

iSeismograph System Guide

1. Overview

iSeismograph is a telepresence-demonstration and educational tool built on top of RDV and Data Turbine. It takes advantage of the in-board video-camera and 3-axis accelerometer built into all new Apple MacBook products to provide an telepresence demonstration often referred to as 'Data Turbine in a Can.'

As such, the software comprises four main parts:

- Data Turbine (rbng.jar) , acting as the server.
- Real-Time Data Viewer (rdv.jar), acting as the client.
- iSight Camera and Apple Accelerometer drivers (commons.jar, isight.jar, libUnimotionLib.jnlib, sms_rbnb.jar, sms.jar), providing real-time data.
- iSeismograph Scripts (Go_iSeismograph, iseismograph.rdv, start.sh) 'gluing' the other three parts together into a user-experience.

This guide covers how to the iSeismograph scripts manage the other three components and how to generate the executable jar file installer.

2. Requirements

Data Turbine and Data Turbine Utilities are currently compatible with Java 1.5+. iSeismograph only works on Apple Macbook and Macbook Pro brand laptops. As Apple is the sole developer for Java implementations on the MacOSX platform, the only real requirement for running this software package is a Macbook (Pro) running MacOSX 10.4 or later.

Although IzPack will compile on multiple platforms, the driver code will not, and as such it is standard practice to compile this software and generate the installer binary on the MacOSX platform.

3. Compilation Overview

iSeismograph can be obtained from the [NEESforge](#) page. If you are unfamiliar with Subversion, you may familiarize yourself using information from the [Subversion homepage](#)

As is standard practice when using Subversion, the iSeismograph directory has three subdirectories labeled tags, branches, and trunk. The trunk directory will contain the latest version of iSeismograph currently under development, while the tags directory will contain trees frozen at release date. iSeismograph does not currently have alternate, or major forks in its development and so the branches directory will remain empty for the foreseeable future.

Within the trunk directory (and tag subdirectories), there should exist two subdirectories, one labeled 'iSeismograph' and one labeled 'IzPack'. These two folders correspond to the iSeismograph source and binary trees and the iSeismograph installer source tree, respectively.

iSeismograph is currently packaged using a modified version of IzPack version 4.0.1. IzPack was selected for its cross-platform capabilities. During the course of development, the IzPack 4.0.1 source tree has been customized for use in iSeismograph and another Telepresence product, Data Turbine Utilities.

Although it is possible to merge the modifications for each product into a single tree, no attempt has yet been made to do so. Primarily this is because some consideration has been made to using platform-specific install packages; these packages are often better than IzPack at customizing platform-specific issues such as environment settings.

Thus, both the DTU and iSeismograph product trees contain separate modified IzPack 4.0.1 source trees. A reference copy of the stock 4.0.1 source tree, installer binary, and documentation can be found on the [NEESforge](#) page.

There are no current plans to update the installer packages to a newer version of IzPack.

4. Compiling the IzPack Installer

IzPack can be installed on most platforms using the Java-based self-installer. As the installer simply unpacks its contents into a directory structure it is common practice at NEESit to simply include the IzPack tree, with a project subdirectory, in the source tree of a product.

Like many Java-based applications, IzPack uses the Ant platform for building its software. Those familiar with the more traditional 'make/Makefile' method of building software will find Ant similar in functionality, if not in build file syntax. More information on Ant can be found here: <http://ant.apache.org/>

Many systems come with Ant pre-installed; from a command-line simply type 'ant' and one should receive an error message if there is no build.xml file in the current directory, if the software has been installed. If the program is not found please verify the location of ant is in your path; otherwise you will need to install Ant manually.

Please note that the above link will reflect documentation for the newest versions of IzPack only.

To generate the IzPack-based iSeismograph self-executable installer, simply type:

```
compile ../iSeismograph/install.xml -b ../iSeismograph
```

from the IzPack/bin directory. This will create the file 'install.jar' in the iSeismograph directory. Please rename this file as you see fit. This file can be put up on the web for download and installation. The iSeismograph directory will appear in the Applications folder.

5. Generating the jar binaries

The iSeismograph jar binaries encapsulate the functionality that iSeismograph provides to the user, and that is accessible through the scripting glue. Currently there is no build process for generating this software, Ant or otherwise. However, it is relatively straightforward to generate a jar given what is found in the Subversion directory.

The subversion directory 'iSeismograph' contains three subdirectories, 'src', 'UNJAR', and 'lib'. The lib directory contains jar files which will be needed to generate the various jars. As such, they should be added to your CLASSPATH variable. Please note that while there hasn't yet been a need to update many of these jars to more current versions, the included rbnb.jar should always be the version the iSeismograph intends to be used with. To create the jar files that will later go in the IzPack/iSeismograph directory, change to the UNJAR/ directory from the command-line and type the following:

```
jar cf META-INF/MANIFEST.MF *
```

This will create a new jar file named containing the contents of the current directory (UNJAR/) and according to the specifications outlined in MANIFEST.MF. This file can now be moved to the installer directory for packaging.

6. Generating contents of the UNJAR directory

Much of the UNJAR directory contains additional class files supplied from third-party vendors such as Unimotion. The UNJAR directory also contains compiled class files from java class files written at NEESit to bridge the functionality of this third-party software to the user. This software can be located in the 'src' directory and compiled in the traditional manner using 'javac' and with the following CLASSPATH parameters outlined above. As of the time of this writing, there are plans to automate the manual portions of the build process.