j60870 User Guide

Table of Contents

1. Intro

j60870 is an implementation of the IEC 60870-5-104 protocol standard for client (i.e. master or controlling station) and server (i.e. slave or controlled station) applications.

You can use j60870 to program your individual client or server applications. A simple console client is part of the library. You can execute it using the scripts found in the folder "run-scripts".

2. Distribution

After extracting the distribution tar file the j60870 library can be found in the folder /build/libs.

3. Getting Started

3.1. Client

The easiest way to get started is by taking a look at the code of the console client which can be found here: src/main/java/org/openmuc/j60870/app/ConsoleClient.java . This application in combination with the javadoc should satisfy most of your documentation needs.

Here is a short summary of the steps to get a client running:

- Create and configure an instance of ClientConnectionBuilder.
- Connect to the server using ClientConnectionBuilder.connect() which returns the connection. The client is now connected to the server via TCP/IP.
- Initialize the data transfer by calling Connection.startDataTransfer.
- Now all incoming ASDUs (except for confirmation messages) will be forwarded to the ASduListener you registered. Every ASDU contains a number of Information Objects. The information objects contain information elements that make up the actual data. You will have to cast the InformationElements of the ASDU to a concrete implementation in order to access the data inside them. Every standardized Information Element is implemented by a class starting with the letters "Ie". The Type Identifier allows you to figure what to cast a particular Information Element to.
- You can use the Connection instance to send commands.

3.2. Server

The easiest way to get started programming an IEC 60870-5-104 server is by taking a look at the SampleServer located at: src/sample/java/SampleServer.java . The easiest way to run the sample server is by importing the project into Eclipse and run it from there. This is explained here: https://www.openmuc.org/faq/

4. Terminology

- **OA -** Originator Address
- **Monitor direction -** direction from server to client
- **Control direction -** direction from client to server
- **CON -** confirmation message
- **COT -** Cause of transmission
- **STARTDT ACT -** Start data tranfer message. Needs to be sent by the client before information messages may be exchanged between client and server.

5. Develop j60870

We use the Gradle build automation tool. The distribution contains a fully functional gradle build file ("build.gradle"). Thus if you changed code and want to rebuild a library you can do it easily with Gradle. Also if you want to import our software into Eclipse you can easily create Eclipse project files using Gradle. Just follow the instructions on our FAQ site: https://www.openmuc.org/faq/

6. Authors

Developers:

- Stefan Feuerhahn