der Package Explorer View sehen Sie das Ergebnis, das JAR:



- Machen Sie nun eine kleine Änderung an Ihrem Programm und speichern Sie die Änderungen. Wenn Sie die „Auto Build“ Option im Properties Fenster des Builder Dialogs aktiviert haben, wird das Builder Skript automatisch ausgeführt (bei jedem Speichern).  
Falls Ihnen die dauernde Ausgabe auf die Nerven geht:  
schalten Sie „Auto Build“ einfach wieder aus.

Mit diesem Einsteiger Builder Skript schliessen wir die Starthilfe für “Ant in Eclipse”.



# ANT IN ECLIPSE

<b>ANT IN ECLIPSE STARTHILFE .....</b>	<b>1</b>
1.1. EINLEITUNG .....	1
1.2. DAS ERSTE ANT SKRIPT IN ECLIPSE .....	1
1.3. SKRIPTS MIT MEHREREN TARGETS UND PROPERTIES .....	4
1.4. DIE ANT LAUNCH CONFIGURATIONS .....	5
1.5. DIE ANT VIEW .....	10
1.6. ANT SKRIPTE ALS ERWEITERUNG DES PROJEKT BUILDERS .....	12
1.6.1. <i>Jar Builder Ant Projekt Builder Skript</i> .....	12
1.6.2. <i>Ausführen des Projekt Builders</i> .....	16