Sphinx-4 Setup on *Eclipse*

Installation

- **Required Software**

  Sphinx-4 has been built and tested on the Solaris™ Operating Environment, Mac OS X, Linux and Win32 operating systems. Running, building, testing, and debugging Sphinx-4 using Eclipse requires additional software. Before you start, you will need the following software available on your machine.

  - **Java SE JDK 5** or better. Go to java.sun.com, and select "Java SE". At the time of writing, the latest release version is 6, which is the one we recommend.

  - **Eclipse 3.0** or better, available at www.eclipse.org. The current documentation was written based on **Eclipse 3.1**. If you are using other version, some of the options may not be exactly as described here, but semantically similar. The site has documentation that includes various user guides. You will need to install the **Java SE JDK** before installing **Eclipse** distribution. Or else, you will need to specify the path to the **Java SE JDK** within **Eclipse**.

- **Building Sphinx-4 using Eclipse**

  The software required for building Sphinx-4 using Eclipse are listed in the Required Software section. The following sections document one way and perhaps, the simplest way, of setting up Sphinx-4 in Eclipse. However, there are other possible ways to set up Sphinx-4 in Eclipse as well that are not covered in the following sections.

  Start Eclipse

  Start Eclipse by either double-clicking on the Eclipse icon or through a command line. Select a suitable path to the workspace when prompted by Eclipse. We will install Sphinx-4 source code from an existing location, but Eclipse requires a workspace to logically handle projects.

  Getting Sphinx-4

  You can get Sphinx-4 as a release package, a nightly build, or directly from the Subversion (SVN) repository. The CVS repository, even though still active, is kept mainly for regression test purposes. We recommend getting Sphinx-4 directly from the Subversion (SVN) repository.

  Eclipse does not have built-in support for SVN as it does for CVS. Subversion provides an add-on for Eclipse, but since there are other stand alone clients for SVN, we will not get into details. For more information about SVN, please visit the Subversion web site. For details specific to SourceForge.net, please check their SourceForge.net: Subversion page. For instructions specifically about how to download Sphinx-4, check the Sphinx-4 user guide.

  This set of instructions assumes that you already downloaded the Sphinx-4 code to a location of your choice. For future reference, we will refer to this location as $SPHINX4_ROOT.

  Create Sphinx-4 as an Eclipse Project

  - Select `Java Project` in the `New Project - Select a Wizard` window and click on `Next` button. This will pop a `New Project - Create a Java project` window.
  - Enter an appropriate name (recommended name is `sphinx4`) in the `Project name` field.
  - Choose `Create project from existing source` as `Contents`.
  - Click on `Browse` and browse to the location where you downloaded Sphinx-4, $SPHINX4_ROOT.
  - Click on `Finish` button.

  Setup JSAPI 1.0

  Before you build Sphinx-4, it is important to setup your environment to support the Java Speech API (JSAPI), because a number of tests and demos rely on having JSAPI installed.

  Add JSAPI 1.0 as a jar to the Eclipse Project

  While the previous step sets up a `jsapi.jar` file in the `lib` directory, this steps makes `jsapi.jar` visible within the Eclipse Project. Note that in the instructions that follow, we assume that the project name is `sphinx4`. We named it while creating a project under Eclipse earlier.

  - Select `Refresh` from the `File` menu or just press the `F5` function key on your keyboard. This will refresh the project and the `Package Manager` within Eclipse will show the jsapi.jar created in the last step.
  - Select `Properties` from the `Project` menu. This will pop a `Properties for sphinx4 - Info` window.
  - Select `Java Build Path` option in the left column. This will show up the `Java Build Path` on the right hand side of the window.
  - Click on the `Libraries` tab and then, click on `Add JARs...` button. This will pop a browsing window.
  - Select `jsapi.jar` by navigating to `lib` directory, and then click `Open` button on the browsing window.
  - Next, click on `OK` button on the `Properties for sphinx4 - Java Build Path` window. This will add `jsapi.jar` as a jar to the Eclipse Project.

Build Sphinx-4 using Ant within Eclipse

This steps creates a number of directories and files. However, for debugging purposes within Eclipse, we are concerned only with `edu.cmu.sphinx.model` directory, and `TIDIGITS_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar, WSJ_8gau_13dCep_8k_31mel_200Hz_3500Hz.jar and WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar jars`. Later, in the next step, we will `refresh` the `sphinx4` project within Eclipse to include `edu.cmu.sphinx.model` directory as project resource. We will also include the `TIDIGITS_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar, WSJ_8gau_13dCep_8k_31mel_200Hz_3500Hz.jar and WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar jars` as jars for the `sphinx4` project.

- Select `External Tools->External Tools...` from the `Run` menu. This will pop a `External Tools - Create, manage, and run configurations` window.
- Select `Ant Build` option in `Configurations`. Next, click on `New` button at the bottom left. This will display a new configuration in the `External Tools - Create, manage, and run configurations` window.
- Enter an appropriate name (we recommend `sphinx4`) of the Ant build in the `Name` field.
- Under the `Main` tab, click on `Browse Workspace...`. This will pop a `Choose Location` window.
- Select `build.xml` in the right side space, and then, click on `OK` button.
- Click on `Run` button to build Sphinx-4 using Ant.
- When compiling subsequently, the procedure is simplified. Select `External Tools->External Tools...` from the `Run` menu. Then simply select `sphinx4`.

Refresh the `sphinx4` project and select required source folders

While the previous step creates `edu.cmu.sphinx.model` directory, this step refreshes the project that allows the inclusion of `edu.cmu.sphinx.model` directory as a `sphinx4` project source. Selection of required source folders is accomplished in this step to eliminate redundant folders in the Java Build Path within Eclipse.

- Select `Refresh` from the `File` menu or just press the `F5` function key on your keyboard. This will refresh the project and the `Package Manager` within Eclipse will show all the directories, created by Ant in the previous step, as packages.
- Next, select `Properties` from the `Project` menu. This will pop a `Properties for sphinx4` window.
- Select `Java Build Path` option in the left column. This will show up the `Java Build Path` on the right hand side of the window.
- Click on the `Source` tab and then, click on `sphinx4` folder in the window pane. This will expand the `sphinx4` folder into `Included: (All)` and `Excluded: (None)`.
- Next, double-click on `Included: (All)`. This will pop a `Inclusion and Exclusion Patterns - Included and excluded resources for 'sphinx4'` window.
- Click on `Add Multiple...` button in the `Inclusion patterns` section. This will pop a browsing window, `Inclusion Pattern Selection`. Select `com` folder and click on `OK` button. Next, click on `OK` button on the `Add Inclusion Pattern` window. This will add the folders `com`, `demo`, and `edu` within the `Inclusion patterns` section, in the `Inclusion and Exclusion Patterns - Included and excluded resources for 'sphinx4'` window. Select multiple folders as you would normally do on your machine. For example, in Windows, you can select multiple folders by clicking on the folder name at the same time that you hold the CTRL key.
- Next, click on `OK` button in the `Inclusion and Exclusion Patterns - Included and excluded resources for 'sphinx4'` window.
- Next, click on `OK` button in the `Properties for sphinx4 - Java Build path` window.

Add Sphinx-4 specific jars as jars to the Eclipse Project

This step makes the `TIDIGITS_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar`, `WSJ_8gau_13dCep_8k_31mel_200Hz_3500Hz.jar`, and `WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar` in the `lib` directories visible within the Eclipse Project.

- Select `Properties` from the `Project` menu. This will pop a `Properties for sphinx4 - Info` window.
- Select `Java Build Path` option in the left column. This will show up the `Java Build Path` on the right hand side of the window.
- Click on the `Libraries` tab and then, click on `Add JARs...` button. This will pop a browsing window.
- Select `TIDIGITS_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar` by navigating to `lib` directory, and then click `Open` button on the browsing window.
- Select `WSJ_8gau_13dCep_8k_31mel_200Hz_3500Hz.jar` by navigating to `lib` directory, and then click `Open` button on the browsing window.
- Select `WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar` by navigating to `lib` directory, and then click `Open` button on the browsing window.
- Next, click on `OK` button on the `Properties for sphinx4 - Java Build Path` window. This will add these jar files as jars to the Eclipse Project.
- At this point, you should be able to run and debug applications such as demo.sphinx.hellodigits using Eclipse.

- **Run or Debug Sphinx-4**

In this step, we provide instructions to run Sphinx-4 using Eclipse.

- Select `Run...` or `Debug...` from the `Run` menu.
- Click on `Java Application` and hit the `New` button.
- Choose a name for the configuration (*e.g.* `HelloWorld`) on the `Name:` field.
- Click the `Search...` in the `Main Class` pane.
- Select of the `main` methods found (*e.g.* `HelloWorld`).
- Click on `OK` and then on `Apply` to confirm the configuration.
- Start the Debug or Run session by hitting the `Debug` or `Run` button.

- **Troubleshooting**
  1. When I try to compile Sphinx-4 with Eclipse, I get a message suggesting JAVA_HOME is not defined. What's missing?
     - Sphinx-4 uses `ant` to build, and `ant` requires the environment variable `JAVA_HOME` to be defined. `JAVA_HOME` contains the path to the Java JDK. You can define the variable as follows.
     - Unix (t)csh
       - `setenv JAVA_HOME /lab/speech/java/jdk1.5.0_04`
     - Unix (ba)sh
       - `export JAVA_HOME='/lab/speech/java/jdk1.5.0_04'`
     - Windows 2k/XP
       - Right click on the "My Computer" icon
       - On the menu, click on "Properties"
       - Click the "Advanced" tab
       - Select "Environment Variables"
       - Click "New" on the "User variables for ...." box
       - On "Variable Name", type "JAVA_HOME"
       - On "Variable Value", type "c:\Program Files\Java\jdk1.5.0_04"