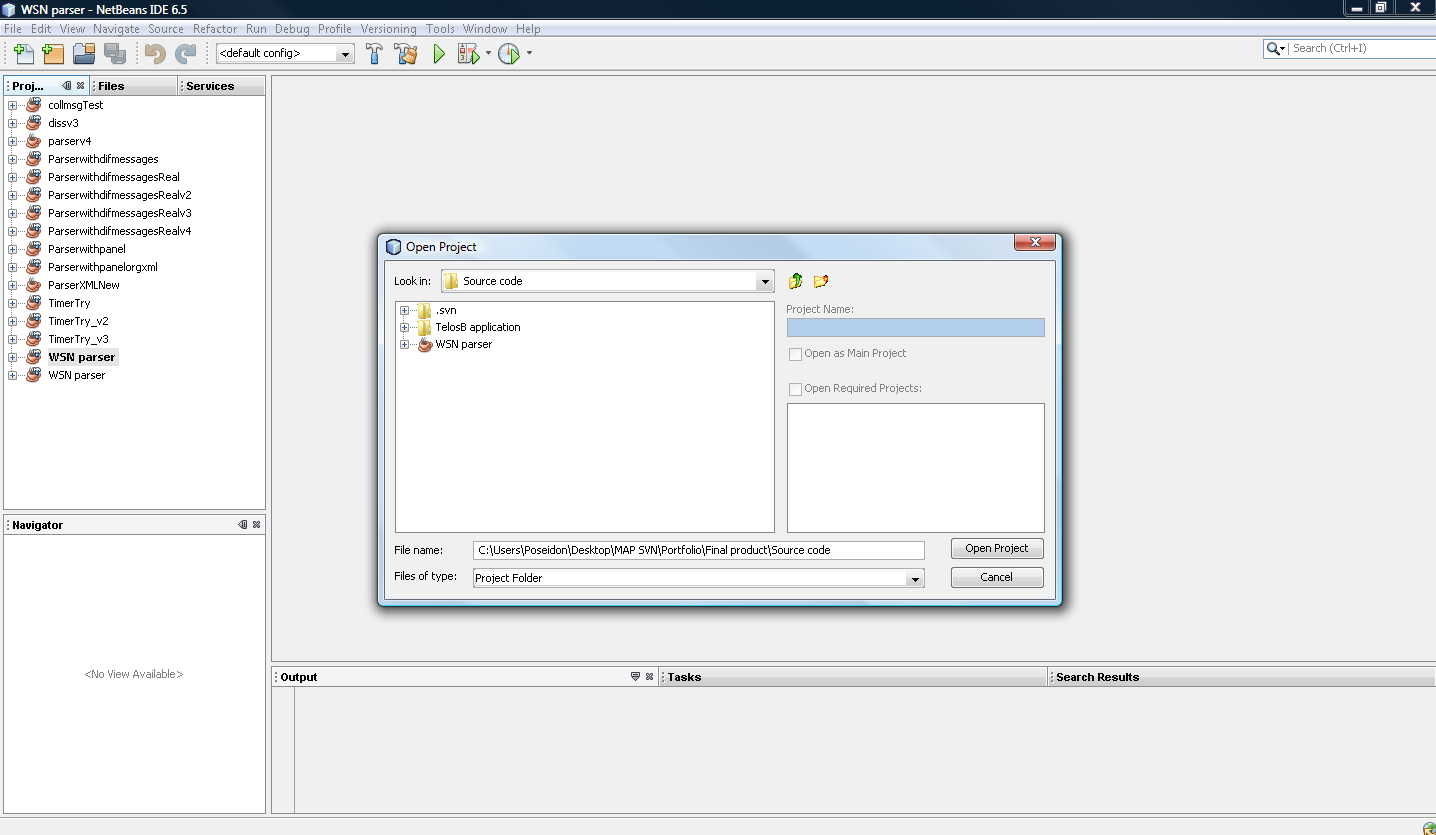
WSN parser quickstart

# Introduction

This document describes the WSN parser written in java. Before you start reading this you should preferably have read the conceptual introduction of the framework. The purpose of this document is to clarify how the parser works. Important concepts specific to the parser will be explained as well. For a more in depth explanation of the source code, head over to the code documentation.

# Opening the project

1. Open NetBeans and open a project
2. Browse to the directory containing the source code, this should be the directory /Final Product/Source
3. Double click the file: WSN\_parser to open the NetBeans project



You will see a couple of files in the project:

* CollMsg: class implements a Java interface to the 'CollMsg' message type.
* DissMsg: class implements a Java interface to the DissMsg message type.
* LocMsg: class implements a Java interface to the LocMsg message type.
* StatMsg: class implements a Java interface to the StatMsg message type.
* MyTimerTask: class to create a Timertask to be scheduled for a timer
* SerialSender: class to transmit request to the WSN
* Connectiondetails: is a form for the panel to enter your connection details
* Dexml: a class to decode a received xml message
* Xml1: a class to create an XML message for the sensor data
* Xml2: a class to create an XML message for the location data
* Xml3: a class to create an XML message for the reboot status message
* Xml4: a class to create an XML message for the reply message
* Serial is a thread that continually listens to packets over the serial interface for data and will transmit this to the controller
* Requesthandler is a thread that listens to requests transmitted by the controller and will also send a reply message.
* Sharedata is a class to share data between the two thread synchronized

# Solution Overview

When you have opened the solution, you can see the solution and the projects in the solution explorer. These projects can be divided into logical groups:

## GUI & Controller / Engine

The GUI, Controller / Engine form the part of the framework as described in the conceptual overview.

## Localization

This division should contain all the logic for the localization

## Tools

These tools are in place to speed up the development

## Simulation

Though very closely related to a unit test, this division serves the purpose to process captured data and analyze it with the different algorithms from the localization part