BlackBerry Jam Americas

Build and Packaging for Native Apps

JAM39

Roberto Speranza (@RSSessantotto)
Taso Perdikoulias

May 14-16, 2013

Overview

- Introduction to the BlackBerry10 NDK build system
- Using third party components and shared libraries
- Release packaging
- Creating an effective build systems which saves developers time
- Supporting multiple operating systems and devices

BlackBerry Jam Americas

Introduction to the BlackBerry10 NDK build system

The qmake build system

The Cascades project template

The Qmake Build System

From the Qt Project

BlackBerry Jam Americas

- Generates Makefiles
- Project configuration in the .pro file
- Supports features (.prf)
  - Same format as .pro file
  - Extensible – make your own
  - Features can have options
- Scopes allow conditional code
- Simple functions available

Commonly Used Qmake Variables

CONFIG	TEMPLATE
QMAKE_CFLAGS	QMAKE_CXXFLAGS
INCLUDEPATH	QMAKE_LFLAGS
DEFINES	LIBS

```
device {
 CONFIG(debug, debug|release) {
   DEFINES += $$(envvar)
 }
}
device:profile:LIBS += -
   lprofilingS
```

The Cascades Project Template 1

BlackBerry Jam Americas

- Top-level Makefile with each build type as a target
  - Includes a release mode package target

```
$ make Device-Debug
$ make package
```

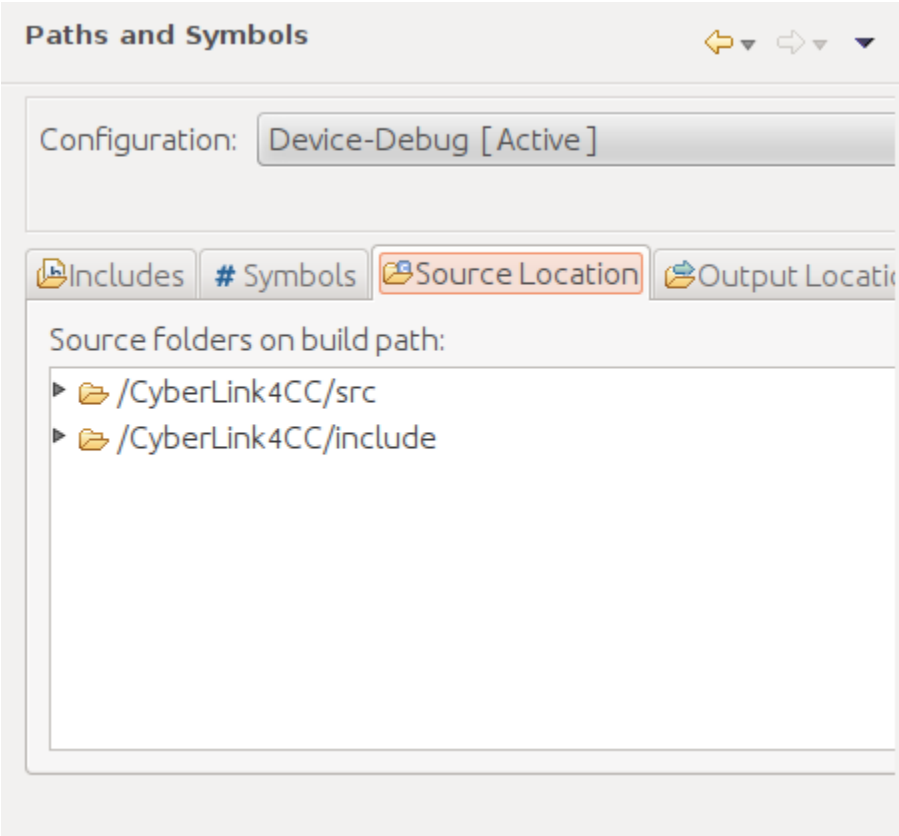
- Main project file (.pro)
  - Common scopes for customization

```
APP_NAME = MyApplication
TEMPLATE = app

CONFIG += qt warn_on cascades10

include(config.pri)

device {
  CONFIG(debug, debug|release) {
    # Device-Debug custom config
  }
  CONFIG(release, debug|release) {
    # Device-Release custom config
  }
}
```

The Cascades Project Template 2

BlackBerry Jam Americas

- Momentics managed qmake file config.pri
  - Automatically sets source, header and translation variables
- Configuration must be set for each project
  - Configuration: all does not work (known bug)



BlackBerry Jam Americas

Using third party libraries and shared components

Building Cascades and Qt library projects

Using third party libraries

Packaging external assets

Building Cascades and Qt Libaries

- Modify the cascades application template .pro file
  - Set TEMPLATE = lib
  - Include dependent includes with INCLUDES += <paths>
  - Link dependent libraries with LIBS += -l<library>
- Modify the Momentics C++ Build settings to include only the relevant sources and includes
  - Filter a folder with patterns or file lists
- Create a .prf file for easy inclusion in other projects
- Full example on github.com/blackberry
  - Also, see the cascades10.prf file shipped with the NDK

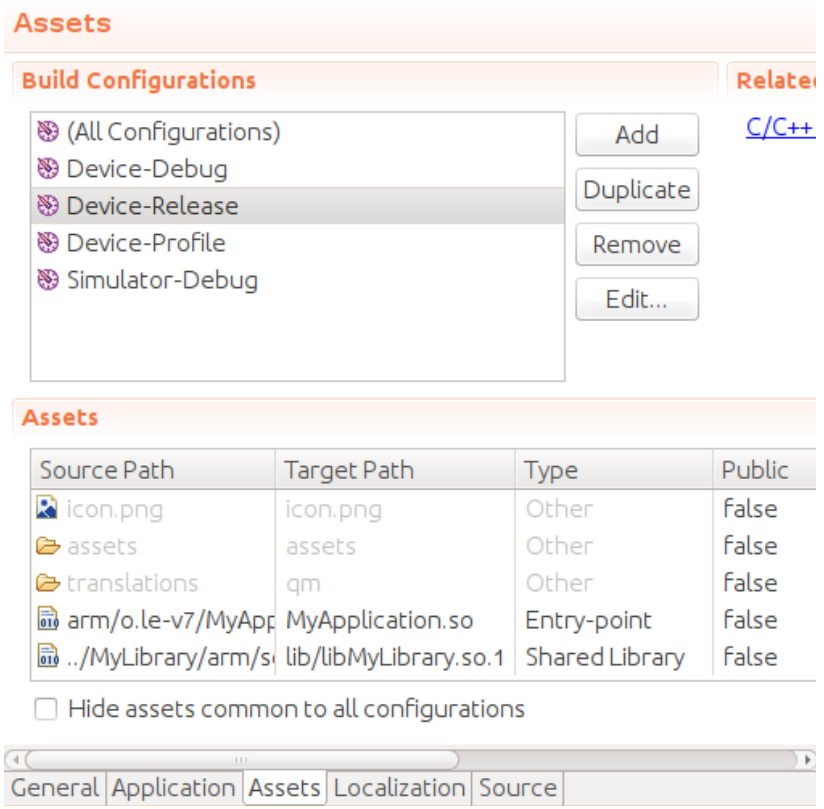
Using Third Party Libraries

- Momentics has an Add Library wizard to automate simple cases
- If the library has .prf files, set the path in the QMAKEFEATURES environment variable
- Modify the your application's .pro file
  - Include library includes with INCLUDES += <library include path>
  - Link the library with LIBS += -l<library>
- Add the library to your bar descriptor as Qnx/Elf asset
- If the library has QML assets, make sure to include them as assets in the application bar descriptor

Packaging External Assets

BlackBerry Jam Americas

- Assets in the bar descriptor are any file you wish to include in your package
- Assets can be configured per build type, ie: debug or release
- Libraries must be marked as type Qnx/Elf
- Warning: Momentics will use special variables for “workspace” assets



BlackBerry Jam Americas

Release Packaging

Dealing with large resources

Tweaking the release build

Complier and Linker Defenses

Dealing with large resources

BlackBerry Jam Americas

- Some Carriers limit application size for cellular download to as little as 50 MB
- For OpenGL applications, use multiple packages if you include compressed or resolution specific textures
- Include compressed resources and decompress on first run
- Dynamically load resources from the internet when needed
- BlackBerry10 has native decompression libraries such as zlib

Tweaking the Release Build 1

- Use a release selector in your .pro file
- Cascades10 feature define a few release specific flags
  - Optimize for size with `-Os`
  - Stack smashing protector with `-fstack-protector-strong`
  - Fortify-source with `-DFORTIFY_SOURCE=2`
  - Release mode stripping
  - Reduces load time with `-fvisibility=hidden`
  - Read-only relocation sections with `-Wl,-z,relro`
- Qcc supports common gcc options directly
  - Other options must be passed using specific compiler mode flags

    `-Wp, -Wc, -Wl, -Wa`

Tweaking the Release Build 2

- Enable link-time optimizations for additional speed and size optimizations
  - Should not be used with debug mode (-g flag)

```
QMAKE_CFLAGS += -flto
QMAKE_LFLAGS += -flto -fuse-linker-plugin
```

- Strip non-zygote or pure native apps

```
QMAKE_POST_LINK += $${QMAKE_STRIP} -strip-all "$@"
```

- Customize the creation of separate debug symbols
  - Release mode with debug symbols for full postmortem debugging
  - Stripping debug symbols from debug mode executable to reduce debug build size and deploy time

Compiler and Linker Defenses

- Enable extra format string warnings as errors

  `-Wformat-security -Werror=format-security`
- Warn on trampolines, which generate extra vulnerabilities

  `-Wtrampolines`
- Apply stack smashing protection to smaller buffers

  `-Wc,--param=ssp-buffer-size=4`
- Apply the stack protector to all functions

  `-fstack-protector-all`
- Stack checking is currently broken
  - Incorrect code produced when using the `-fstack-check` option

BlackBerry Jam Americas

Creating an effective build systems which saves developers time

Use the NDK from command line

Automated build systems

Using the NDK CLI

Unleash the power of the command line

BlackBerry Jam Americas

- All of the NDK tools are available from the command line
  - Source the bbndk-env.sh script, or bbndk-env_10_1_0_1020.sh for a specific version
  - Windows users can run the corresponding .bat file
- Make a project configuration:

```
$ make Device-Release
```

- Make an exportable release build (interactive):

```
$ make package
```

- Traditional cross-compiling:
  - The binutils 2.22 suite and gcc/g++ 4.6.3 are available

```
$ ./configure --host=arm-unknown-nto-qnx8.0.0eabi
```

Automated Build Systems

BlackBerry Jam Americas

- Use a wide range of build systems such as ant to run periodic or commit-triggered builds

```
$ blackberry-nativepackager -package MyApp.bar
-configuration Device-Release -sign
-storepass <password> -cskpass <cskpass> bar-
descriptor.xml
```

- Use a webserver to deploy in-house builds via the browser

```
$ blackberry-deploy –installApp –device <ip
address> -password <user password> <bar file>
```

BlackBerry Jam Americas

Supporting multiple operating systems and devices

Architectural strategy

Wrapping an Objective-C library

Architectural Strategy

- BlackBerry 10 supports POSIX, as well as many gnu extensions to the standard
- Separate business logic from UI Code
  - This model is encouraged by the Qml / C++ split on BlackBerry10
- Use cross-platform libraries, frameworks and engines
  - BlackBerry10 includes a number of cross platform libraries
- Isolate platform specific code
  - If your app interacts with functionality that is specific to one OS, package the associated code in distinct classes

Wrapping an Objective-C Library

- In the public header, add a forward declaration of the implementation struct
- Use the struct type to hold the members you want to exclude from the public header
- Add a pointer to the struct as a class instance variable
- Define the members of the struct's in the .cpp file
- Construct an instance of the struct using the new operator
- Set the instance variable to the newly created struct
- Make sure delete is called on destruction.

For More Information …

BlackBerry Jam Americas

- Examples of building and using a third party library
  - Cascades-Community-Samples on github.com/blackberry
- KB Articles
  - Cascades and Qt library projects
  - Using Third party libraries
- The qmake Manual on qt-project.org
  - qmake variable and function reference
  - qmake advanced usage
- BlackBerry10 Developer Microsite Documentation
  - Best practices

For More Information …

BlackBerry Jam Americas

- Other courses that discuss strategies for supporting multiple BlackBerry devices
  - JAM27: Cascades 201: Three Devices, One App! Developing for Tablet, All Touch, and Physical Keyboard Devices.

BlackBerry Jam Americas

THANK YOU

JAM39

Roberto Speranza (@RSSessantotto)
Taso Perdikoulias

May 14-16, 2013