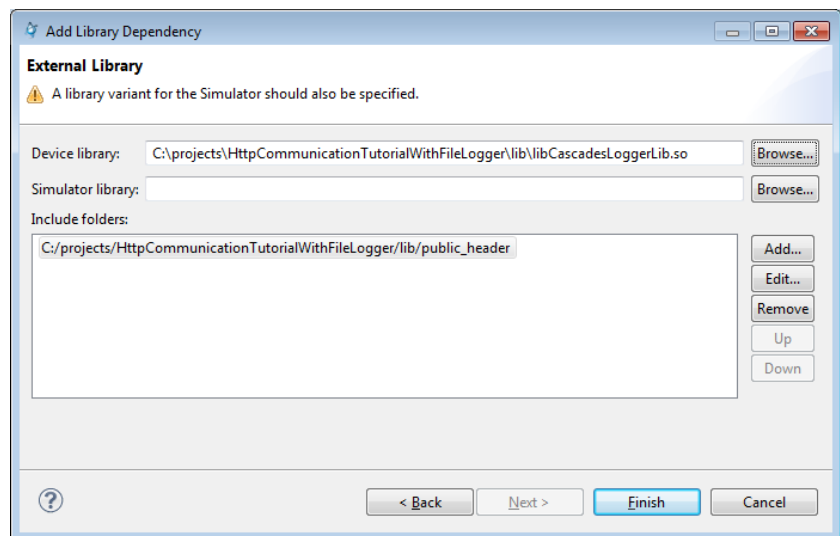
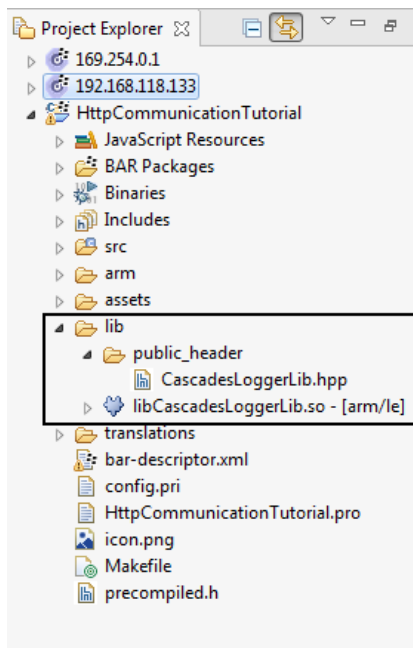


The Penrillian logging library v1.0

The Penrillian logging library allows BB10 apps to write log statements to the file system. This will assist in debugging apps away from the development environment. This has been tested with SDK 10.1 and is thread safe.

Instructions

Copy `libCascadesLoggerLib.so` and `CascadesLoggerLib.hpp` into a folder in the project's (the project which will be using the logger) file structure. Using the IDE (v10.2 is used in this example), right click the project, select "Configure" then "Add Library". Choose "External Library" then select `libCascadesLoggerLib.so` for the "Device Library" and the folder containing `CascadesLoggerLib.hpp` as the "Include Folder".

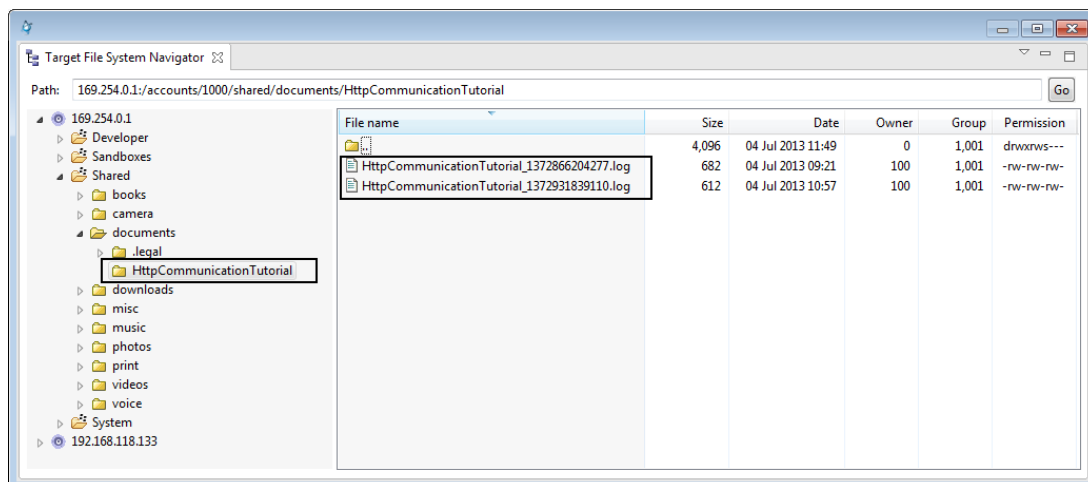
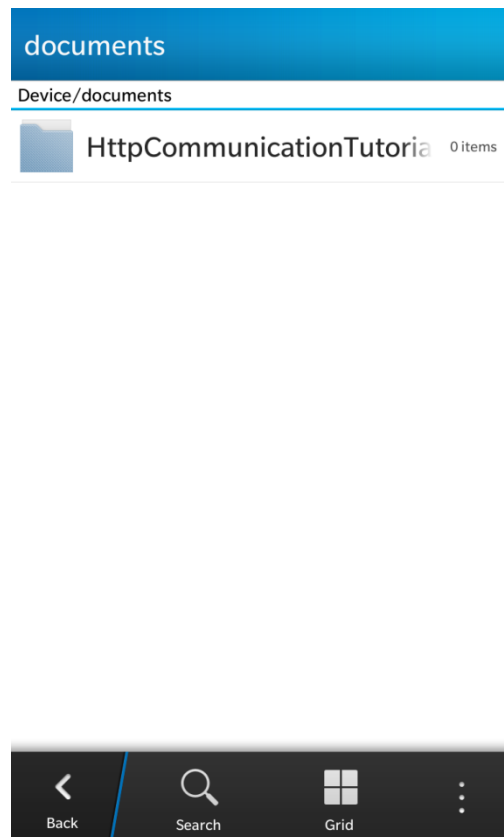
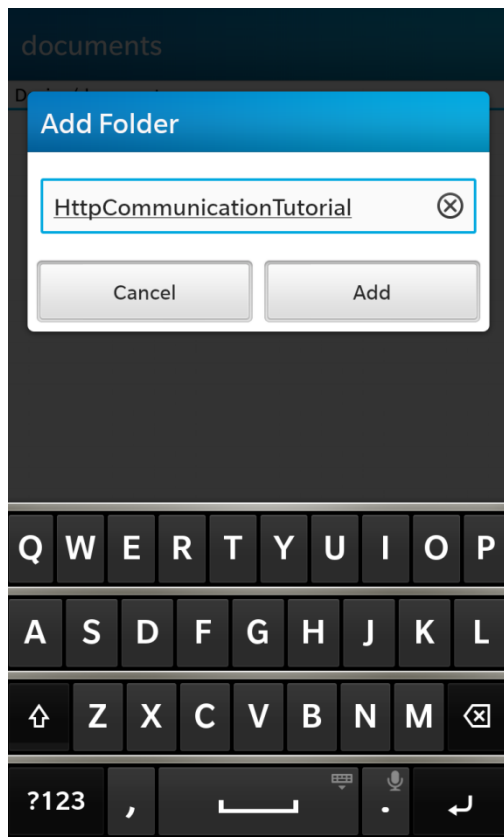


To direct all `qDebug()` (C++) and `console.debug()` (QML) statements to the console and file system:

- in `main.cpp`, add `#include "CascadesLoggerLib.hpp"`
- in the `main()` method, add `CascadesLoggerLib::init()`

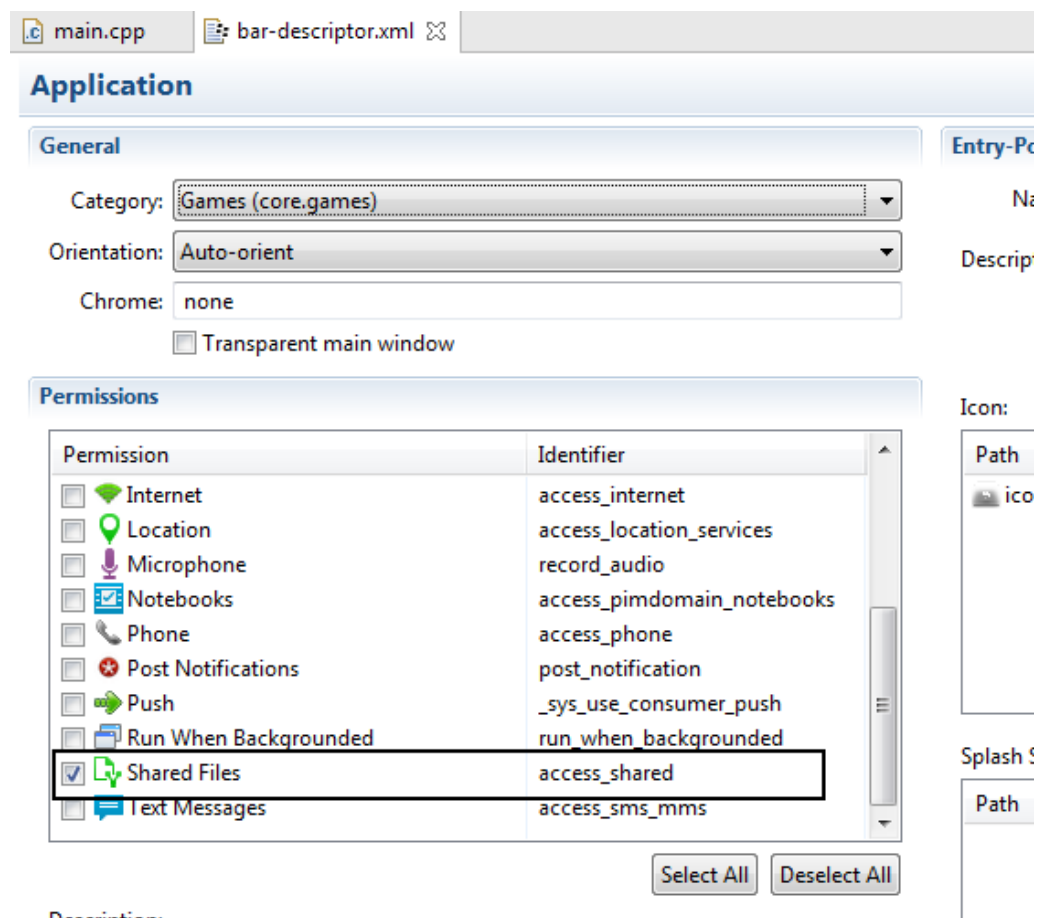
```
main.cpp
1 #include <bb/cascades/Application>
2 #include <QLocale>
3 #include <QTranslator>
4 #include "applicationui.hpp"
5 #include "CascadesLoggerLib.hpp"
6
7 using namespace bb::cascades;
8
9 Q_DECL_EXPORT int main(int argc, char **argv) {
10     CascadesLoggerLib::init();
11     Application app(argc, argv);
12
13     QTranslator translator;
14     QString locale_string = QLocale().name();
15     QString filename = QString("HttpCommunicationTutorial_%1").arg(
16         locale_string);
17     if (translator.load(filename, "app/native/qm")) {
18         app.installTranslator(&translator);
19     }
20
21     new ApplicationUI(&app);
22
23     return Application::exec();
24 }
25
```

This will automatically enable logging to console. To also enable logging to the file system, add a folder with the name of the app to /shared/documents.



Note:

In order to write into the shared directory it is required to add the “access_shared” permission (Shared Files) to the project.



To perform ad-hoc logging:

- add `#include "CascadesLoggerLib.hpp"` to the .cpp file whether logging is to be performed
- call `CascadesLoggerLib::sharedInstance()->log("logging statement")` when required
- an optional tag can also be included to make reading the log file easier:
 - `CascadesLoggerLib::sharedInstance()->log("logging statement", "A TAG")`

```

void ApplicationUI::requestFinished(QNetworkReply* reply) {

    const QString NETWORK_ERROR_TAG = "NETWORK_ERROR";

    qDebug() << "requestFinished";

    myActivityIndicator->stop();
    myLabel->setVisible(false);

    if (reply->error() != QNetworkReply::NoError) {
        if (!myFile->open(QIODevice::ReadWrite)) {
            return;
        }
        myFile->write(reply->readAll());
        myFile->flush();
        myFile->close();

        XmlDataModel *dataModel = new XmlDataModel();
        dataModel->setSource(QUrl("file://" + QDir::homePath() + "/model.xml"));

        myListView->setDataModel(dataModel);
    } else {

        CascadesLoggerLib::sharedInstance()->log(reply->errorString(), NETWORK_ERROR_TAG);

        myLabel->setText("Problem with the network");
    }

    reply->deleteLater();
}

```

The result:

```

main.cpp applicationui.cpp HttpCommunicationTutorial_1... HttpCommunicationTutorial_1... HttpCommunicationTutorial_1...
1 Thu Jul 4 12:03:02 2013
2 =====
3
4 Start of log...
5 12:03:02:432 OrientationSupportPrivate::getSupportedDisplayOrientation: auto
6 12:03:02:437 OrientationSupportPrivate::initDirectionAndOrientation:
7 isCardOrViewer: false
8 manifestDisplayOrientation: All
9 displayOrientation: All
10 uiOrientation: Portrait
11 displayDirection: North
12 nativeDisplayOrientation: Portrait
13 12:03:02:453 CS Debugging enabled: false
14 12:03:02:511 ApplicationUi Constructor
15 12:03:02:540 initiateRequest
16 12:03:02:610 PpsTimeSettingsObject::newSettingsData() emitted
17 12:03:06:013 requestFinished
18 12:03:06:016 NETWORK_ERROR Unknown error
19

```

Note:

To disable logging, simply delete the folder with the app's name from /shared/documents.