

EMAC OE SDK Qt/Eclipse Integration

EMAC provides customers with the option of development support for Qt 4.6 in its embedded products. This guide provides a simple procedure for setting up and using the standard Qt demonstration projects in Eclipse followed by a more general guide to starting a new Makefile Qt project in Eclipse using the EMAC OE SDK toolchain.

About Qt/qmake

Qt provides a build tool called qmake to simplify the build process for development projects across different platforms. It generates Makefiles for a project from only a few lines of information provided using .pro and .qrc files. To complement these features EMAC provides an OE SDK for Qt complete with libraries, include files, toolchain, and example build scripts which demonstrate how to set up a Qt cross build for EMAC products. In addition, EMAC provides an Eclipse startup script which sources the Qt variables necessary to help qmake generate cross-compilation Makefiles which refer to the aforementioned OE SDK during the build process.

For customers who receive EMAC Linux Development Computers it is not necessary to follow the Setup section as this is all done for you. Do take note however that the EMAC Eclipse distribution is located at /home/developer/eclipse and the EMAC OE SDK is located in the default Eclipse workspace at /home/developer/workspace/EMAC-OE-arm-linux-gnueabi-SDK_4.0.

Setup

To begin developing Embedded Qt applications in the EMAC Eclipse distribution, first download the EMAC OE SDK Qt Addon from the EMAC FTP server and extract it into the default Eclipse workspace as shown in the Installation and Configuration guides for the EMAC OE SDK. EMAC recommends Eclipse be installed in the user's home directory and that the SDK be installed in the default Eclipse workspace.

Starting Eclipse for Qt Development

In order to provide qmake with all the necessary environment variables, Eclipse is started from a shell script found in the top level of the EMAC OE SDK directory named eclipse-qt-startup.sh. This shell script must be modified to indicate the correct path to the Eclipse executable on the development system as well as the correct path to the EMAC OE SDK. The following code block demonstrates which variables should be modified using the EMAC Linux Development Computer (LDC) setup as an example.

```

ECLIPSE=/home/developer/eclipse/eclipse
export SDKBASE=/home/developer/workspace/EMAC-OE-arm-linux-gnueabi-SDK_4.0

```

Import the EMAC OE Qt SDK Into Eclipse

Once the startup script is configured correctly for the current development system, run it to start an Eclipse session. Once it has started up, follow the Eclipse EMAC OE SDK Import guide with the only difference for this instance being that the Qt-specific rather than the standard version should be imported. In addition to this, there is one other important step to preparing the Eclipse workspace which is to ensure that the GNU Make Builder is specified as the C/C++ Builder and to create an External Tool Configuration for qmake.

Setting the C/C++ Builder

1. In the *Project Explorer View* right-click on the *EMAC-OE-arm-linux-gnueabi-SDK_4.0* project and select *Properties* from the context menu.
2. Select *C/C++ Builder* → *Tool Chain Editor* in the *Properties Filter* on the left side of the resulting *Properties* window.
3. Select *Gnu Make Builder* in the *Current builder*: drop-down menu if it is not already selected.

qmake External Tool Configuration

1. In the Eclipse Workspace select *Run* → *External Tools* → *External Tools Configurations*.
2. Right-click *Program* in the External Tools Filter on the left side of the resulting *External Tools Configurations* window, then select *New* from the context menu.
3. It is recommended to change the name of the resulting configuration to something like qmake2 to more easily identify it later.
4. Click *Browse Workspace...* in the *Location:* frame.
5. Select *EMAC-OE-arm-linux-gnueabi-SDK_4.0* → *gcc-4.2.4-arm-linux-gnueabi* → *i686-linux* from the resulting *Choose Location* dialog.
6. Select qmake2 as the binary for this External Tool.
7. Click *Browse Workspace...* in the *Working Directory:* frame.
8. Select *EMAC-OE-arm-linux-gnueabi-SDK_4.0* → *projects* → *qt_demos* from the resulting *Folder Selection* dialog.
9. In the Arguments field, type the following string:

```
demos.pro QT_LIBINFIX=E
```

This, along with the Working Directory specified above, tell the qmake2 executable to use the workspace_loc/EMAC-OE-arm-linux-gnueabi-SDK_4.0/projects/qt_demos/demos.pro file to create Makefiles for the project.

Creating Eclipse Make Targets

For the Qt demonstration projects, it is important to set up Make Targets according to Generate Eclipse Make Targets section of the Eclipse New Project Guide. Note that the specific Make Targets shown in this guide are specific to the specific Makefile under consideration in that guide. In order to create Make Targets for the current example, use that guide as an example for the following make targets:

- **qmake.** Causes GNU Make to recursively call qmake2 on all the subdirectories in the qt_demos project folder. This produces a Makefile from the .pro file in each subdirectory.
- **all.** Causes GNU Make to recursively call make on the Makefiles in the subdirectories of the qt_demos project folder. This produces the cross-compiled binaries for all the qt_demos sub-projects as well as all the intermediate dependency and object files.
- **clean.** Causes GNU Make to recursively call make clean on all the subdirectories in the qt_demos project folder. This removes all the intermediate dependency and object files produced by the make call but leaves the binaries.
- **mrproper.** Causes GNU Make to recursively call make mrproper on all the subdirectories in the qt_demos project folder. This removes all the files produced by the make call, including the binaries.
- **distclean.** Causes GNU Make to recursively remove all files produced by the make call as well as those produced by qmake2. After running this make target, it will be necessary to run the qmake2 external tool again in order to use these Make Targets.

Create a New Qt Project in Eclipse

The process for creating a new Qt Project is the same as that for creating a new C++ Project in Eclipse as described on the Eclipse New Project Guide. EMAC recommends creating the new project within the /workspace_loc/EMAC-OE-arm-linux-gnueabi-SDK_4.0/projects directory although it is possible to create a new project outside of the EMAC-OE-arm-linux-gnueabi-SDK_4.0 project. Additional steps are necessary since this new project will also need to accommodate the qmake2 External Tool and creating Make Targets.

qmake External Tool Configuration

Creating the qmake2 External Tool Configuration is similar to what is described in the preceding section, but different enough to deserve its own step-by-step process. The primary difference is that for a new project, it is

necessary to write a new .pro file which is beyond the scope of this guide. EMAC recommends reading the Qt 4.6 documentation (<http://doc.qt.nokia.com/4.6/index.html>) with special attention for the qmake Tutorial (<http://doc.qt.nokia.com/4.6/qmake-tutorial.html>) and the qmake Manual (<http://doc.qt.nokia.com/4.6/qmake-manual.html>) . In the step-by-step process enumerated below, note that the only difference between the Qt Demonstration projects and this is that both qt_demos and demos have been replaced by new_project where "new_project" can actually be any name the developer wishes to use. For example, demos.pro has been replaced by new_project.pro.

1. In the Eclipse Workspace select *Run → External Tools → External Tools Configurations*.
2. Right-click *Program* in the External Tools Filter on the left side of the resulting *External Tools Configurations* window, then select *New* from the context menu.
3. It is recommended to change the name of the resulting configuration to something like qmake2 to more easily identify it later.
4. Click *Browse Workspace..* in the *Location:* frame.
5. Select *EMAC-OE-arm-linux-gnueabi-SDK-4.0 → gcc-4.2.4-arm-linux-gnueabi → i686-linux* from the resulting *Choose Location* dialog.
6. Select qmake2 as the binary for this External Tool.
7. Click *Browse Workspace..* in the *Working Directory:* frame.
8. Select *EMAC-OE-arm-linux-gnueabi-SDK-4.0 → projects → new_project* from the resulting *Folder Selection* dialog.
9. In the Arguments field, type the following string:

```
new_project.pro QT_LIBINFIX=E
```

This, along with the Working Directory specified above, tell the qmake2 executable to use the workspace_loc/EMAC-OE-arm-linux-gnueabi-SDK-4.0/projects/new_project/new_project.pro file to create Makefiles for the project.

Creating Eclipse Make Targets for a New Project

Creating Make Targets is the same as what is described in the preceding Creating Eclipse Make Targets subsection.

Building the Qt Demo Projects

To actually build the Qt demo projects once everything is set up is a simple procedure. Note that the make targets specified below are found in the Make Targets View.

1. Run the qmake2 External Tool. To do this, select *Run → External Tools → qmake2* in the Eclipse workspace. As described above, this creates the Makefiles with all the rules necessary to build the project.
2. Run the clean Make Target. To do this, double-click *EMAC-OE-arm-linux-gnueabi-SDK-4.0 → projects → qt_demos → clean*. This step is not necessary, all it really does is remove any dependency and object files produced from previous invocations of make. It is included here to demonstrate its use in this example build.
3. Run the all Make Target. To do this, double-click *EMAC-OE-arm-linux-gnueabi-SDK-4.0 → projects → qt_demos → all*. This step produces the binaries which will be sent to the target machine.

Running the Qt Demo Projects

Once the Qt demonstration project binaries have built successfully, they can be uploaded to the remote target using the Eclipse Remote System Explorer Plugin. The process for setting this up and using the Remote Systems View to transfer files is described on the following EMAC Wiki pages:

- Remote System Explorer Configuration
 - RSE Setup
 - RSE SFTP Setup

■ Remote Shell/Terminal Setup

The RSE Setup page describes how to begin using the Remote System Explorer, the RSE SFTP Setup page describes how to use Drag-and-Drop file transfer between two hosts, and the Remote Shell/Terminal Setup guide describes how to open a remote shell inside a running instance of Eclipse. The following steps will assume a basic familiarity with the techniques described in these guides and will not describe the step-by-step procedure of using the Eclipse GUI for transferring files.

1. Choose a demo sub-project binary to transfer to the target machine. A Good example to start with is the EMAC-0E-arm-linux-gnueabi-SDK_4.0/projects/qt_demos/books/books binary.
2. Copy the file from localhost:/workspace_loc/EMAC-0E-arm-linux-gnueabi-SDK_4.0/projects/qt_demos/books/books to remote_target_host:/path/.
3. To run the binary, type the following commands:

```
$ export QWS_MOUSE_PROTO="tslib:/dev/input/touchscreen0"
$ /path/books -qws &
```

This starts the books application as a background process in its controlling terminal.

4. Note that the Books demo project takes up the full screen. To stop the books application, type the following command at the controlling terminal:

```
$ kill %1
```

Next Steps

Once the Qt demonstration projects have been successfully built, Qt applications can be developed for EMAC Open Embedded products.

See Also

- Linux Development Computer
 - Getting Started
 - EMAC Eclipse IDE
 - Making Connections
 - Installing Packages

» packages » linux_start » eclipse

-
- linux/ldc/eclipse.txt · Last modified: 2011/05/12 15:50 by wwarren
 - Except where otherwise noted, content on this wiki is licensed under the following license: CC Attribution-No Derivative Works 3.0 Unported (cc-by-nd)