Porting applications to Qt



Kevin Funk, Software Engineer KDAB



- Some other toolkit \rightarrow Qt
 - QNX Photon
 - Motif
 - MFC

...

– Java AWT

• Older Qt version \rightarrow Qt5

			Goo	gle Search	i: QNX					
<u>F</u> ile <u>E</u>	Edit <u>V</u> ie	w <u>G</u> o	H <u>o</u> tlist	Help						
ᄸ Back	₽ Forward	Coord Home	🔇 Reload	Font+	🙆 Print	(III) Hotlist	Add Url	© Stop		
Site: ht	tp://www.go	iogle.ca/se	arch?q=Q	NX&ie=ISC	-8859-18	hl=en&me	eta=			
0		-	Advan	ced Search	Pret	ferences	Languag	e Tools	Search Tip	
						earch				
-	~~	0**	Search	n: 🖲 the v	veb 🔾 p	ages from	Canada		_	
Web	Images	Groups	Directo	ry New	5	0				
Searched	the web for	QNX.		Resi	uts 1 - 1 0	of about 2	85,000. Sea	arch took ().13 seconds.	
Category	: Comp	uters > S	oftware >	Operating	g System	s > Realt	ime > QI	NX		
QNX R	ealtime of	perating	system (1	RTOS) s	oftware,	<u></u>				
development tools							Sponsored Links			
QNX Software Systems: Reliable RTOS and Productive Development						Free Onx and Rtos Report				
Tools for Embedded							Premier sites, Demos, White papers			
Systems IntraLase cuts the risk of laser eye surgery with QNX-based							Free Info on Qnx and other Rtos			
system								Interest:		
Descriptio	on: Canadia	an firm tha	develops,	maintains,	and mark	ets QNX	11		10	
RTOS an	d related pr	oducts: Ph	oton micro	GUI,			ONX D	stribution		
Category:							IDE Runtimes Third Party BSP			
									+	
۵ III I	Document: I	Done					We	stern Euro	opean (ISO 88	

∄KDAB

Why migrate at all?

- Hard to find developers who know the legacy toolkit
- Legacy toolkit is not actively developed any more
- Applications have an outdated look-andfeel
- Support for more platforms
- Job satisfaction it is easier to attract good developers to work on something cool than something outdated

🗖 Widg	Widgets Demo										
Label Button 🗌 Checkbox	first choice 😐										
TextArea											
<u>s</u>		-									
TextField	two										
	1										

∄KDA

Qt Framework to the rescue!

- Actively developed C++ framework
- Under Open-Source Governance model
- Cross-platform
- Deployable to embedded devices
- Used in over 70 types of industries world-wide





- Migration techniques
 - Throw out all code, just copy the basics over.
 - Port code gradually using framework integrations
 - your app will be part Qt & part legacy toolkit (think of MFC)
 - Note: Not suitable for production code!
 - Comment out all code that doesn't compile on the target platform, and port it bit by bit.



∄KDA

- Potential tasks
 - Changing a legacy toolkit push button to a QPushButton.
 - Reworking mechanism like callbacks to signals/slots.
 - Converting code for painting and color allocation.
 - Rewriting usage of third party libraries, like
 - Custom JSON parser library
 - ..

- More potential tasks :(
 - Replace your own implementation of what Qt now supports natively (XML/Multi threading/File support)
 - Replacing unsupported third party tools like charting engines with new or custom made components.
 - Take advantage of the target frameworks capabilities like model/view or Qt designer to improve overall design quality.
 - Now think we have a few million lines of code, and understand the complexity!



- Some of the tricky parts :((
 - Should you use *char* *, *QString*, or your own string implementation including Unicode support?
 - Legacy toolkit may have been plain C, but you likely had a C++ wrapper layer, how do you port that?
 - You'd like to keep your custom style of the application how to bring to the Qt world?
 - User interface description files of legacy toolkits e.g. how to port QNX Photon's *.wgt* files?



- Issues to consider
 - Which build system to use QMake and CMake are both good candidates
 - It might be a good time to update your naming convention
 - It might be a good idea to update to a newer C++ standard
 - Still using CVS? Consider upgrading to SVN or Git
 - Peer reviews may help you towards a consistent port needs experts in the target framework



- Automatic Conversions
 - A one-to-one mapping is likely only possible for the simplest part of a port.
 - Improve your editors to support you while porting
 - Code snippet plugins
 - Stored RegEx-based replacement routines
 - Use scripting!
 - Popular these days: Clang tooling!



- Pitfalls
 - Feature Frenzy It is more fun to add a new feature than doing porting, so you risk you will never get done
 - Refactoring/Clean-up Frenzy when porting it isn't the right time to clean up(!)
 - Bug fixing Frenzy when you have a new look at the code you may find new bugs, or new ways to fix old bugs. Be careful not to drift away in a bug fixing frenzy where you fix a bug that everyone has accepted is there.



- General recommendations
 - Remind management why you want to port don't push for new features.
 - Remind **developers** why you want to port stay with what is the default in Qt (colors, fonts, ...) – *for now*
 - Remind customers why you want to port this is the hardest one!

KDAB Migration

- What we offer as part of a migration
 - free in-depth analysis report if you decide to use us for the project
 - Using our own set of code analysis tools
 - Reviews by our Senior engineers
 - porting of your code base from another toolkit or older Qt version to an up-todate Qt version
 - Using both proprietary and our open-sourced tools
 - In collaboration with the customer via code reviews, meetings
 - delivering a ready to ship version of the complete source code



Thank you!

- KDAB Services
 - Consulting

- ...

Customer projects

• KDAB Trainings

...

- QML for Embedded Linux
- Debugging & Profiling
- C++11 and C++14
- Modern OpenGL

Visit our booth for demos!

Contact: kevin.funk@kdab.com

