User Quickstart

# Introduction

This document describes how to set-up the senseless system. Prerequisites are given as well. However instructions to install external applications fall out of the scope of this document. We will refer to external manuals. Before you start reading this you should preferably have read the conceptual introduction of the framework. Doing so you will understand the functionality of each part of the framework.

This document does not describe how to use each part of the framework, just how to start them. For more information on how to use the framework we gladly refer you to the user manual.

Contents

[Introduction 1](#_Toc233004474)

[System Prerequisites 3](#_Toc233004475)

[WSN: TelosB 3](#_Toc233004476)

[Operating System 3](#_Toc233004477)

[Requirements 3](#_Toc233004478)

[Licenses 3](#_Toc233004479)

[Wsn: XML Parser 3](#_Toc233004480)

[Operating System 3](#_Toc233004481)

[Requirements 3](#_Toc233004482)

[Licenses 3](#_Toc233004483)

[Database 4](#_Toc233004484)

[Operating System 4](#_Toc233004485)

[Requirements 4](#_Toc233004486)

[Licenses 4](#_Toc233004487)

[GUI, Controller 4](#_Toc233004488)

[Operating system 4](#_Toc233004489)

[Requirements 4](#_Toc233004490)

[TelosB 5](#_Toc233004491)

[TelosB XML Parser 6](#_Toc233004492)

[MySQL DataBase 7](#_Toc233004493)

[ResToring the Database 7](#_Toc233004494)

[Commandline 7](#_Toc233004495)

[MySQL Administrator 7](#_Toc233004496)

[Using the database 10](#_Toc233004497)

[Controller & GUI 11](#_Toc233004498)

[Configuring the Controller 11](#_Toc233004499)

# System Prerequisites

This section describes the required software to start running the framework. It also described the operating systems you can use and under which license the software is covered.

## WSN: TelosB

### Operating System

* Linux: Xubuntu(os)
* Windows with Cygwin installed

### Requirements

* TinyOS 2.0.2 environment required for compilation

The preferred working environment is Xubuntos which is a Ubuntu distribution with all the necessary files for TinyOS installed. You can find more information on this link:

<http://toilers.mines.edu/Public/XubunTOS>

### Licenses

* BSD license

## Wsn: XML Parser

### Operating System

* cross-platform, TinyOS-libraries required for compilation

### Requirements

* JDK 1.5, USB-port

### Licenses

* GPL license

## Database

### Operating System

* i5/OS, Linux, Mac OS X, Netware, Solaris Windows (2000 or higher) systems.

### Requirements

* MySQL server 5.0
* MySQL Connector/ODBC 3.51-driver
* MySQL Query Browser (optional)
* MySQL Administrator (optional)
* MySQL Workbench (optional)

### Licenses

* GPL license

## GUI, Controller

### Operating system

* Windows 2000 or higher

### Requirements

* .NET framework 3.5 SP1
* Microsoft Visual Studio 2008

# TelosB

A node needs to be connected to the computer through the serial USB port. The node can be configured as the root (node id 0) of the network or as a blind node (any other node id). The network makes use of Collection: the network is symbolized as a tree with the root at the base, where the data is sent to.

## software

We program and install the applications on the motes with Linux: Xubuntos. XubunTOS simplifies the installation of TinyOS by using a Linux live CD. The bootable live CD contains a working TinyOS environment and offers the option to perform a full installation. XubunTOS is built from Xubuntu and TinyOS 2.x Debian packages (plus the TinyOS 1.x CVS repository). After installation, Debian's APT package manager can keep your software up-to-date.



## installation off the application on the mote

In the directory of your application ( /…/Senseless ) we open a terminal and type:

* *Make telosb:* De TinyOS application is compiled from this directory
* *Make telosb reinstall,nodeID:* This compiles an image from the application with the noteID, which is compatible with the Telos rev. B platform. If you choose nodeID 0, then the node is set as the root of the network, if you choose any other number that the node is a blind node (node without a know location).

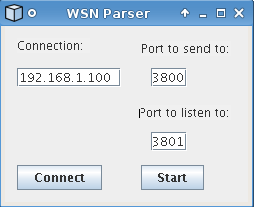
# TelosB XML Parser

Open a terminal in Xubuntos and go to the path of where the WSN\_parser.jar is situated

The following commando must be typed for compiling this file successfully.

|  |
| --- |
| export MOTECOM=serial@/dev/ttyUSB0:telosb  env  java -jar javatest.jar -comm serial@/dev/ttyUSB0:telosb |

The fist command is used to set up the USB port for the serial transmission. With the second command, you check if previous command has been successfully executed. The last command will start the parser and you will see the following window:



Under the textbox “Connection” you enter the IP of the controller.

Under the textbox “Port to send to:” you enter the port you must transmit your data to.

Under the textbox “Port to listen to:” you enter the port you want to listen on for requests transmitted by the controller

# MySQL DataBase

This section describes how to set up your MySQL Server. Information about how to run the installer can be found on:

* <http://dev.mysql.com/doc/refman/5.0/en/installing.html>

It is recommended that you install MySQL Tools (MySQL Administrator, MySQL QueryBrowser and MySQL MigrationToolkit) as well to ease the use of the MySQL server.

You have to install the MySQL Connector/ODBC 3.51-driver as well. This is needed to connect to the server from the controller.

The installation files can be found in /Resources/Installers

# Restoring the Database

Before you can work with the database you have to import it into your MySQL server. This can be done via command line or via MySQL Administrator.

A backup of the entire database can be found in /Resources/Database

### Commandline

First you have to create the database in which you wish to restore the data. This can be done with the command:

mysql> create database if not exists <database name>;

with <database name> being the name that you wish to give to the database. You can then check if the database was successfully made with the command:

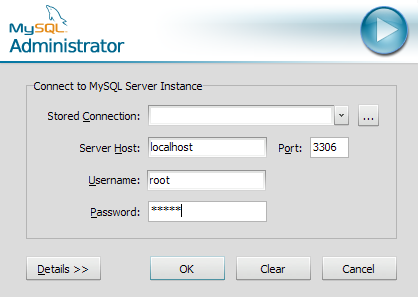
mysql> show databases;

Now you can import the data into your database with the command:

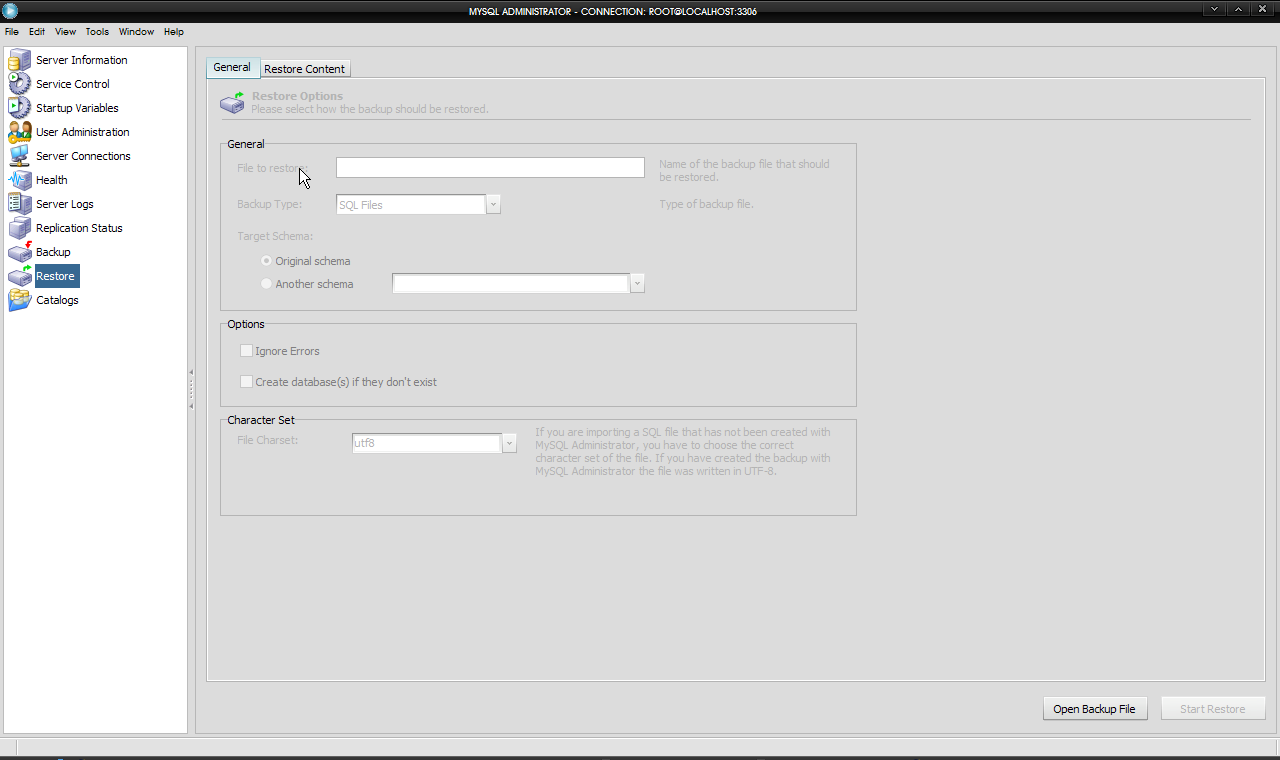
mysql -u root -p <databasename> < FinalDatabase 20090616.sql

with <databasename> again being the name of the database in which you wish to import the data. Make sure you have navigated to the folder where the backup is located!

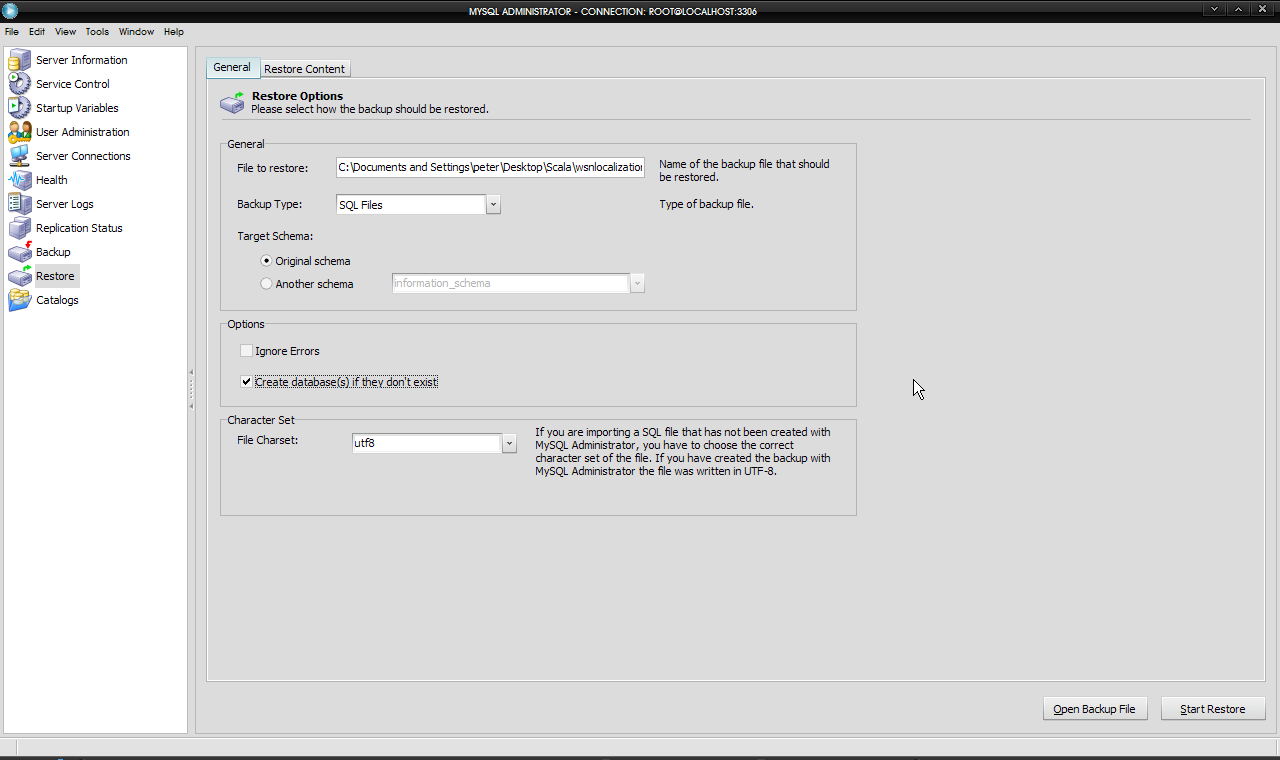
### MySQL Administrator

First, launch the MySQL Administrator tool. First you have to login to the server with the password and username that you created during the installation.

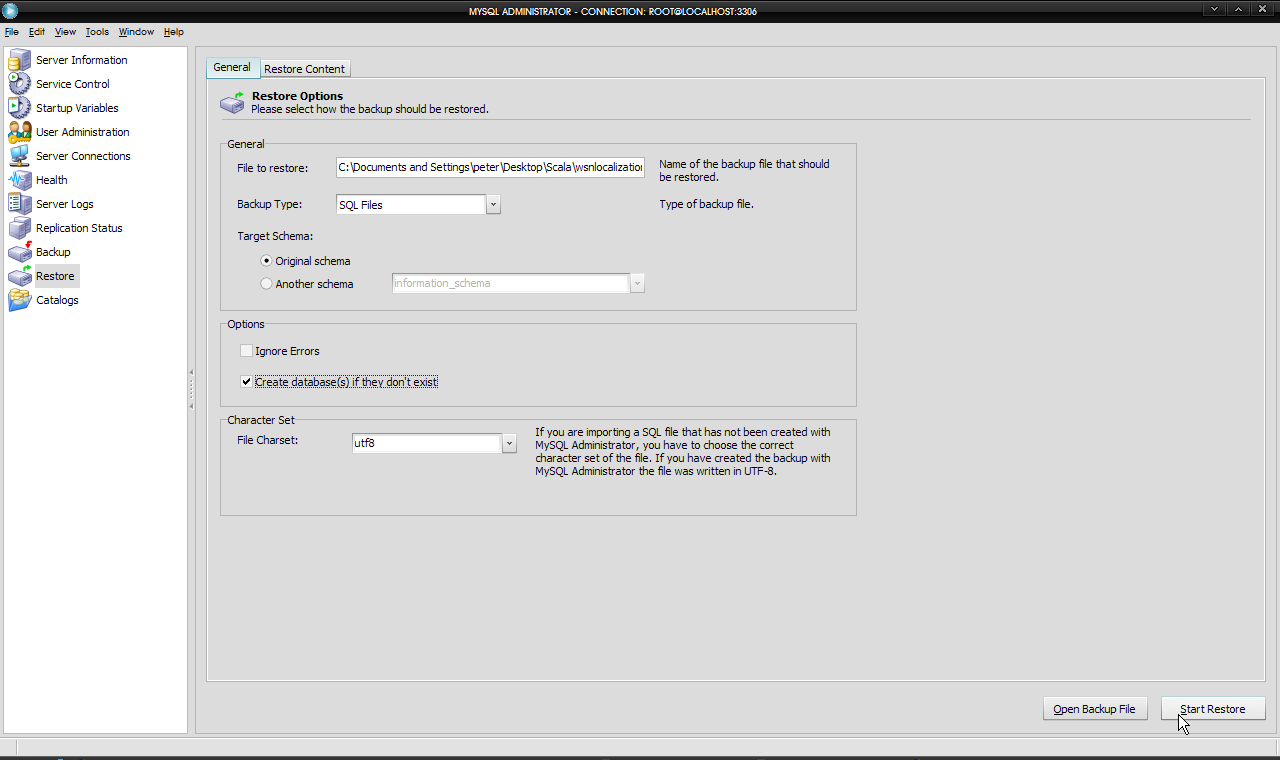
Then open up the backup tab by clicking on its icon on the left of the window. Press the Open Backup File button at the bottom of your screen. A Open File Dialog will open. Navigate towards the backup file of the database. This is found in the folder /Resources/Database on the cd of this master thesis



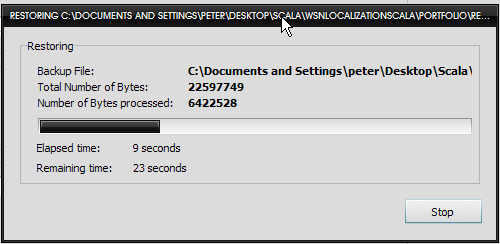
After opening the file you can edit the options on this tab. The only thing you have to do is check the box “Create database(s) if they don’t exist



You can now press the Start Restore button at the bottom right of your screen.



A dialog should show up showing the progress of the restore action. You can press the stop button at any time. If all is well in the world, no errors should occur.



Congratulations, you have successfully imported the database! You can now close the MySQL Administrator tool.

## Using the database

Again you can use the database via the command line or via a GUI tool. This time you should use MySQL Query Browser. We advise MySQL novices to use this tool. If however you want to use the command line you have the enter the following command before you can start using your database:

Mysql> use <database name>

With <database name> of course being the name of the database which you want to use.

# Controller & GUI

As the controller/engine & GUI form part of the same C# solution we will explain how to start them in this section. However these projects are not dependent upon each other in a strict sense. This means that you can start them without having to start the other. But, the GUI will do you little good if you do not start the controller because you then cannot access the rest of the network.

## Configuring the Controller

Starting the controller and GUI is extremely simple. But before you start the controller you have to set the correct parameters for network and database access in a configuration file.

Open the file config.txt in the folder /Final product/Build

<?xml version="1.0" encoding="utf-8" ?>

<config>

<SocketServer>

<Use>WSN</Use>

<Port>3800</Port>

</SocketServer>

<SocketServer>

<Use>GUI</Use>

<Port>1900</Port>

</SocketServer>

<SocketClient>

<Use>TelosB</Use>

<Port>3801</Port>

<HostName>10.22.41.81</HostName>

</SocketClient>

<SocketClient>

<Use>SunSpot</Use>

<Port>1802</Port>

<HostName>localhost</HostName>

</SocketClient>

<ConnectionString>

<ID>MySQL</ID>

<ConnString>DRIVER={MySQL ODBC 3.51 Driver};SERVER=localhost;DATABASE=senseless;UID=root;PASSWORD=admin;OPTION=3;</ConnString>

</ConnectionString>

</config>

We will now describe this file and what you should adjust to meet your personal environment.

<?xml version="1.0" encoding="utf-8" ?>

This line identifies the file as an XML file and identifies the text encoding.

Everything between the opening tag <config> and </config> identifies the configuration of the controller.

### Socketserver

<SocketServer>

<Use>WSN</Use>

<Port>3800</Port>

</SocketServer>

<SocketServer>

<Use>GUI</Use>

<Port>1900</Port>

</SocketServer>

These are the parameters for the ports on which the controller should listen for incoming messages. You should also make sure that the ports are not blocked by any firewall, including the firewall integrated into Windows.

### Socketclient

<SocketClient>

<Use>TelosB</Use>

<Port>3801</Port>

<HostName>10.22.41.81</HostName>

</SocketClient>

<SocketClient>

<Use>SunSpot</Use>

<Port>1802</Port>

<HostName>localhost</HostName>

</SocketClient>

These are the parameters for the addresses and ports which the controller should **send** to. There is one socketclient entry for each WSN. You will probably only use the TelosB SocketClient.

### Database connection

<ConnectionString>

<ID>MySQL</ID>

<ConnString>DRIVER={MySQL ODBC 3.51 Driver};SERVER=localhost;DATABASE=senseless;UID=root;PASSWORD=admin;OPTION=3;</ConnString>

</ConnectionString>

This section configures the ODBC connection to the MySQL database. Here you can change the IP address of the database (SERVER parameter), the default database (DATABASE parameter), the login and password (UID and PASSWORD parameters). You should the OPTION parameter as it is.

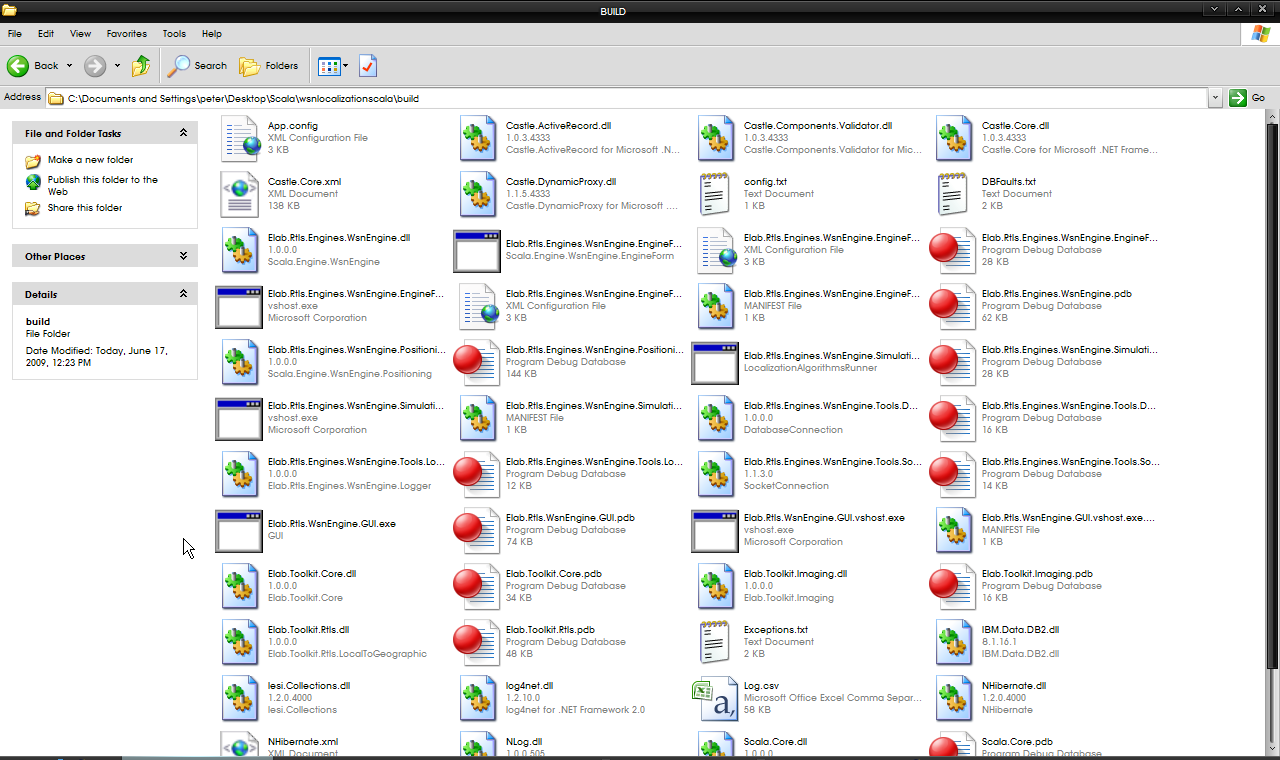
Congratulations, you should now have successfully configured the controller. You can now start the controller and GUI.

# Starting the Controller and GUI

This could not be any easier. First navigate to the folder /Final product/Build. In this folder you will find all the required executables, class libraries and configuration files. You may also find the debug files of Visual Studio (.pdb)

Then double click on:

* Elab.Rtls.Engines.WsnEngine.EngineForm.exe: to start the controller
* Elab.Rtls.WsnEngine.GUI.exe: to start the GUI



Simple isn’t it?

# Using the framework

For more information on how to use the framework, we refer you to the user manual.