

Sphinx-4-Tutorial

Norman Rosner

nrosner@uni-potsdam.de

July 11, 2008

1 Quellen

1.1 Option A: SVN Checkout

<https://cmusphinx.svn.sourceforge.net/svnroot/cmusphinx/trunk/sphinx4>

```
[user@helios ~] svn co https://cmusphinx.svn.sourceforge.net/svnroot/cmusphinx/trunk/sphinx4
```

1.2 Option B: Kopieren aus ...

```
[user@helios ~] cp -r /home/rosner/practical/sphinx4 .
```

2 Umgang mit Eclipse

2.1 Was ist Eclipse

Eclipse^a ist ein Open-Source-Framework zur Entwicklung von Software nahezu aller Art. Die bekannteste Verwendung ist die Nutzung als Entwicklungsumgebung (IDE) für die Programmiersprache Java. Aber auch für die Entwicklung von Rich-Client-Applikationen auf Basis der Eclipse Rich Client Platform (RCP) wird es zunehmend häufiger eingesetzt. Eclipse ist nicht auf Java festgelegt und wird aufgrund seiner offenen plug-in-basierten Struktur mittlerweile für sehr unterschiedliche Entwicklungsaufgaben eingesetzt, da so die Notwendigkeit für jeden Nischenhersteller entfällt, seine eigene IDE für jedes Entwicklungstool zu entwickeln.[...] ^b

^a<http://www.eclipse.org>

^bhttp://de.wikipedia.org/wiki/Eclipse_%28Software%29

Eclipse Homepage

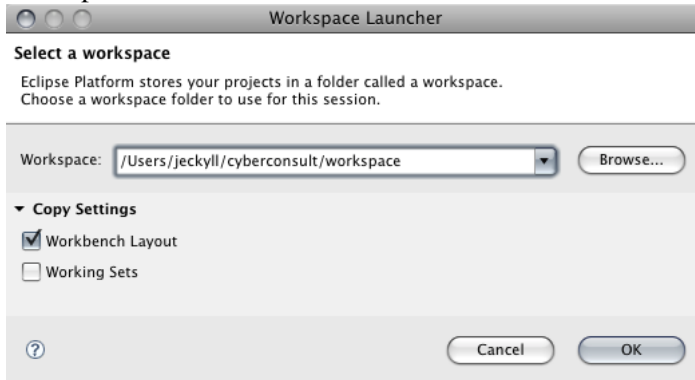
Wikipedia-Eintrag

2.2 Starten von Eclipse

1. Aufruf auf helios über

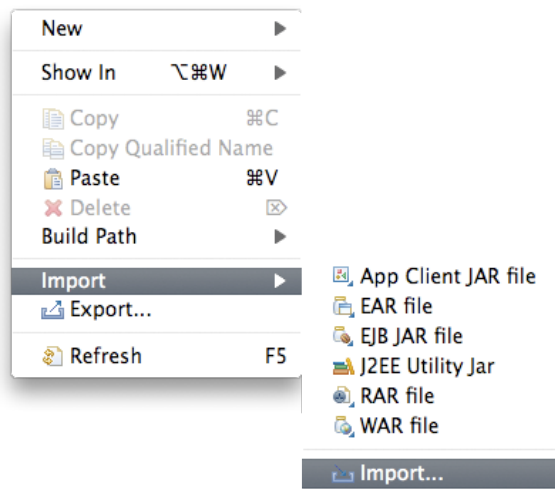
```
[user@helios ~] eclipse &
```

2. Workspace auswählen

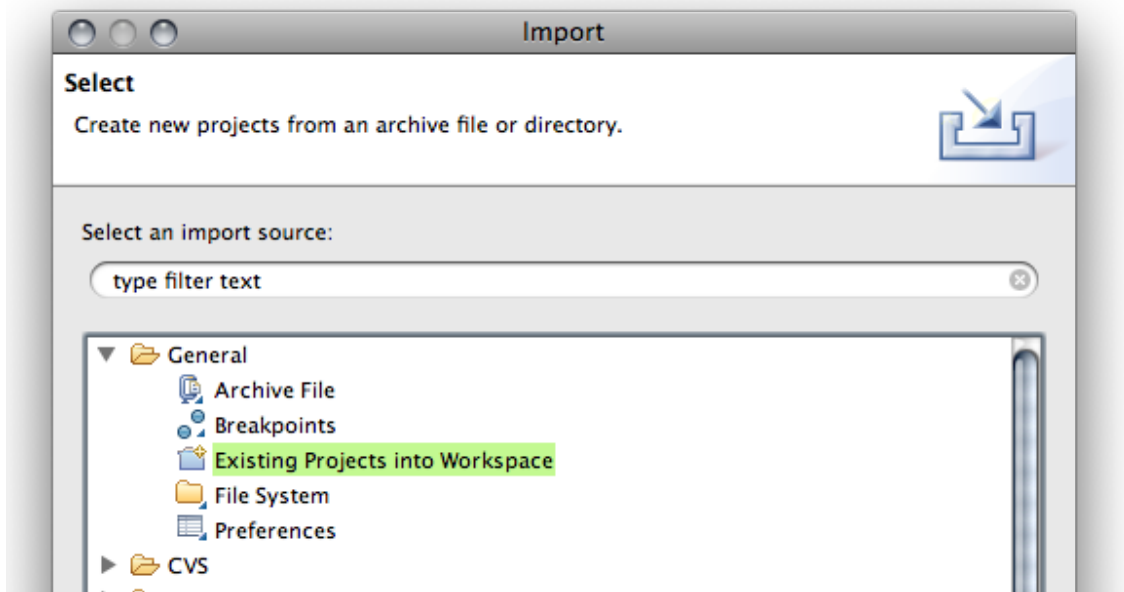


2.3 Anlegen des Projektes Sphinx4

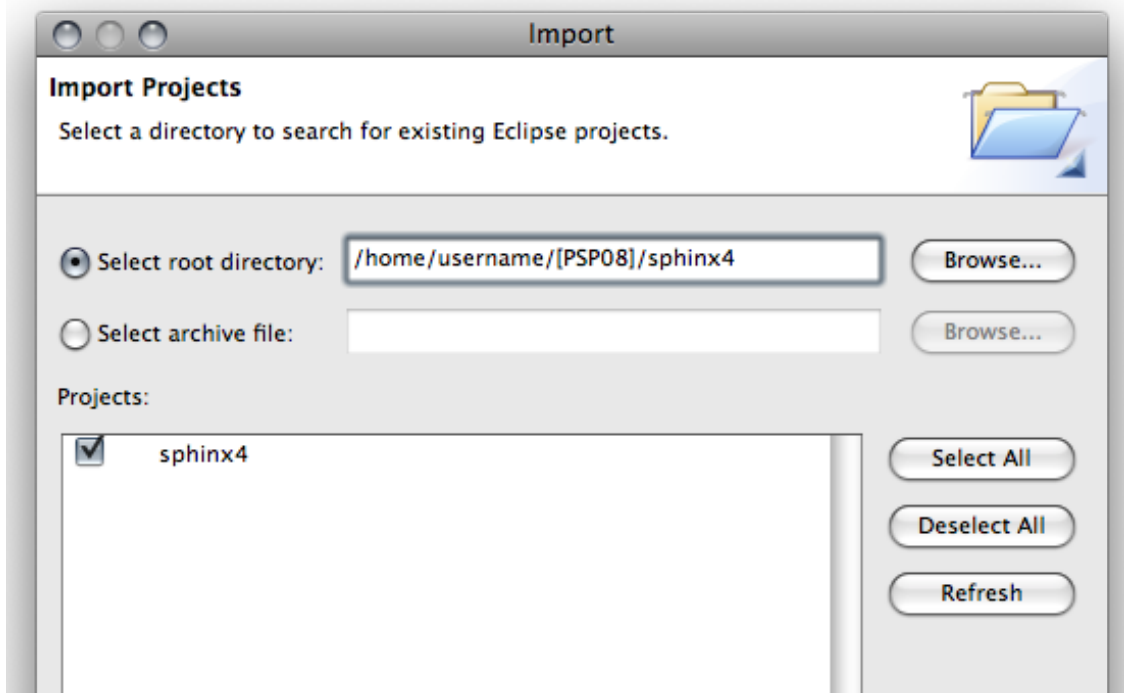
1. in der Project-Explorer-View: Kontext-Menu > Import > Import



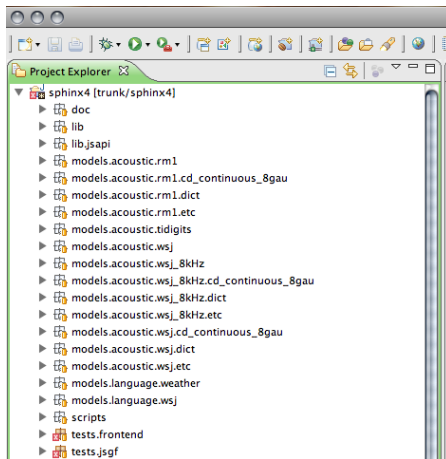
2. General > Existing Projects into Workspace



3. Select root directory: Pfad zu Deinem Sphinx4-Ordner

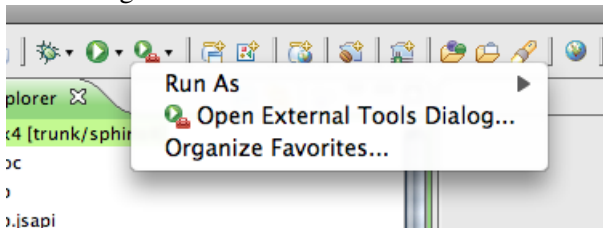


4. Wenn alles richtig gemacht wurde sieht es ungefähr so aus im Project-Explorer:

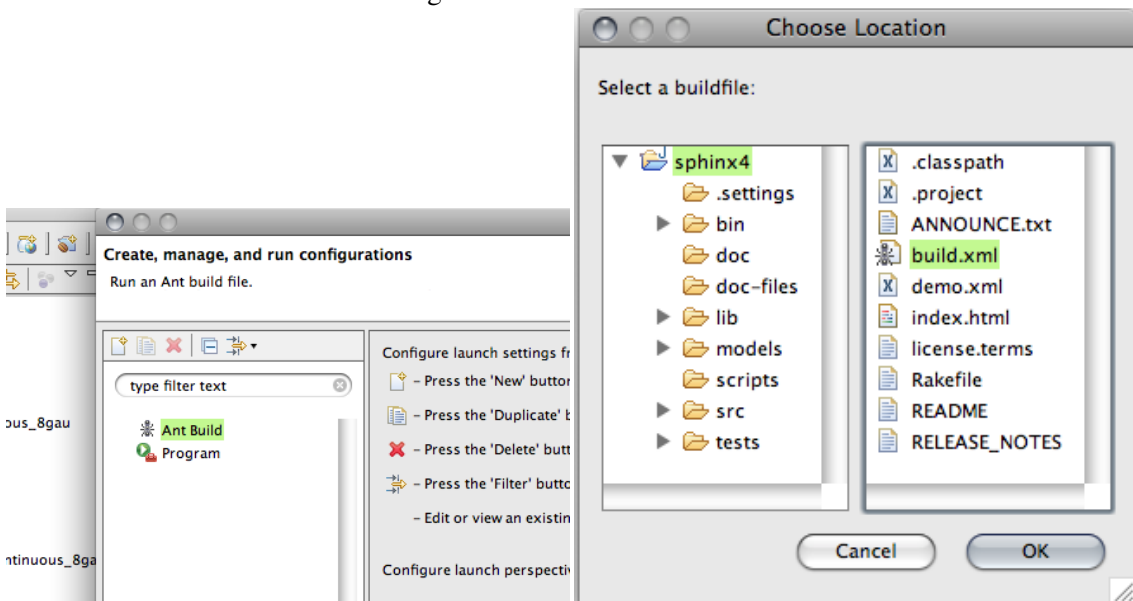


2.4 Sphinx 4 kompilieren

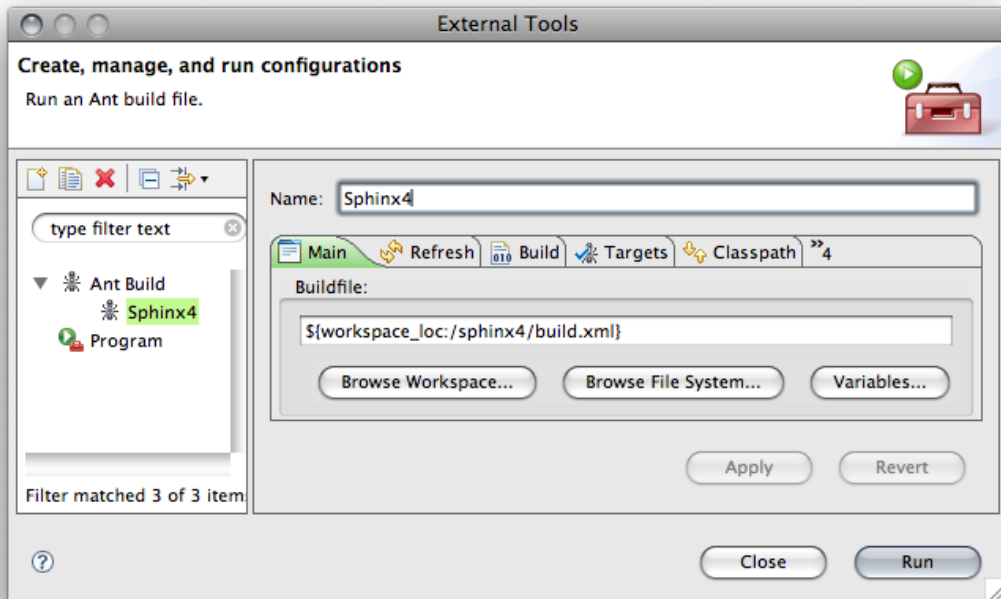
1. Ant Dialog aufrufen:> Run > External Tools > Open External Tools Dialog



2. Links oben auf New Launch Configuration klicken
3. New Launch Configuration konfigurieren:
Browse Workspace: Sphinx4 in der linken Spalten auswählen und in der rechten Spalte build.xml auswählen und bestätigen.



4. Resultat ausführen über Apply > Run

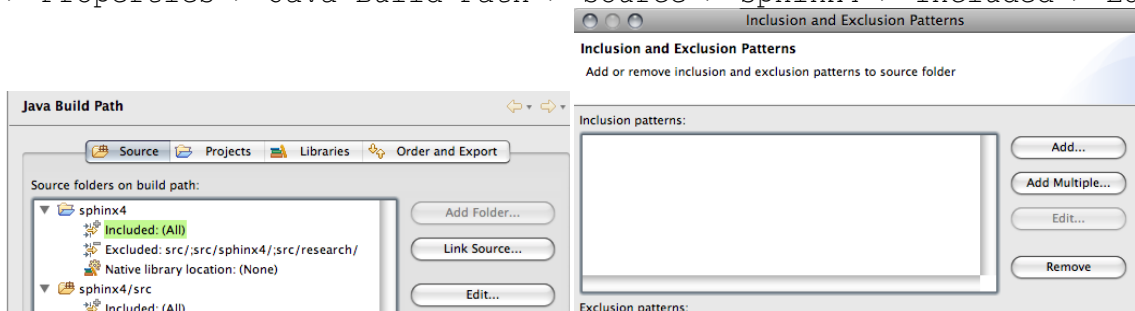


5. F5 oder im Kontext-Menu auf Refresh klicken

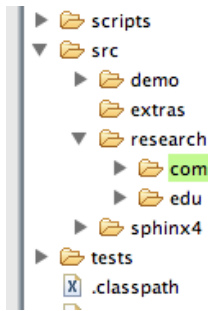
2.5 Einstellungen

2.5.1 Include-Patterns

1. Sphinx4 im Package-Explorer markieren und das Kontext-Menu aufrufen.
> Properties > Java Build Path > Source > sphinx4 > Included > Edit



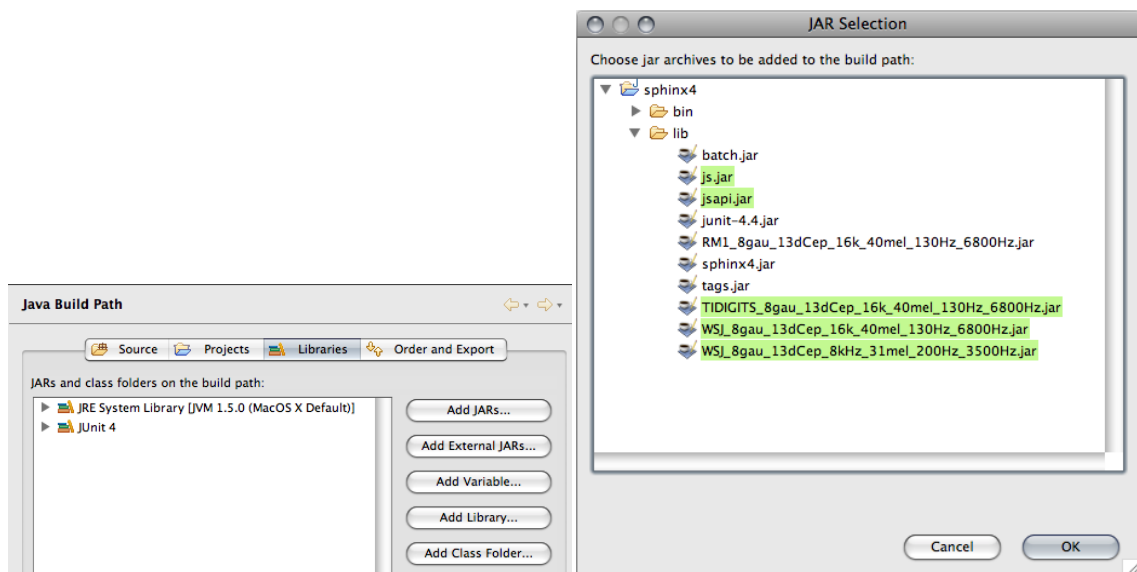
2. Add Multiple: Dort den Ordner /src/research/com/ auswählen und akzeptieren und dann unten rechts auf Finish klicken.



2.5.2 Bibliotheken

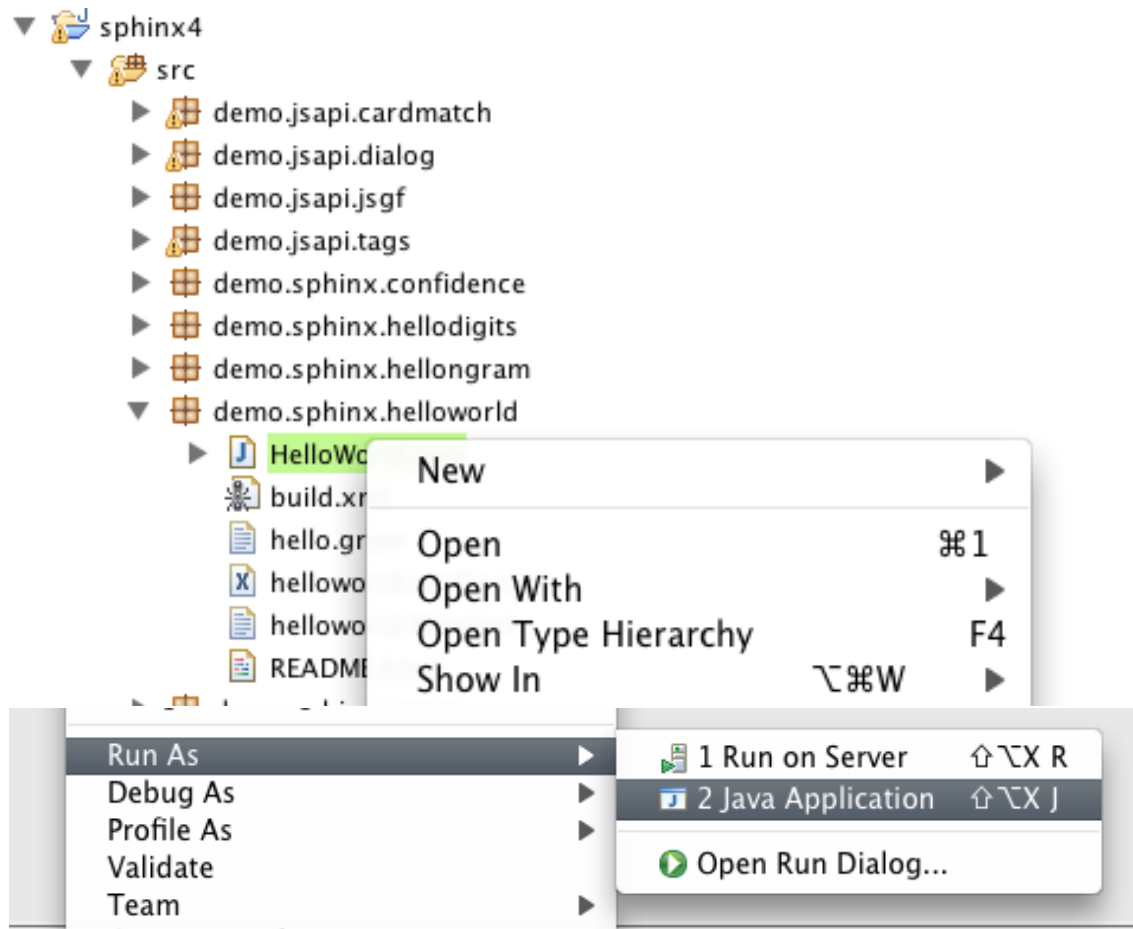
1. Sphinx4 im Package-Explorer markieren und das Kontext-Menü aufrufen.
 > Properties > Java Build Path > Libraries > Add JARs...
 Aus dem Verzeichnis /lib/ die folgende Bibliotheken auswählen und bestätigen:

- js.jar
- jsapi.jar
- TIDIGITS_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar
- WSJ_8gau_13dCep_8kHz_31mel_200Hz_3500Hz.jar
- WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar



2.6 Demos ausprobieren

1. Da nun alles soweit eingerichtet ist, dass keine Kompilationsfehler auftreten, kann nun eine Demo ausgeführt werden. Dazu einfach eine Demo aus dem Paket demo.sphinx auswählen. Zum Beispiel helloworld. Dort das Kontextmenü der Datei HelloWorld.java aufrufen und den Menüpunkt Run As und dort den Untermenüpunkt As Java Application.



2. Es gibt nur noch ein Problem: Die virtuelle Maschine benötigt mehr Speicher, um allein die Demo-Anwendung auszuführen! Um das zu beheben, öffnet man zunächst den Run Dialog. Der entsprechende Menüpunkt befindet sich unter dem Untermenüpunkt fürs Ausführen.



3. Nun muss der Virtuellen Maschine mitgeteilt werden, dass sie mehr Speicher benutzen darf und vor allem wie viel Speicher benutzt werden darf. Dazu im Run Dialog die ausführbare Klasse HelloWorld unter Java Application auswählen und in dem Reiter (x) = Arguments unter VM Arguments den Wert -Xmx128M eintragen. Wenn mehr Speicher zur Verfügung steht, kann natürlich auch dementsprechend mehr zugewiesen werden, wobei 128M ausreichend sind für die HelloWorld-Anwendung.

