train_rec

FQT

Team 4A

Installation

Requirements:
- A computer running Debian 7.8, this is the build computer.
- The following software packages installed
  - gcc/g++ >= 4.6
  - Qt >= 5.4
  - Qt-Creator >= 3.4
  - MYSQL driver for Qt
- Access to a mysql server with table structure agreed upon in class. Can be generated by the sql dump files on this team's github project. https://github.com/ctag/cpe453/tree/master/sql_loader
- Mysql-workbench

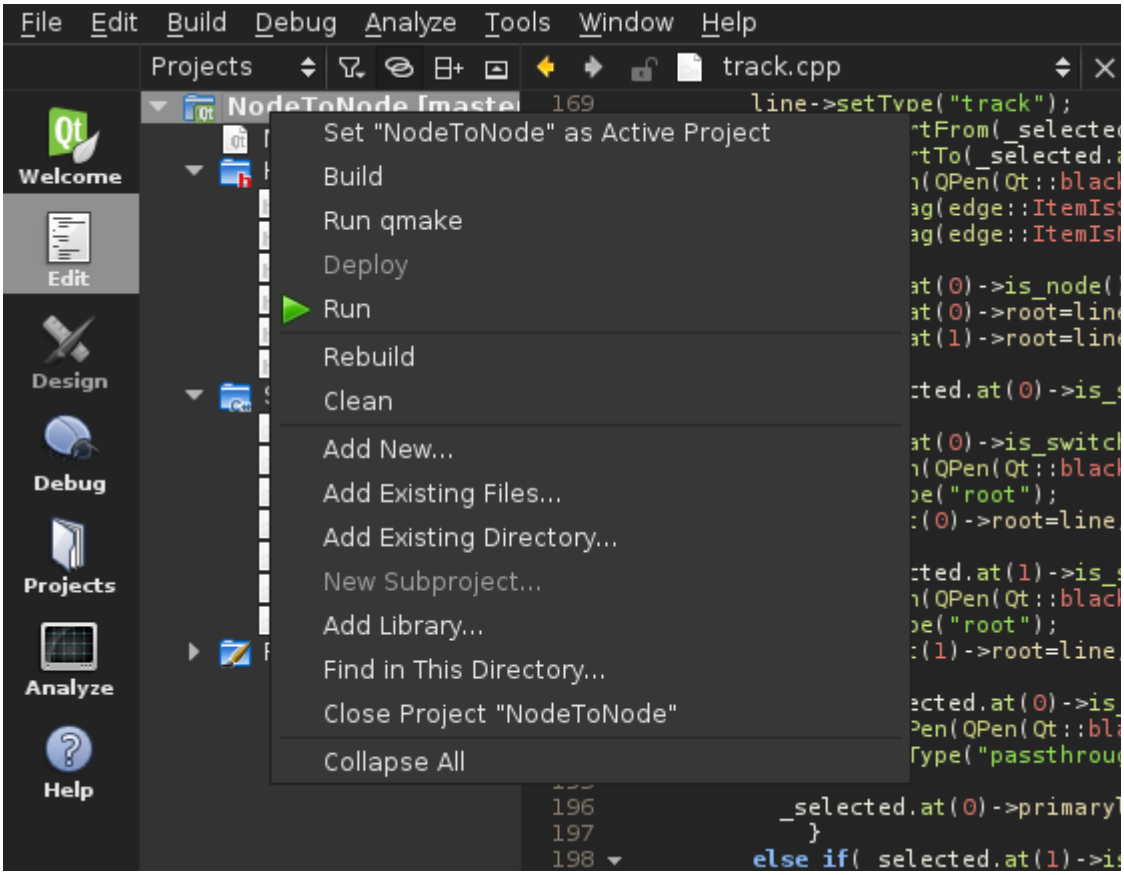
Note on SQL:
The above sql files are a standard for mySQL and their implementation to populate a server is separate from our project. A helper script is included with the .dump files for use with *ONLY* pavelow, to use it:
- Copy the script `clobber_server.sh` to the SQL*/ directory desired.
- Run the shell script with `./server_clobber.sh root cstrapwi 33153 cpe453`

Procedure to fetch source code

- Insert project CD.
- Navigate to CD files in Nautilus.
- Copy 'train_rec' and 'train_rec_sql' folders to user's home directory.
- Ensure that your user has proper permissions to access files in these folders.

Procedure to build train_rec

- Open the 'train_rec.pro' file from the 'train_rec' folder in your user's home directory with Qt-Creator. Do not have any other projects open in Qt-Creator.

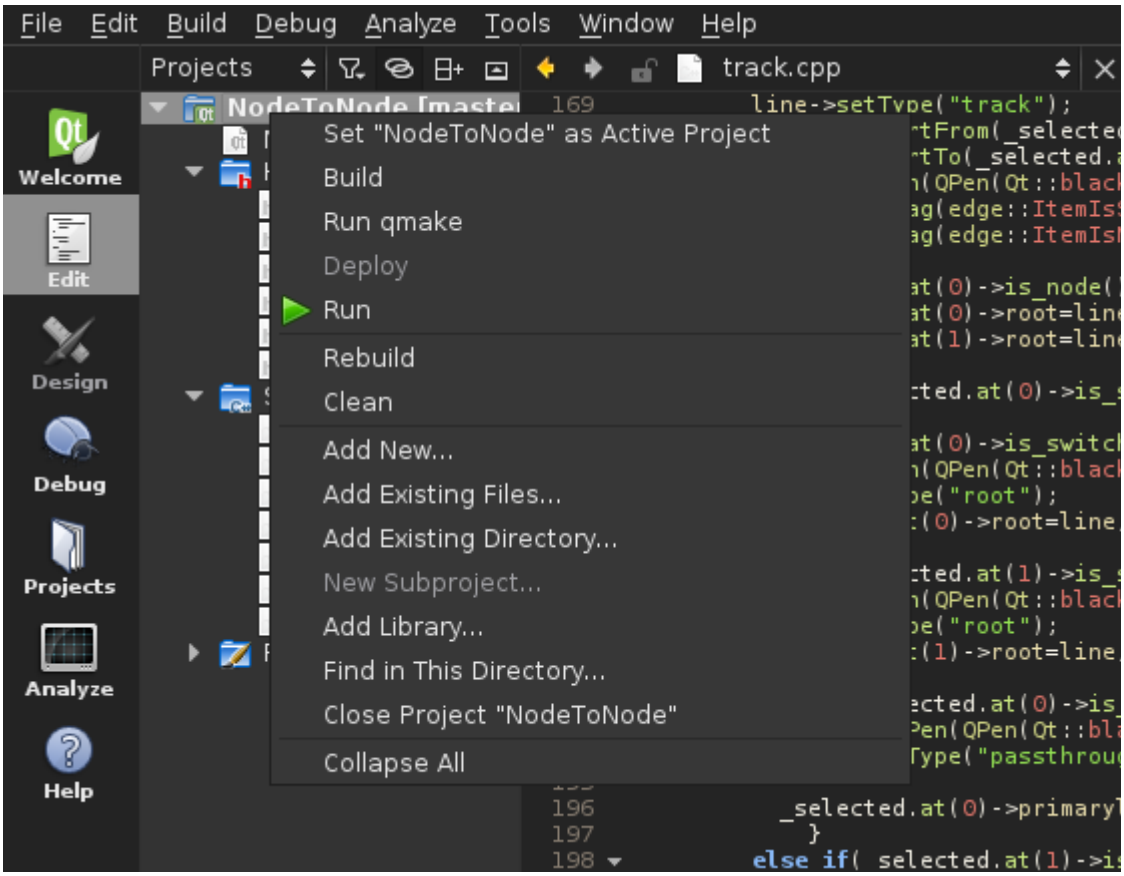


- See above figure for the next three items.
- Right click the project → click 'clean'.
- Right click the project → click 'Run qmake'.
- Right click the project → click 'Build'.

- You should now be able to run the program with Ctrl-r.

Procedure to build train_rec_sql

- Open the 'train_rec_sql.pro' file from the 'train_rec_sql' folder in your user's home directory with Qt-Creator. Do not have any other projects open in Qt-Creator.
- If train_rec is already open in Qt-Creator, right click on the project 'train_rec_sql' and click “Set “train_rec_sql” as active project”.



- See above figure for the next three items.
- Right click the project → click 'clean'.
- Right click the project → click 'Run qmake'.
- Right click the project → click 'Build'.

- You should now be able to run the program with Ctrl-r.

Testing

With train_rec and train_rec_sql running. Plug in the usb buffer to the computer.

Action: Click the 'refresh' button on 'Packets' tab.
Expected Response: Drop down list of usb ports is updated.
Pass/Fail: _____

Action: Select appropriate usb device from drop down list on 'Packets' tab.
Expected Response: The device is selected. Congratulations.
Pass/Fail: _____

Action: Click 'connect' on 'Packets' tab.
Expected Response: The program will attempt to connect to the usb device and report status on 'Console' tab.
Pass/Fail: _____

Action: Fill/verify fields on 'SQL' tab.
Expected Response: Self evident fields are filled.
Pass/Fail: _____

Action: Click 'connect' on 'SQL' tab.
Expected Response: The program will attempt to connect to the declared mySQL server. Status is logged to 'SQL' tab.
Pass/Fail: _____

Action: Run the mysql command “INSERT INTO `cpe453`.`req_macro` (`macro`, `arg1`) VALUES ('SLOT_SCAN', '3');” on the SQL server.
Expected Response: The program will delete the row and send a packet `BB0300XX` to loconet.
Pass/Fail: _____

Action: Run the mysql command “INSERT INTO `cpe453`.`req_macro` (`macro`, `arg1`) VALUES ('SLOT_DISPATCH', '3');” on the SQL server.
Expected Response: The program will delete the row and send a packet `BA0300XX` to loconet.
Pass/Fail: _____

Action: Run the mysql command “INSERT INTO `cpe453`.`req_macro` (`macro`, `arg1`) VALUES ('SLOT_CLEAR', '3');” on the SQL server.
Expected Response: The program will delete the row and send a packet `B50300XX` to loconet.
Pass/Fail: _____

Action: Run the mysql command “INSERT INTO `cpe453`.`req_macro` (`macro`, `arg1`) VALUES ('SLOT_REQ', '1');” on the SQL server.
Expected Response: The program will delete the row and send a packet `BF0001XX` to loconet.
Pass/Fail: _____

Action: Run the mysql command “INSERT INTO `cpe453`.`req_macro` (`macro`) VALUES ('TRACK_RESET');” on the SQL server.
Expected Response: The program will delete the row and send a packet `82XX` and then `83XX` to loconet.
Pass/Fail: _____

Action: Run the mysql command “INSERT INTO `cpe453`.`req_macro` (`macro`) VALUES ('TRACK_OFF');” on the SQL server.
Expected Response: The program will delete the row and send a packet `82XX` to loconet.
Pass/Fail: _____

Action: Run the mysql command “INSERT INTO `cpe453`.`req_macro` (`macro`) VALUES ('TRACK_ON');” on the SQL server.
Expected Response: The program will delete the row and send a packet `83XX` to loconet.
Pass/Fail: _____

Action: With track powered on (see above macro), run a train across two detection sections which are listed in the 'track_ds' table of the mySQL server.
Response: The table will update to show a '1' in the status field of the occupied detection section. This can be seen via the train_rec_sql program on the 'Blocks' tab with the list detection sections button.

Action: Click 'Disconnect' on 'Packets'.
Response: The program will close the usb connection.

Action: Click 'Disconnect' on 'SQL'.
Response: The program will close the SQL connection.

That's it, all requirements from a loconet bridge. Switch status is not returned because it is not available. Extra features in the train_rec suite, including train_rec_sql, are provided as-is.