mipt-smb-search

Yulyugin Evgeny (yulyugin@gmail.com)
Morgen Matvey (melges.morgen@gmail.com)
and others.

May 26, 2013

1 Description

Project to write search engine for mipt campus network with video preview. Our web-cite http://code.google.com/p/mipt-smb-search/.

SVN: https://mipt-smb-search.googlecode.com/svn/trunk

2 Dependencies

- g++-4.7
- libmysql++
- libsmbclient
- libprocps
- libcppunit
- libmagic

3 How to build

3.1 Build with Qt-Creator

When you open Qt-Creaotr at first you can see window like fig. 1. Clear "Shadow build".

Release setting should look like fig 2(a). Debug setting should look like fig 2(b).

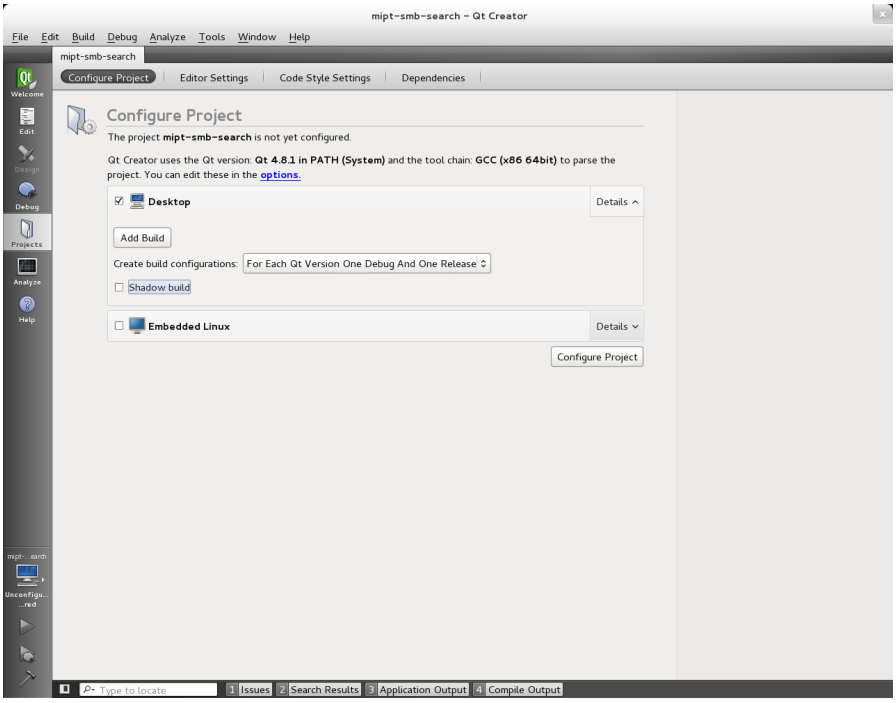


Figure 1: Qt-Creator settings main.

3.2 Build without Qt-Creator

By default project will build in svn_dir/build/release in release mode and svn_dir/build/debug in debug mode.

```
$ make install
```

To build in debug mode you should use next command:

```
$ make DEBUG=yes install
```

4 How to run tests

4.1 Without Qt-Creator

```
$ ./mipt-smb-search/tests/testing.sh <path_to_build_directory>
```

4.2 With Qt-Creator

This is an example of Qt-Creator settings to run fulltest. This settings should be done after build settings (look at section 3 for more details).

1. First of all you should choose pass to executable file. In our example it's situated in %buildDir/build/debug/tests/bin/fulltest

2. Than add path to working directory. Full test should executes from directory when it situates. In our example it's situated in %buildDir/build/debug/tests/bin

3. You should set LD_LIBRARY_PATH variable to detect new shared libraries which created in our project. You should add path to lib and tests/lib folder. LD_LIBRARY_PATH should contain full path to this folder.

The result of this settings could be shown in fig. 3
You can configure run settings for each executable file based on this algorithm.

5 Coading style

We use google coading style for more details see "Google C++ Style Guide" (http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml) for C++ codes.

6 How to commit changes

Before commit your changes you should pass code review.

To send your patch on code review you should use upload.py script (https://codereview.appspot.com/static/upload.py).

```
$ python2.7 upload.py -m "My important change description."
```

For more imformation about usage and option of upload.py see http://code.google.com/p/rietveld/wiki/UploadPyUsage.

Then go to https://codereview.appspot.com/, choose your subject.
Reviewers: yulyugin@gmail.com, melges.morgen@gmail.com
CC: mipt-smb-search-review@googlegroups.com
Description: Description of your changes and testlog.
Than start codereview review.
After "LGTM" you can commit your changes.

```
svn commit -m "My importand change description."
```

7 How to write tests

8 How to test coverage

1. Build with TEST_COVERAGE flag:

   ```
   $ make TEST_COVERAGE=yes
   ```

2. Run tests. For more information see section 4

3. Go to the directory with tested file.

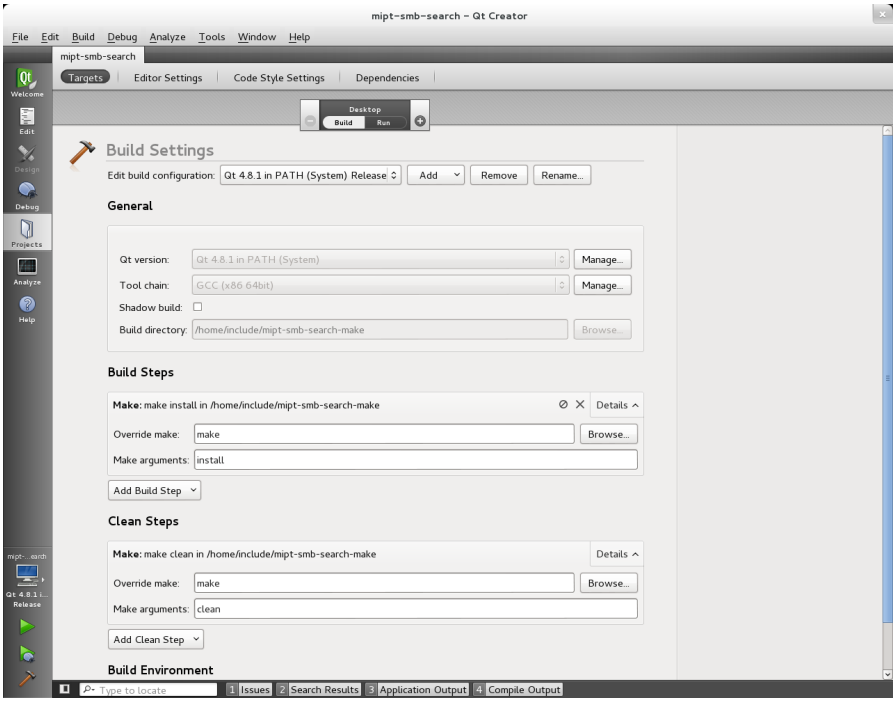
   ```
   $ cd mipt-smb-serach/spider
   ```

4. Test coverage with gcov programm

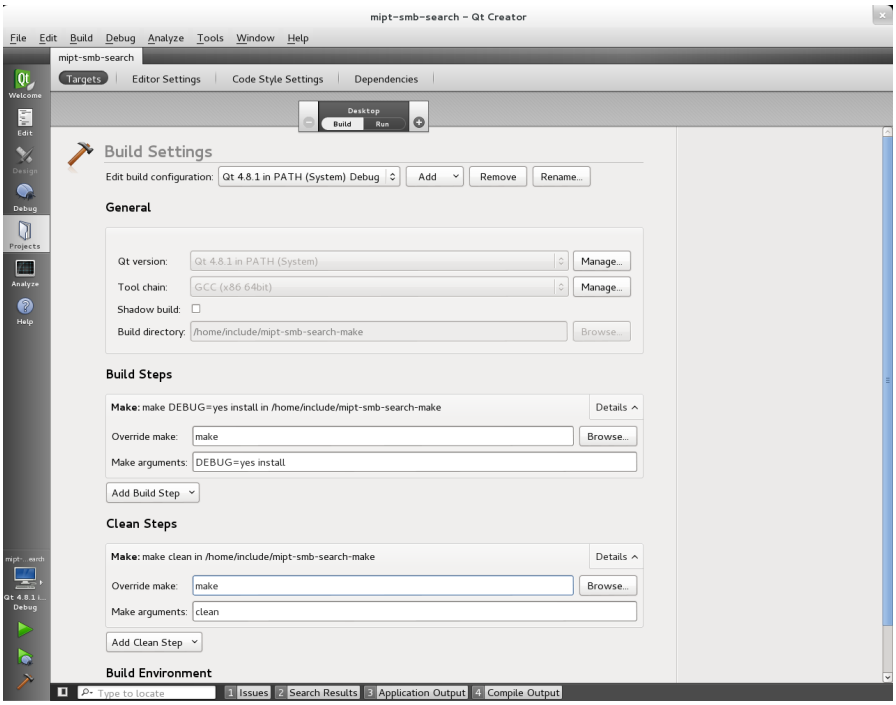
```
$ gcov spider.gcda
File 'mipt-smb-search/spider/spider.cpp'
Lines executed:55.23% of 239
Creating 'spider.cpp.gcov'
```

5. You can see information about what lines and how often been executed.

```
$ nano spider.cpp.gcov
```



(a)



(b)

Figure 2: Build settings for Qt-Creator: (a) for release build, (b) for debug build.

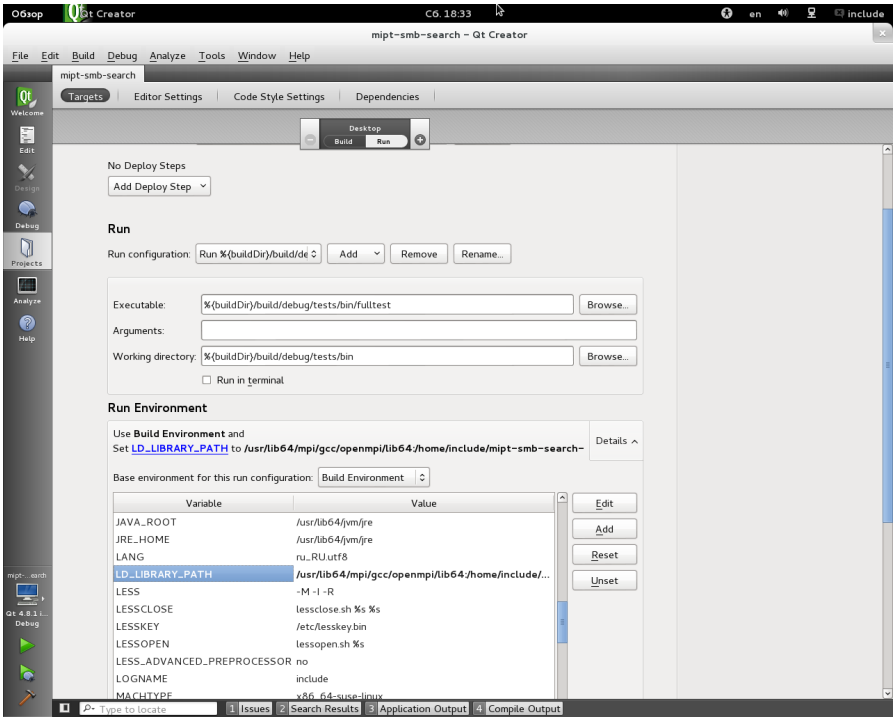


Figure 3: Qt-Creator settings to run.