

## Qt embedded 配置选项

我的 qt-everywhere-opensource-src-4.6.0.tar.gz 是在主文件夹中.就是 你  
\$cd ~ 然后那个位置,接下来解压这个包包.  
~\$tar zxvf qt-everywhere-opensource-src-4.6.0.tar.gz  
~\$cd qt-everywhere-opensource-src-4.6.0  
接下来就应该 configure 了.现在开始配置参数了  
.configure -prefix /usr/local/qte-arm -debug-and-release -qt3support -qt-zlib -qt-libtiff -qt-libpng  
-qt-libmng -qt-libjpeg -make libs -nomake examples -nomake demos -nomake docs -no-cups  
-iconv -xplatform qws/linux-arm-g++ -embedded arm -little-endian -qt-freetype -depths  
8,16,24,32 -qt-gfx-linuxfb -no-gfx-transformed -no-gfx-qvfb -no-gfx-vnc -no-gfx-multiscreen  
-no-dbus -qt-sql-sqlite -qt-kbd-qvfb -qt-kbd-tty -qt-mouse-pc -no-glib -plugin-mouse-tslib  
-I/usr/local/tslib/include -L/usr/local/tslib/lib -D\_\_ARM\_ARCH\_5TEJ\_\_  
这个参数要全部输入进去复制+粘贴,你懂的哈.  
但是你要作一下修改的啊,因为我们的已其实不同的阿.  
先给你些解释:(configure 后面参数的意义)  
.configure  
-prefix /usr/local/qte-arm (强制安装在此路径)  
-debug-and-release 编译和链接两个版本的 Qt  
-qt3support 支持 QT3  
-qt-zlib 捆绑使用的 Qt 的 zlib。  
-qt-libtiff 使用 Qt 的捆绑的 libtiff。  
-qt-libpng 使用 Qt 的捆绑的的 libpng。  
-qt-libmng 使用 Qt 的捆绑的 libmng。  
-qt-libjpeg 使用 Qt 的捆绑的的 libjpeg。  
-make libs 新增 libs  
-nomake examples 排除 ..  
-nomake demos 排除..  
-nomake docs 排除。  
-no-cups 不要编译 CUPS 的支  
-iconv 编译 iconv 拥有支持 (3)  
-xplatform qws/linux-arm-g++  
-embedded arm  
-little-endian  
-qt-freetype  
-depths 8,16,24,32  
-qt-gfx-linuxfb  
-no-gfx-transformed  
-no-gfx-qvfb  
-no-gfx-vnc  
-no-gfx-multiscreen  
-no-dbus  
-qt-sql-sqlite

-qt-kbd-qvfb  
-qt-kbd-tty  
-qt-mouse-pc  
-no-glib  
-plugin-mouse-tslib -I/usr/local/tslib/ -L/usr/local/tslib/lib -D\_\_ARM\_ARCH\_5TEJ\_\_  
(包含路径) (库路径) (定义的预处理器)

注意拉:

-plugin-mouse-tslib 是说你需要用到 tslib 这个插件.(我理解的好像不太准确吧, 呵呵不要笑话我水平有限)

-I/usr/local/tslib/lib 这个是你 tslib.h 这个头文件的包含路径

-L/usr/local/tslib/lib 这个是你的库的路径

-D\_\_ARM\_ARCH\_5TEJ\_\_ 这个是定义的预处理器,唉百度找出来的,不太理解那几个字木的含义哈.

(如果你的 configure 一直出错误那么就看最下面那一大部分东西,然后找有用的自己慢慢理解哈.)

help: unknown argument 帮助: 未知参数

Usage: configure [-h] [-prefix <dir>] [-prefix-install] [-bindir <dir>] [-libdir <dir>]

用法: [-docdir <dir>] [-headerdir <dir>] [-plugindir <dir>] [-datadir <dir>]

[-translationdir <dir>] [-sysconfdir <dir>] [-examplesdir <dir>]

[-demosdir <dir>] [-buildkey <key>] [-release] [-debug]

[-debug-and-release] [-developer-build] [-shared] [-static] [-no-fast] [-fast]

[-no-largefile] **www.linuxidc.com** [-largefile] [-no-exceptions] [-exceptions] [-no-accessibility]

[-accessibility] [-no-stl] [-stl] [-no-sql-<driver>] [-sql-<driver>]

[-plugin-sql-<driver>] [-system-sqlite] [-no-qt3support] [-qt3support]

[-platform] [-D <string>] [-I <string>] [-L <string>] [-help]

[-qt-zlib] [-system-zlib] [-no-gif] [-qt-gif] [-no-libtiff] [-qt-libtiff] [-system-libtiff]

[-no-libpng] [-qt-libpng] [-system-libpng] [-no-libmng] [-qt-libmng]

[-system-libmng] [-no-libjpeg] [-qt-libjpeg] [-system-libjpeg] [-make <part>]

[-nomake <part>] [-R <string>] [-l <string>] [-no-rpath] [-rpath] [-continue]

[-verbose] [-v] [-silent] [-no-nis] [-nis] [-no-cups] [-cups] [-no-iconv]

[-iconv] [-no-pch] [-pch] [-no-dbus] [-dbus] [-dbus-linked]

[-no-separate-debug-info] [-no-mm] [-no-3dnow] [-no-sse] [-no-sse2]

[-qtnamespace <namespace>] [-qtlabinfix <infix>] [-separate-debug-info] [-armfpa]

[-no-optimized-qmake] [-optimized-qmake] [-no-xmlpatterns] [-xmlpatterns]

[-no-multimedia] [-multimedia] [-no-phonon] [-phonon] [-no-phonon-backend]

[-phonon-backend]

[-no-audio-backend] [-audio-backend] [-no-openssl] [-openssl] [-openssl-linked]

[-no-gtkstyle] [-gtkstyle] [-no-svg] [-svg] [-no-webkit] [-webkit] [-no-javascript-jit]

[-javascript-jit]

[-no-script] [-script] [-no-scripttools] [-scripttools] [-no-declarative] [-declarative]

[additional platform specific options (see below)] #额外的平台特定的选项 (见下文)

Installation options: 安装选项:

These are optional, but you may specify install directories.

这些都是可选的，但您可以指定安装目录。

-prefix <dir> ..... This will install everything relative to <dir>

这一切都将安装相对于<dir>

(default /usr/local/Trolltech/Qt-4.6.2)

\* -prefix-install .... Force a sandboxed "local" installation of

强制 a sandboxed "local"安装 QT。

Qt. This will install into

这将安装到

/usr/local/Trolltech/Qt-4.6.2, if this option is

如果此选项

disabled then some platforms will attempt a

然后禁用某些平台将尝试

"system" install by placing default values to

“系统”通过将安装默认值

be placed in a system location other than

被放置在一个系统以外的其他位置

PREFIX.

字首。

You may use these to separate different parts of the install:

你可以使用这些分开的安装不同的部分：

-bindir <dir> ..... Executables will be installed to <dir>

可执行文件将被安装到<目录>

(default PREFIX/bin)

( 默认 PREFIX/bin)

-libdir <dir> ..... Libraries will be installed to <dir>

Libraries (库) 将被安装到<目录>

(default PREFIX/lib)

默认 PREFIX/lib

-docdir <dir> ..... Documentation will be installed to <dir>

文件 (使用说明) 将被安装到<目录>

(default PREFIX/doc)

-headerdir <dir> ..... Headers will be installed to <dir>

标题。头文件？

(default PREFIX/include)

-plugindir <dir> ..... Plugins will be installed to <dir>

插件

(default PREFIX/plugins)

-datadir <dir> ..... Data used by Qt programs will be installed to <dir>

QT 程序所使用的数据

(default PREFIX)

-translationdir <dir> . Translations of Qt programs will be installed to <dir>

翻译 QT 的程序

(default PREFIX/translations)

-sysconfdir <dir> ..... Settings used by Qt programs will be looked for in <dir>

QT 程序的使用设置

(default PREFIX/etc/settings)

-examplesdir <dir> .... Examples will be installed to <dir>

范例

(default PREFIX/examples)

-demosdir <dir> .... Demos will be installed to <dir>

演示

(default PREFIX/demos)

You may use these options to turn on strict plugin loading.

你可以使用这些选项打开严格装入插件。

-buildkey <key> .... Build the Qt library and plugins using the specified

<key>. - buildkey 的<key> ....生成 Qt 库和插件使用指定

<key>. When the library loads plugins, it will only

的<key>。当库加载插件，它只会

load those that have a matching key.

负载那些具有匹配的关键。

Configure options:

配置选项:

The defaults (\*) are usually acceptable. A plus (+) denotes a default value

that needs to be evaluated. If the evaluation succeeds, the feature is

included. Here is a short explanation of each option:

默认值 (\*) 通常都是可以接受的一个加号 (+) 表示默认值

需要进行评估。如果评估成功，功能是

包括在内。下面是每个选项的简短说明:

\* -release ..... Compile and link Qt with debugging turned off.

关闭调试 QT 的编译和连接

-debug ..... Compile and link Qt with debugging turned on.

打开调试 QT 的编译和连接

-debug-and-release . Compile and link two versions of Qt, with and without

编译和链接两个版本的 Qt

debugging turned on (Mac only).

并且不进行调试 (仅在 MAC 系统中)。

-developer-build.... Compile and link Qt with Qt developer options (including auto-tests exporting)

编译和链接与 Qt 的开发选项 (包括自动测试出口的 Qt)

-opensource..... Compile and link the Open-Source Edition of Qt.

编译和链接的 Qt 开源版

-commercial..... Compile and link the Commercial Edition of Qt.

编译和链接的 Qt 商业版。

\* -shared ..... Create and use shared Qt libraries.

创建和使用共享 Qt 库。

-static ..... Create and use static Qt libraries.

创建和使用静态的 Qt 库。

\* -no-fast ..... Configure Qt normally by generating Makefiles for all  
通过生成所有的 Makefile  
project files.

项目文件正常配置 Qt。

-fast ..... Configure Qt quickly by generating Makefiles only for  
通过配置 Qt 的快速生成 Makefile 文件只以  
library and subdirectory targets. All other Makefiles  
library and subdirectory (子目录) 为目标, 所有其它的 Makefile  
are created as wrappers, which will in turn run qmake.

封装创建, 浙江反过来运行 qmake。

-no-largefile ..... Disables large file support.

禁用大文件支持

+ -largefile ..... Enables Qt to access files larger than 4 GB.

使 Qt 来访问大于 4 GB 的档案

-no-exceptions ..... Disable exceptions on compilers that support it.

在编译器, 支持禁用例外。

(例外: 没有受到防火墙, 程序, 服务, 或者端口的阻挡的, 自由进出的数据包。)  
\* -exceptions ..... Enable exceptions on compilers that support it.

在编译器, 支持启用例外。

-no-accessibility .. Do not compile Accessibility support.

不编辑可接入性支持

\* -accessibility ..... Compile Accessibility support.

编辑可接入性支持

-no-stl ..... Do not compile STL support.

不要编译 STL 的支持。

\* -stl ..... Compile STL support.

要编译 STL 的支持。

(STL 现在是 C++的一部分, 因此不用额外安装什么。它被内建在你的编译器之内。)

在 C++ 标准中, STL 被组织为下面的 13 个头文件: <algorithm>、<deque>、  
<functional>、<iterator>、<vector>、<list>、<map>、<memory>、  
<numeric>、<queue>、<set>、<stack> 和<utility>。)

-no-sql-<driver> ... Disable SQL <driver> entirely.

完全禁用 SQL 驱动

-qt-sql-<driver> ... Enable a SQL <driver> in the QSql library, by default

在 QSql 库启用一个 SQL <驱动>, 默认情况下

none are turned on.

没有一个是开启

-plugin-sql-<driver> Enable SQL <driver> as a plugin to be linked to

启用 SQL <驱动> 作为一个插件挂钩

at run time.

在运行时。

Possible values for <driver>:

为<驱动>可能值:

[ db2 ibase mysql oci odbc psql sqlite sqlite2 sqlite\_symbian tds ]

-system-sqlite ..... Use sqlite from the operating system.  
使用 sqlite 的从操作系统

-no-qt3support ..... Disables the Qt 3 support functionality.  
禁用的 Qt 3 支持的功能

\* -qt3support ..... Enables the Qt 3 support functionality.  
启用了 Qt 3 支持的功能。

-no-xmlpatterns .... Do not build the QtXmlPatterns module.  
不要建立 QtXmlPatterns 模块。

+ -xmlpatterns ..... Build the QtXmlPatterns module.  
构建 QtXmlPatterns 模块。  
QtXmlPatterns is built if a decent C++ compiler  
QtXmlPatterns 如果建立一个像样的 C ++ 编译器  
is used and exceptions are enabled.  
例外的是使用和启用

-no-multimedia ..... Do not build the QtMultimedia module.  
不要建立 QtMultimedia 模块

+ -multimedia ..... Build the QtMultimedia module.  
构建 QtMultimedia 模块。

-no-audio-backend .. Do not build the platform audio backend into QtMultimedia.  
不要建立平台的音频后端到 QtMultimedia。

+ -audio-backend ..... Build the platform audio backend into QtMultimedia if available.  
建立在平台的音频后端到 QtMultimedia 如果可用。

-no-phonon ..... Do not build the Phonon module.  
不要建立声子模块。

+ -phonon ..... Build the Phonon module.  
建立声子模块。  
Phonon is built if a decent C++ compiler is used.  
如果声子是建立一个像样的 C ++ 编译器使用。

-no-phonon-backend.. Do not build the platform phonon plugin.  
不要建立在平台声子插件。

+ -phonon-backend..... Build the platform phonon plugin.  
建立在平台声子插件。

-no-svg ..... Do not build the SVG module.  
不要建立 SVG 的模块。

+ -svg ..... Build the SVG module.  
建立 SVG 的模块  
(SVG: 静止无功发生器)

-no-webkit ..... Do not build the WebKit module.  
不要建立 WebKit 的模块

+ -webkit ..... Build the WebKit module.  
建立 WebKit 的模块  
WebKit is built if a decent C++ compiler is used.  
WebKit 是如果建立一个像样的 C ++ 编译器使用。

-no-javascript-jit . Do not build the JavaScriptCore JIT compiler.

不要建立 JavaScriptCore 的 JIT 编译器。

+ -javascript-jit .... Build the JavaScriptCore JIT compiler.

建立 JavaScriptCore 的 JIT 编译器。

-no-script ..... Do not build the QtScript module.

不要建设作为 QtScript 模块

+ -script ..... Build the QtScript module.

建设作为 QtScript 模块

(QtScript:QT 脚本)

-no-scripttools .... Do not build the QtScriptTools module.

不建立 QT 脚本工具模块

+ -scripttools ..... Build the QtScriptTools module.

建立 QT 脚本工具模块

+ -no-declarative .....Do not build the declarative module.

不要建立的声明模块

-declarative ..... Build the declarative module.

建立的声明模块

-platform target ... The operating system and compiler you are building

操作系统和编译器你是建立在 (与 linux - g ++的)。

on (linux-g++).

See the README file for a list of supported

查看支持的操作系统和编译器列表中的自述文件。

operating systems and compilers.

-graphicssystem <sys> Sets an alternate graphics system. Available options are

设置一个备用系统的图形。可用的选项有

raster - Software rasterizer

软件光栅

opengl - Rendering via OpenGL, Experimental!

通过 OpenGL 渲染，实验！

-no-mmx ..... Do not compile with use of MMX instructions.

没有编制与 MMX 指令使用。

-no-3dnow ..... Do not compile with use of 3DNOW instructions.

没有编制与 3DNOW 指令使用。

-no-sse ..... Do not compile with use of SSE instructions.

没有编制与 SSE 指令使用。

-no-sse2 ..... Do not compile with use of SSE2 instructions.

没有编制与 SSE2 指令使用。

-qtnamespace <name> Wraps all Qt library code in 'namespace <name> {...}'.

隐藏所有的 Qt'命名空间<名称 {...}'.库代码

-qtlibprefix <infix> Renames all libQt\*.so to libQt\*<infix>.so.

重命名所有的 libqt \*.所谓的 libqt \* <infix>。如此。

-D <string> ..... Add an explicit define to the preprocessor.

添加一个明确的定义的预处理器。

-I <string> ..... Add an explicit include path.

添加一个明确的包含路径。

**-L <string>** ..... Add an explicit library path.

添加一个明确的库路径

**-help, -h** ..... Display this information.

显示此信息

Third Party Libraries:

第三方库：

**-qt-zlib** ..... Use the zlib bundled with Qt.

捆绑使用的 Qt 的 zlib。

+ **-system-zlib** ..... Use zlib from the operating system.

使用 zlib 的从操作系统

See <http://www.gzip.org/zlib>

详情 <http://www.gzip.org/zlib>

**-no-gif** ..... Do not compile the plugin for GIF reading support.

不要编译的 GIF 阅读支持插件

\* **-qt-gif** ..... Compile the plugin for GIF reading support.

编译的 GIF 阅读支持插件。

See also src/plugins/imageformats/gif/qgifhandler.h

**-no-libtiff** ..... Do not compile the plugin for TIFF support.

不要编译为 TIFF 的支持插件

**-qt-libtiff** ..... Use the libtiff bundled with Qt.

使用 Qt 的捆绑的 libtiff。

+ **-system-libtiff** .... Use libtiff from the operating system.

使用的 libtiff 从操作系统。

See <http://www.libtiff.org>

**-no-libpng** ..... Do not compile in PNG support.

不要编译 PNG 支持。

**-qt-libpng** ..... Use the libpng bundled with Qt.

使用 Qt 的捆绑的的 libpng。

+ **-system-libpng** .... Use libpng from the operating system.

使用的 libpng 从操作系统。

See <http://www.libpng.org/pub/png>

**-no-libmng** ..... Do not compile the plugin for MNG support.

不编译 MNG 支持插件。

**-qt-libmng** ..... Use the libmng bundled with Qt.

使用 Qt 的捆绑的 libmng。

+ **-system-libmng** .... Use libmng from the operating system.

使用 libmng 从操作系统。

See <http://www.libmng.com>

**-no-libjpeg** ..... Do not compile the plugin for JPEG support.

不要编译为 JPEG 支持插件

**-qt-libjpeg** ..... Use the libjpeg bundled with Qt.

使用 Qt 的捆绑的的 libjpeg。

+ **-system-libjpeg** .... Use libjpeg from the operating system.

使用的 libjpeg 从操作系统。

See <http://www.ijg.org>

-no-openssl ..... Do not compile support for OpenSSL.

没有编制 OpenSSL 的支持。

+ -openssl ..... Enable run-time OpenSSL support.

启用运行时 OpenSSL 的支援。

-openssl-linked .... Enabled linked OpenSSL support.

启用与 OpenSSL 的支持

-ptmalloc ..... Override the system memory allocator with ptmalloc.

重写与 ptmalloc 系统内存分配器。

(Experimental.)

(实验)

Additional options:

其他选项:

-make <part> ..... Add part to the list of parts to be built at make time.

新增部分的零件清单，建在腾出时间

(libs tools examples demos docs translations)

(例如诱导探测工具，演示文档翻译)

-nomake <part> ..... Exclude part from the list of parts to be built.

排除从零件清单的一部分，才是上策。

-R <string> ..... Add an explicit runtime library path to the Qt

添加一个明确的运行时库路径的 Qt 库

libraries.

-l <string> ..... Add an explicit library.

添加一个明确的库

-no-rpath ..... Do not use the library install path as a runtime

不要使用库安装为一个运行时库路径的路径。

library path.

+ -rpath ..... Link Qt libraries and executables using the library  
install path as a runtime library path. Equivalent

to -R install\_libpath

Qt 库和可执行文件链接使用库安装路径为运行时库的路径。

当量对 -R 的 install\_libpath

-continue ..... Continue as far as possible if an error occurs.

如果发生错误就继续

-verbose, -v ..... Print verbose information about each step of the  
打印有关的配置过程中的每个步骤详细的信息。

configure process.

-silent ..... Reduce the build output so that warnings and errors  
减少产量，以便建立警告和错误可以看到更容易

can be seen more easily.

\* -no-optimized-qmake ... Do not build qmake optimized.

不建 qmake 的优化。

-optimized-qmake ..... Build qmake optimized.

建立 qmake 的优化

-no-nis ..... Do not compile NIS support.  
没有编制 NIS 支持。  
\* -nis ..... Compile NIS support.  
编制 NIS 支持。

-no-cups ..... Do not compile CUPS support.  
不要编译 CUPS 的支持  
\* -cups ..... Compile CUPS support.  
编译 CUPS 的支持  
Requires cups/cups.h and libcups.so.2.  
需求 cups/cups.h 和 libcups.so.2.

-no-iconv ..... Do not compile support for iconv(3).  
不要编译 iconv 拥有支持 (3)。  
\* -iconv ..... Compile support for iconv(3).  
编译 iconv 拥有支持 (3)。

-no-pch ..... Do not use precompiled header support.  
不使用预编译头的支持。  
\* -pch ..... Use precompiled header support.  
使用预编译头的支持。

-no-dbus ..... Do not compile the QtDBus module.  
不要编译 QtDBus 模块。  
+ -dbus ..... Compile the QtDBus module and dynamically load libdbus-1.  
编译 QtDBus 模块和动态加载的 libdbus - 1。  
-dbus-linked ..... Compile the QtDBus module and link to libdbus-1.  
编译并链接到 QtDBus 模块的 libdbus - 1。

-reduce-relocations ..... Reduce relocations in the libraries through extra  
linker optimizations (Qt/X11 and Qt for Embedded Linux only;  
experimental; needs GNU ld >= 2.18).  
减少库重定向，通过减少额外 连接器优化  
(Qt/X11 和 Qt 的嵌入式 Linux 只; 实验性; 需要的 GNU ld 的 >= 2.18)。

-no-separate-debug-info . Do not store debug information in a separate file.  
不要调试信息存储在一个单独的文件。  
\* -separate-debug-info .... Strip debug information into a separate .debug file.  
调试到一个单独的信息。调试文件。