

Meet Qt 6.6, and more!

Dr. Cristián Maureira-Fredes @cmaureir

@ (?) in У





Slides available qtinfo.dev/gnuhealthcon23



After 3 online years it's good to be here in person



My motivation to be here

Qt et's talk about the Ot framework





Cross platform framework for UI interfaces, and applications.

- Started in 1991
- With many, many modules.

More info at qt.io

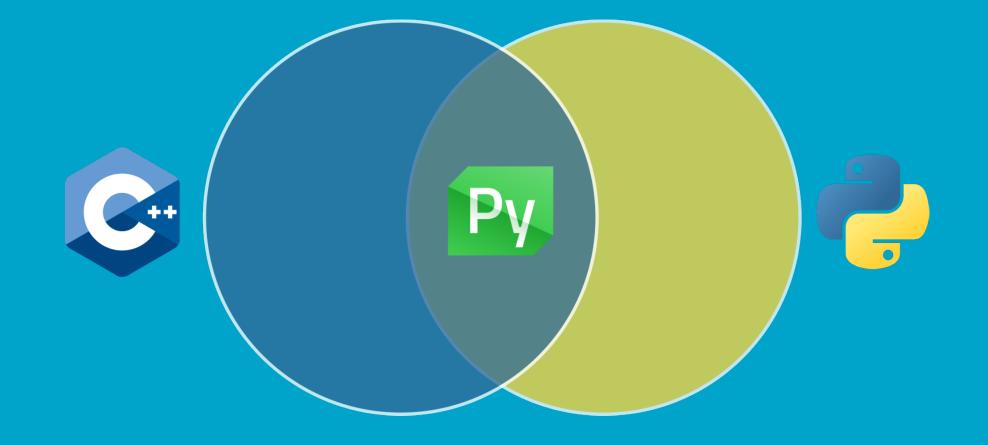


The 6.6 Release RC: 26.09.2023 Final Release: 10.10.2023





Where were we? 🤔





Motivation



Motivation

Python popularity *m*



Motivation

Python popularity *f*Relevance of C++ *b*



Motivation

Python popularity *f*Relevance of C++

Features

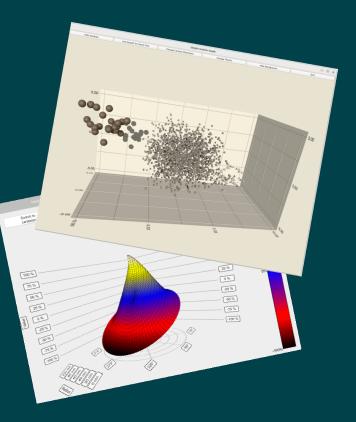
- PyPy compatibility
- New wheel structure
- Embedded support
- Tooling
 - pyside6-qml (app preview)
 - pyside6-deploy (using nuitka)
 - pyside6-project (new projects)
 - pyside6-qtpy2cpp (converter)

A few highlights of the 6.6 release

doc-snapshots.qt.io/qt6-6.6/whatsnew66.html



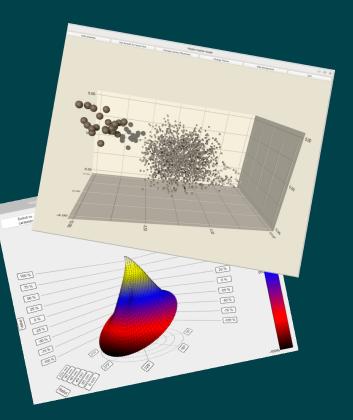
 New module, replacing OpenGL based DataVisualization



Surface Graph Gallery



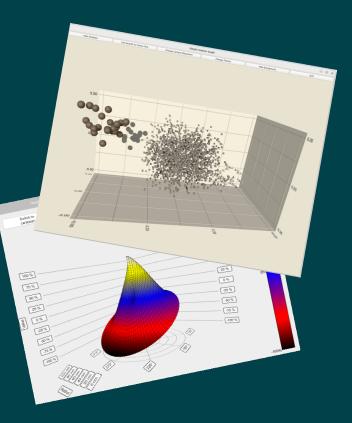
- New module, replacing OpenGL based DataVisualization
- Use cases
 - Vis of large quantities of dynamic data
 - Creation of depth maps



Surface Graph Gallery



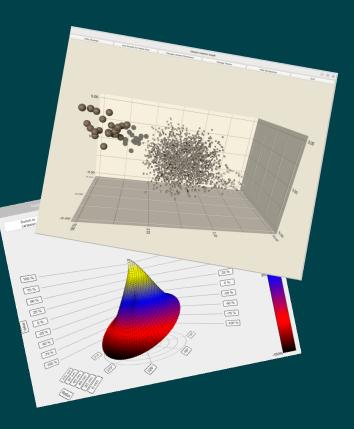
- New module, replacing OpenGL based DataVisualization
- Use cases
 - Vis of large quantities of dynamic data
 - Creation of depth maps
- Value
 - Using QML APIs from Quick3D, using bare metal interface and acceleration through RHI
 - Customizable charts with themes, items and labels
 - Possibility to embed Qt Quick 3D effects in graphs



Surface Graph Gallery

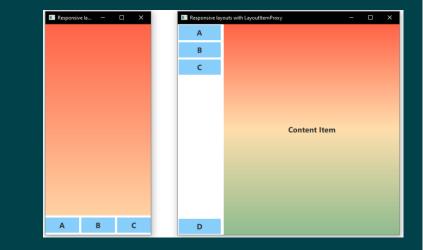


- New module, replacing OpenGL based DataVisualization
- Use cases
 - Vis of large quantities of dynamic data
 - Creation of depth maps
- Value
 - Using QML APIs from Quick3D, using bare metal interface and acceleration through RHI
 - Customizable charts with themes, items and labels
 - Possibility to embed Qt Quick 3D effects in graphs
- How to find
 - https://doc-snapshots.qt.io/qt6-6.6/qtgraphs-index.html
 - https://doc-snapshots.qt.io/qt6-6.6/graphs-examples.html



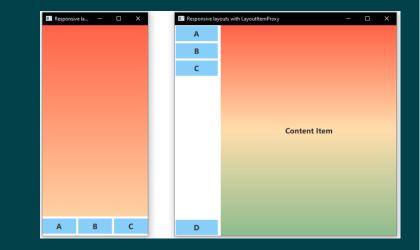
Surface Graph Gallery

 Intuitive way to create responsive layouts



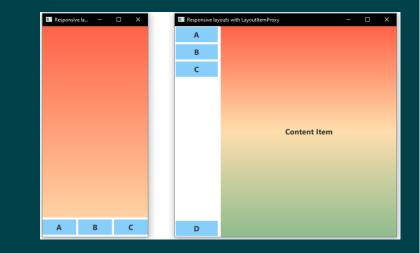
Responsive Layout Example

- Intuitive way to create responsive layouts
- Use cases
 - Re-usable GUI implementation for different window sizes
 - Dynamic handling portrait and landscape devices



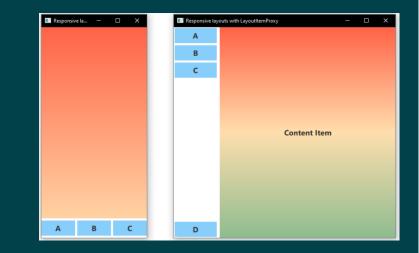
Responsive Layout Example

- Intuitive way to create responsive layouts
- Use cases
 - Re-usable GUI implementation for different window sizes
 - Dynamic handling portrait and landscape devices
- Value
 - Foster best practices and brand consistency across devices
 - Bring QML closer to web design standards and "mobile first" design approach



Responsive Layout Example

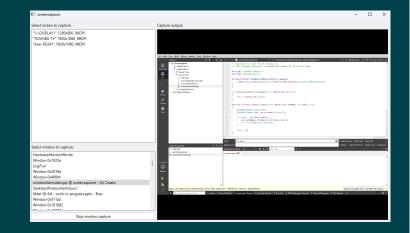
- Intuitive way to create responsive layouts
- Use cases
 - Re-usable GUI implementation for different window sizes
 - Dynamic handling portrait and landscape devices
- Value
 - Foster best practices and brand consistency across devices
 - Bring QML closer to web design standards and "mobile first" design approach
- How to find
 - https://doc-snapshots.qt.io/qt6-6.6/qml-qtquick-layoutslayoutitemproxy.html
- The Qt Company | Cristián Maureira-Fredes



Responsive Layout Example



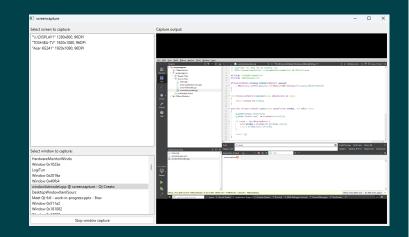
 Capture videos of individual desktop applications windows



Screen capture example



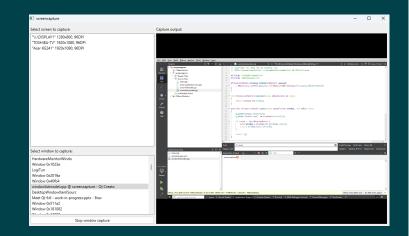
- Capture videos of individual desktop applications windows
- Use cases
 - Record windows for specific video
 - Streaming capabilities



Screen capture example



- Capture videos of individual desktop applications windows
- Use cases
 - Record windows for specific video
 - Streaming capabilities
- Value
 - Better control on what to capture from a screen
 - Increase safety on sharing content
 - Reduce effort on post-recording editing
 - Optimize resources usage



Screen capture example

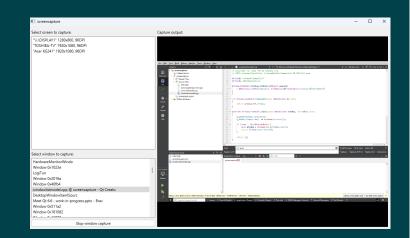


- Capture videos of individual desktop applications windows
- Use cases
 - Record windows for specific video
 - Streaming capabilities
- Value
 - Better control on what to capture from a screen
 - Increase safety on sharing content
 - Reduce effort on post-recording editing
 - Optimize resources usage

How to find

 https://doc-snapshots.qt.io/qt6-6.6/qml-qtmultimediawindowcapture.html





Screen capture example



 Advanced access to font shaping features

Text {

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
ext {
```

```
Text {
```

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
font.features: { "frac": 1 }
```



- Advanced access to font shaping features
- Use cases
 - Support OpenType features directly from Qt Framework

Text {

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
```

Text {

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
font.features: { "frac": 1 }
```



- Advanced access to font shaping features
- Use cases
 - Support OpenType features directly from Qt Framework
- Value
 - Enable users to deliver better font experiences

```
Text {
```

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
```

```
Text {
```

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
font.features: { "frac": 1 }
```



- Advanced access to font shaping features
- Use cases
 - Support OpenType features directly from Qt Framework
- Value
 - Enable users to deliver better font experiences
- How to find
 - https://doc-snapshots.qt.io/qt6-6.6/qml-qtquicktext.html#font.feature-prop

Text {

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
```

Text {

```
anchors.centerIn: parent
text: "One divided by two is 1/2"
font.features: { "frac": 1 }
```



 Small improvements on text to speech module

Rate:						
Pitch:						
Engine	Default				\sim	
Language:	English (United	States)			\sim	
Voice name:	Microsoft Mark	- Male - Other Age			\sim	
Spe	eak	Pause	Resume	Sto	p	
			dit->toPlainTe:			



 Small improvements on text to speech module

Use cases

- Text to audio synthetization
- Audio queueing
- Querying for available voices and engine capabilities

		1			
.fault				~	
	ates)			~	
				~	
	Pause	Resume	S	Stop	
9		lish (United States) rrosoft Mark - Male - Other Age	lish (United States) rosoft Mark - Male - Other Age	lish (United States) rosoft Mark - Male - Other Age	lish (United States) ~ rosoft Mark - Male - Other Age ~



 Small improvements on text to speech module

Use cases

- Text to audio synthetization
- Audio queueing
- Querying for available voices and engine capabilities

Value

New abilities to produce read-aloud fluent interfaces

Volume:			•			_
Rate:						
Pitch:						
Engine	Default					\sim
Language:	English (United State	es)				\sim
Voice name:	Microsoft Mark - Ma	ale - Other Age				\sim
Sp	beak	Pause	Resume		Stop	
ech->sy		const QAudio	it->toPlainText pFormat & format ut("test.mp3");	, cons	t QByt	teArray &by



 Small improvements on text to speech module

Use cases

- Text to audio synthetization
- Audio queueing
- Querying for available voices and engine capabilities

Value

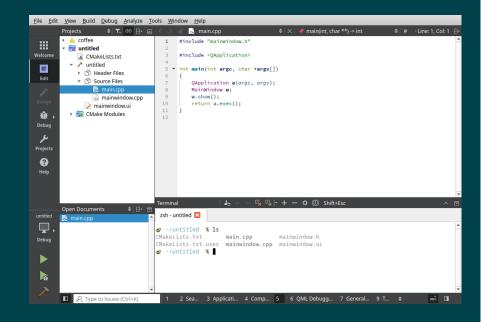
- New abilities to produce read-aloud fluent interfaces
- How to find
 - https://doc-snapshots.qt.io/qt6-6.6/qtexttospeech.html

Volume:						
Rate: Pitch:						
	Default		-		~	
2						
	English (United				~	
/oice name:	Microsoft Mar	k - Male - Other Age			\sim	
S	peak	Pause	Resume	Sto	p	
			Resume	Sto	p	



Qt Creator Improvements Developer Experience

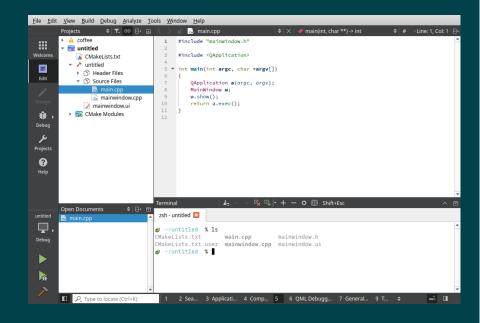
- Github Copilot support
- Integrated terminal





Qt Creator Improvements Developer Experience

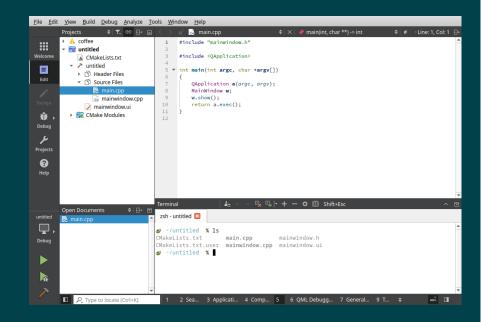
- Github Copilot support
- Integrated terminal
- Use cases
 - Get AI help for creating code, tests or document the code
 - Run command line tasks using the terminal





Qt Creator Improvements Developer Experience

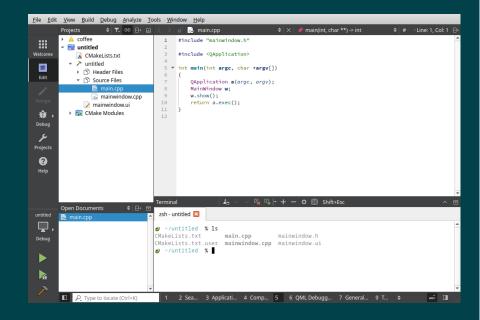
- Github Copilot support
- Integrated terminal
- Use cases
 - Get AI help for creating code, tests or document the code
 - Run command line tasks using the terminal
- Value
 - More productive SW development with Qt Creator

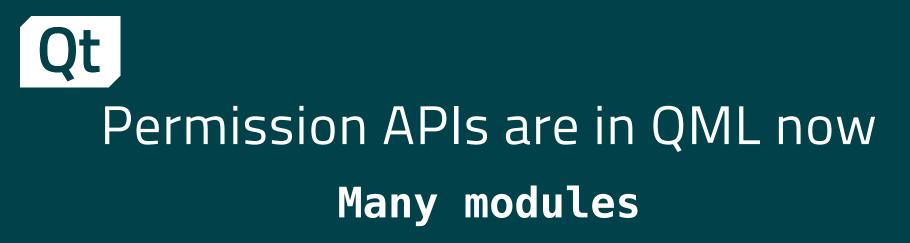




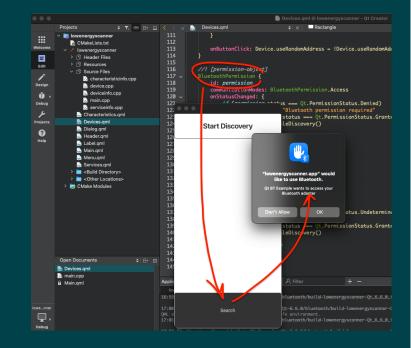
Qt Creator Improvements Developer Experience

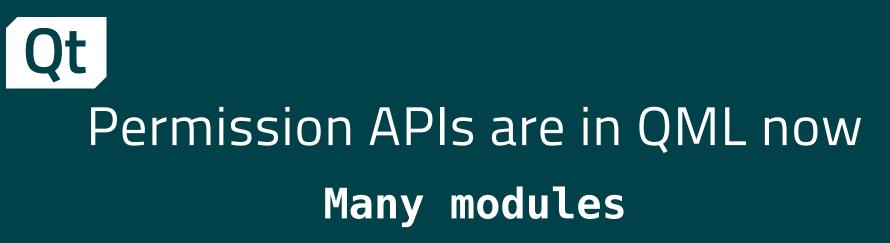
- Github Copilot support
- Integrated terminal
- Use cases
 - Get AI help for creating code, tests or document the code
 - Run command line tasks using the terminal
- Value
 - More productive SW development with Qt Creator
- How to find
 - Qt Creator 11+



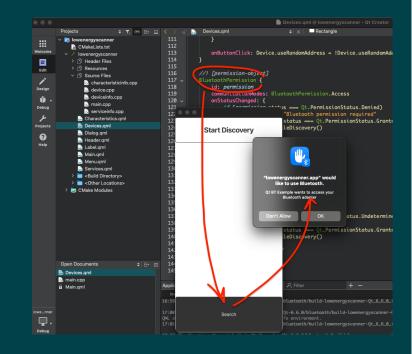


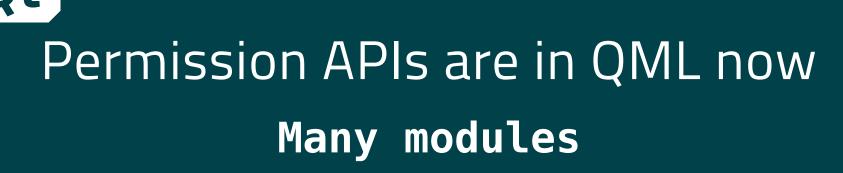
 New QML API for handling of permissions



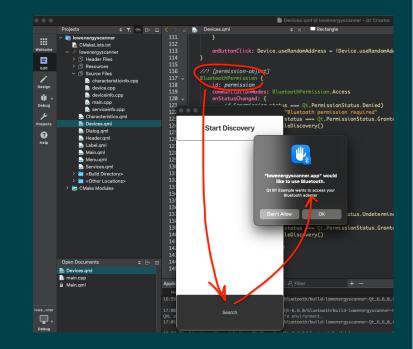


- New QML API for handling of permissions
- Use cases
 - An app for recording audio, using BLE, etc



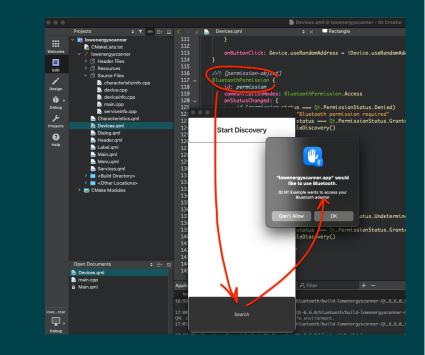


- New QML API for handling of permissions
- Use cases
 - An app for recording audio, using BLE, etc
- Value
 - Better UX: users get a permission request on an action
 - Permission requests just on start can be eliminated



Permission APIs are in QML now Many modules

- New QML API for handling of permissions
- Use cases
 - An app for recording audio, using BLE, etc
- Value
 - Better UX: users get a permission request on an action
 - Permission requests just on start can be eliminated
- How to find
 - https://doc.qt.io/qt6/qtcore-qmlmodule.html





 QML code fully compiled by the Qt Quick Compiler can be discarded from the app package

qt_standard_project_setup(REQUIRES 6.5) # <- this is needed!
qt_add_executable(appuntitled1</pre>

```
main.cpp
)
```

```
set_source_files_properties(
    Main.qml
    PROPERTIES
    QT_DISCARD_FILE_CONTENTS true # <- here, you discard it
)</pre>
```

qt_add_qml_module(appuntitled1
 URI untitled1
 VERSION 1.0
 QML_FILES Main.qml



- QML code fully compiled by the Qt Quick Compiler can be discarded from the app package
- Use cases
 - Avoid access to code with sensitive IP
 - Prevent manipulations of apps

```
qt_standard_project_setup(REQUIRES 6.5) # <- this is needed!
qt_add_executable(appuntitled1
    main.cpp
)
set_source_files_properties(
    Main.qml
    PROPERTIES
    QT_DISCARD_FILE_CONTENTS true # <- here, you discard it
)
qt_add_qml_module(appuntitled1
    URI untitled1
    VERSION 1.0
    OML_FILES Main.qml</pre>
```



- QML code fully compiled by the Qt Quick Compiler can be discarded from the app package
- Use cases
 - Avoid access to code with sensitive IP
 - Prevent manipulations of apps
- Value
 - Better IP protection and application integrity for Qt Quick based applications

```
qt_standard_project_setup(REQUIRES 6.5) # <- this is needed!</pre>
```

```
set_source_files_properties(
    Main.qml
    PROPERTIES
    QT_DISCARD_FILE_CONTENTS true # <- here, you discard it
)</pre>
```

qt_add_qml_module(appuntitled1
 URI untitled1
 VERSION 1.0
 QML_FILES Main.qml



- QML code fully compiled by the Qt Quick Compiler can be discarded from the app package
- Use cases
 - Avoid access to code with sensitive IP
 - Prevent manipulations of apps
- Value
 - Better IP protection and application integrity for Qt Quick based applications
- How to find
 - https://doc-snapshots.qt.io/qt6-6.6/cmake-source-file-property-qt-discardfile-contents.html
 - https://doc-snapshots.qt.io/qt6-6.6/resources.html#discarding-the-filecontents

qt_standard_project_setup(REQUIRES 6.5) # <- this is needed!</pre>

```
qt_add_executable(appuntitled1
______main.cpp
```

```
set_source_files_properties(
    Main.qml
    PROPERTIES
    QT_DISCARD_FILE_CONTENTS true # <- here, you discard it
)</pre>
```

```
qt_add_qml_module(appuntitled1
    URI untitled1
    VERSION 1.0
    QML_FILES Main.qml
```



 Create custom static analysis for QML code

😮 🖉 👼 best.gnd 🛛 🕸 🗶 🗱 bern	e Unix U/I
 // Copyright (C) 2023 The Qt Company Ltd. // SPDX-License-Identifier: LicenseRef-Qt 	
3 4 import QtQuick	
<pre>5 6 - Item { 1</pre>	Immediate heavy * Immediate heavy *
18 Text (text: root.greeting)	*****
QML Static Analysis 1 - Basic Setup	
Qt 6.7 + Qt QML Compiler + QML Static Analysis 1 - Basic Setup	Qt 6.7.0 Reference Documentation
To create our plugin, we first need to make the QmICompiler module available: find_package(gts #tgutHEB components gelCompiler)	
We then create a plugin, and link it against the QmlCompiler module.	
<pre>qt_sdt_plagi(vt)lame/dPlagis) target_savers(vt)lame/dPlagis repart hilledgis.h hilledgis.h target_ids_liberis(vt)lame/dPlagis MFINATE qt::qbt(aulier) </pre>	
The implementation follows the pattern for extending Qt with a plugin: We subc	ass the QGmISA::LintPlugin,
class HelloWorldPlugin : public QObject, public QQmlSA::LintPlugin {	



- Create custom static analysis for QML code
- Use cases
 - Write additional rules for QML Lint
 - Post additional messages for selected languages statements in the code

🕻 🔿 🚽 🔜 Seat.graf 🛛 🕸 🛇	and the second sec			e Unix (UFI
1 // Copyright (C) 20	323 The Qt Company Ltd.			
	entifier: LicenseRef-Qt-Comme	rcial OR BSD-3-Clause		
3 4 import OtDuick				
5				
6 - Item (kristinge@AMD factoring X 🧕 kristinged	pAd-Substring - X 🚯 InterimpedAdd Substring - X 🛞 Interim	water with the second second
7 id: root		istalrope@AAD-faultstrim	ng:~/gt6-build\$ gtbase/bin/gmllint -	P ~/at5/atdec
9 property strin 10	qt6	5/qtdeclarative/examples,	build-chapter2-Qt_qt6_build-Debug/ -/ /qmlcompiler/tutorials/helloworld/ch	
11 component MyTe 12	Inf		ted t6/qtdeclarative/examples/qmlcompile ng [Plugin.HelloWorld.hello-world]	er/tutorials/h
	Text : Item {	MyText { text:		
15 } 16			t6/qtdeclarative/examples/qmlcompile	er/tutorials/h
17 Text { text:	"Hello world:" } ml: root.greeting }	Text { text: "Goodbye		
QML Static Analysis 1 - Basic Se	tup			
Qt 6.7 + Qt QML Compiler + QML Static Analysi			Ot 6.7.0 Reference Documentation	
Gt 6.7 F Gt GML Compiler F GML Static Analysi				
	O C	QML Static Analysis Tutorial	QML Static Analysis 2 - Custom Pass 📀	
This chapter introduces the basic structure of a que can be used with gmllint.	nllint extension plugin, and how it			
To create our plugin, we first need to make the Qr	mICompiler module available:			
find_package(Qt6 REQUIRED COMPONENTS Qmlc	Compiler)			
We then create a plugin, and link it against the Qr	nlCompiler module.			
qt_add_plugin(HelloWorldPlugin)				
target sources(HelloWorldPlugin				
PRIVATE				
helloplugin.h helloplugin.cpp				
)				
target_link_libraries(HelloWorldPlugin P#	IVATE Qt::QmlCompiler)			
The implementation follows the pattern for extend	ing Qt with a plugin: We subclass the	QQmISA::LintPlugin,		
class HelloWorldPlugin : public QObject,	public QQmlSA::LintPlugin			
1				



- Create custom static analysis for QML code
- Use cases
 - Write additional rules for QML Lint
 - Post additional messages for selected languages statements in the code
- Value
 - make QML code compliant with your company's coding guidelines ns

🕻 🔿 📄 best.gnd 👘 🛛 🖉 Bern	+ Unix (Uni
 // Copyright (C) 2023 The Qt Company Ltd. // SPDX-License-Identifier: LicenseRef-Ot 	-Commercial OR BSD-3-Clause
3 4 import OtOuick	
5	
6 - Item (7 id: root	😢 interingengi MAD Backering: X 🕘 interingengi AAD-backering - X 🌒 interingengi MAD-backering - X 🕘 interingengi AAD-backering - X
<pre>8 9 property string greeting: "Hello" 10 10 12 23 33 - component MyText : Text {} 14 14 15 15 15 15 15 15 15 15 15 15 15 15 15</pre>	icristalope0AD-faultating:-/gite-builds gtbase/hi/gullit - 9-/cd6/gtee line/tutristalope0AD-faultating:-/gite-builds gtbase/hi/gullit - 9-/cd6/gtee line/tutristalope0AD-faultating:-/gite-build-build-build-build-build- build welle build sambled line: /howicristalope0gtd/declarative/samble/gle/complex/ltsti nl/226: incerret greeting [Fugin Hellewild helleworld] hylet [tex: 'codeworld']
15 } 16 17 Text { text: "Hello world!" } 18 Text { text: root.greeting }	<pre>Info: /home/icristalrope/qt5/qtdeclarative/examples/qulcompiler/tutorials/hu ml:19:19: Incorrect greating [Plugin.HelloWorld.hello-world] Text { text: "Goodby world!" }</pre>
QML Static Analysis 1 - Basic Setup	
Qt 6.7	Qt 6.7.0 Reference Documentation
	O QML Static Analysis Tutorial QML Static Analysis 2 - Custom Pass O
This chapter introduces the basic structure of a gmillint extension plugin, and ho can be used with gmillint. To create our plugin, we first need to make the QmICompiler module available:	w ł
<pre>find_package(Qt6 REQUIRED COMPONENTS QmlCompiler)</pre>	
We then create a plugin, and link it against the QmICompiler module.	
qt_ast_plagi(wilase/d#lagis) taggt_serve(wilase/d#lagis #Nilasiats, %lingigt.cg taggt_link_linvist(wilase/d#lagis #ETWIF qt:;@dCompler)	
The implementation follows the pattern for extending Qt with a plugin: We subcli-	ass the QQmISA::LintPlugin,
class HelloWorldPlugin : public QObject, public QQmISA::LintPlugin	

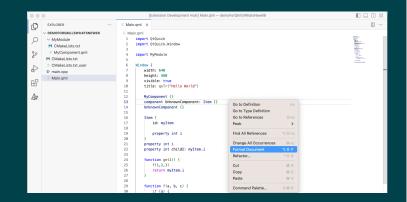


- Create custom static analysis for QML code
- Use cases
 - Write additional rules for QML Lint
 - Post additional messages for selected languages statements in the code
- Value
 - make QML code compliant with your company's coding guidelines ns
- How to find
 - Check "QML Static Analysis Tutorial"

🔿 🚽 🔜 🖬	st.qnd 8 × @ Bern	e unixiU1
	Copyright (C) 2023 The Qt Company Ltd.	
2 //	SPDX-License-Identifier: LicenseRef-Qt-	-Commercial OR BSD-3-Clause
	port OtOuick	
5		
6 - It		😢 innaneeeBAD fasherey X 🔕 inisangeBAD-fashtning - X 🚯 inisangeBAD-fashtning - X 😣 inisangeBAD-fashtning - X
7	id: root	icristalrope@AAD-faultstring:~/gt6-build\$ gtbase/bin/gmllint -P ~/gt6/gtdec
9	property string greeting: "Hello"	<pre>iler/tutorials/helloworld/build-chapter2-Qt_qt6_build-Debug/Plugin.Hellow qt6/qtdeclarative/examples/qnlcompiler/tutorials/helloworld/chapter2/test.qu</pre>
11 12	component MyText : Text {}	Hello World plugin is enabled Info: /home/icristalrope/qt6/qtdeclarative/examples/qmlcompiler/tutorials/hu
13 - 14	component NotText : Item { property string text	<pre>ml:22:26: Incorrect greeting [Plugin.HelloWorld.hello-world] HyText { text: "Goodbye world!" }</pre>
15	}	Info: /home/icristalrope/ot6/otdeclarative/examples/gmlcompiler/tutorials/hu
16 17 18	Text { text: "Hello world!" } Text { text: root.greeting }	<pre>http://www.secure.com/do/dowciation/www.secure.complety/dioriation/ mli9:i9: Incorrect greating [Pugin.HelleWorld.hello-world] Text { text: "Goodbye world!" }</pre>
QML Static Analys	sis 1 - Basic Setup	
Qt 6.7 ► Qt QML Compiler	 QML Static Analysis 1 - Basic Setup 	Qt 6.7.0 Reference Documentation
		O QML Static Analysis Tutorial QML Static Analysis 2 - Custom Pass O
This chapter introduces the	basic structure of a qmllint extension plugin, and how	w it
can be used with qmllint.		
To exact one share an fee	at need to make the QmICompiler module available:	
to create our plugin, we first	a need to make the QmiCompiler module available:	
find_package(Qt6 REQ	UIRED COMPONENTS QmlCompiler)	
We then create a plugin, an	d link it against the QmlCompiler module.	
qt_add_plugin(Hellow	lor1dPlugin)	
target_sources(Hello PRIVATE	MorldPlugin	
helloplugin.	h	
helloplugin.	cpp	
'		
target_link_librarie	s(HelloWorldPlugin PRIVATE Qt::QmlCompiler)	
The implementation follows	the pattern for extending Qt with a plugin: We subcla	ass the QQmISA::LintPlugin,
class HelloworldBlue	in : public ODbject, public ODmISA::LintPlugin	
{	and here designed here designed and a	
A 003107		

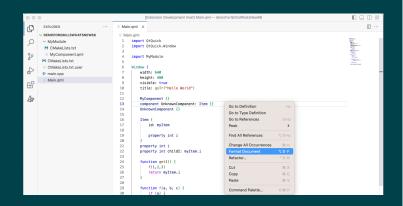


 Can be used in Visual Studio Code, and QtC 10+





- Can be used in Visual Studio Code, and QtC 10+
- Use cases
 - Far more for developers than just a spellchecker in Word





- Can be used in Visual Studio Code, and QtC 10+
- Use cases
 - Far more for developers than just a spellchecker in Word
- Value
 - 100% coverage of the actual QML language model
 - Interworking with other tools, like Qt Quick Compiler

00	0	[Extension Development Host] Main.qml — demoForQmllsWhatsNew66	
Ø	EXPLORER	S Main.qml × S Main.qml	□ …
2	 MyModule M CMakeLists.txt 	1 import QtQuick 2 import QtQuick.Window	Banna Banna Martines Martines Martines Martines Martines Martines Martines Martines
ĥ	MyComponent.qml M CMakeLists.txt	4 import MyModule 5	NL Net-
÷	CMakeLists.txt.user main.cpp Main.gml	6 Window (7 width: 640 8 height: 480	-
₿	- Martigrit	9 visible: true 10 title: qsTr("Hello World") 11	
A		12 MyComponent {} 13 component UnknownComponent: Item {} Go to Definition 4 UnknownComponent Q Fig.	- 1
		Item { Go to Type Definition 16 Item { Go to References OF12	
		18 19 property int i Find All References COPI2	
		20) 21 property int i 22 property int childI: myltem.i Format Document X:0 F	
		23 function getI() { Refactor ^O R 25 f(1,2,3) Cut % ×	
		26 return myItem.i Copy % C 27 } Paste % ∨	
		29 function f(a, b, c) { 30 if (a) { Command Palette ○※ P	



- Can be used in Visual Studio Code, and QtC 10+
- Use cases
 - Far more for developers than just a spellchecker in Word
- Value
 - 100% coverage of the actual QML language model
 - Interworking with other tools, like Qt Quick Compiler
- How to find
 - Post: https://qt.io/blog/whats-new-in-qml-languageserver-qmlls-shipped-with-qt-6.6
 - qmlls --help

000	[Extension Development Host] Main.qml — demoForQmlIsWhatsNew66	
CONCRETE CONCRETE CONCRETE CONCRETE CONCRETE	··· E Main.gml x	E OR
<i>43</i> ¢	13 MyComponent () Octo Definition rul 13 OutsourComponent () Octo Definition rul 14 OutsourComponent () Octo Definition rul 15 Tites (Octo Definition rul 16 Tites (Octo Definition rul 17 Property int i Prod All Informaces Conv 18 property int i Othore Million Conv 19 property int i Othore Million Conv 10 function grites.i Octo Million Conv 10 function grites.i Octo Million Conv 16 file file Convert Million Convert Million	



Qt gRPC: overview of the ongoing works (TP) - (1/2) QrGrpc and Qt Protobuf

• Qt 6.6

- Added support for channel and call options
- Integrated QML-types support in code generation
- New QML API: in discussion and under reviews
- Support for "oneof"
- qmake support



Qt gRPC: overview of the ongoing works (TP) - (2/2) QrGrpc and Qt Protobuf

Plans for Qt 6.7

- Improve HTTP/2 support in Qt, then Qt gRPC
- Reduce dependencies to "gRPC" itself
- Support more well-known types
- Finalize QML API
- QProtobuf conformance testing
- gRPC interoperatibility testing
- Review and extended/improve examples if needed



Qt gRPC: overview of the ongoing works (TP) - (2/2) QrGrpc and Qt Protobuf

Plans for Qt 6.7

- Improve HTTP/2 support in Qt, then Qt gRPC
- Reduce dependencies to "gRPC" itself
- Support more well-known types
- Finalize QML API
- QProtobuf conformance testing
- gRPC interoperatibility testing
- Review and extended/improve examples if needed

- Wish to fully release in 6.7. It is going to be popular modules, we better do them well
- How to find
 - Since Qt 6.5, Qt gRPC and Qt Protobuf are in docs and Qt Bug Reports
 - They are listed as modules and components



- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more

bor. The	* Modifier_ob.	
Rirror of	s modifier_ob ject to mirror mirror_object	
- Tor lod	lject to sirror airror_object	
	aurror_object	
diring a section .	MIRNOR_X*1	
alror ad	HURBOR X":	
lirror and	use_x True use_y False	
operation	use y False use False ATTRACK y*	
ror_ed.	MIRROR Y=	
	use y - True	
-Operation	False	
Pror and	HUNOR_2*1	
ror_mod.	use_z = True	
	at the end -add	
ob.selec	t= 1	
Ler_ob.sel	lect-1	
	ne.objects.active	
Selecter	select = 0	
The best cont	wet selected observe	
ata, objec	ts[one.name].selation	
int("plea	se select exactly	
COERA	OR CLASSES	
types. Ope	rator): r to the selected	
Cypes in	e to the person	
iect.mit	ror_mirror_x	
	ve_object is not	
(text)	whiert is not	
ontent, act	ve_ous	



- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more
- Use cases
 - Have a more control on how Chromium runtime behaves

	Shirton Oblas
	a store object to store
	airror dejact to sirror airror mediatror dejact
	arron allert to at
	Sirror all ror
	Peraties
	IL TOP MILE MILE AND A STATE OF A
	peration = TORON /*: direct and one x = TORON /*: direct and one x = Toron direct and
	Group and use y take Group and use y take Operation == Traise
	and the second se
	False
	operation set false
	operation == ?UBD0_2*1 From mod.use x False
	From the second se
	rror_mod.use_y = False
	Lror_mod.use_z = True
	Selection at the end add
	_OD.select=1
	ler_ob.select=1
	mentext.scene.objects.action
	"Selected" + str(modifier
	Firm ob.select = 0
	bpy.context.selected_ob
	bpy:contextine.name].telent
	and an
	int("please select exactly
	OPERATOR CLASSES
	OPERATOR CLOSES
	types. Operator): X signor to the selected X signor stror_x
	sypesticate to the second
	X mirror sirror
	X mirror to the man sject.mirror_sirror_x
and the second se	POF 0
	30m a 400
	motext) a object 18
	active-
	unterst): net-setive_sbject is not



- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more
- Use cases
 - Have a more control on how Chromium runtime behaves
- Value
 - Additional API covering more use cases

t airror delas	
airror object to sirre	
aliron object to airon	
and all to since	
and the second s	
approximation - Tutana Pri alizzaria da Santana Pri alizzaria da Santa Pri alizzaria da Santa Prima	
Print Hold Head	
- Alle	
I CION IN MILLION	
in the second seco	
irror.mod.use_x False	
mod-use_z = False	
operation == "MIRROR_2"	
FTOT_mod_use_x = False	
Califor_mod.use_z = True	
selection at the end add	
_ob.select= 1	
ler_ob.select-1	
intext.scene.objects.actin	
("Selected" + str(modifier	
irror_ob.select = 0	
a here context, selected of	
Mate.objects[one.name].tel	
mint("please select exactly and	
OPERATOR CLASSES	
ypes. Operator): X airror to the selected	
types, one to the select	
X mirror mirror	
A sirror_sirror_s	
antext): extra active_object is not	
tart): tient is not	
active object	



- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more
- Use cases
 - Have a more control on how Chromium runtime behaves
- Value
 - Additional API covering more use cases
- How to find
 - https://doc-snapshots.qt.io/qt6-6.6/whatsnew66.html#qt-webenginemodule

	t sirror bias	
	t sirror object to sirror sirror_sol_sirror_solject	
	airror object to airror peration a arror object	
	-mod-mirror object	
	Birror-Bod.use_x = Trag Birror-Bod.use_x = Trag Birror-Bod.use_y = False Dame Bod.use y = False	
	Contraction of the second seco	
	HTTPs	
	Operation == "MIRROR_"	
	False	
	Chror_mod.use_z = True	
	selection at the end -add	
	ob.select= 1	
	ler_ob.select-1	
	"Selected" + str(modifier)	
	pippor ob.select = 0	
	bpy.context.selected_ob mata.objects[one.name].sel	
	mint("please select exactly and	
	OPERATOR CLASSES	
	t (men	
and the second se	x mirror sirror x	
	a sirror sirror	
	entext): ext.active_object is not	
	axt.active_	



- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for QtLoader.js

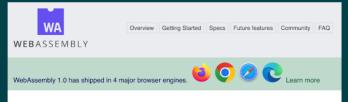


WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

Generated file	Brief Description
app.html	HTML container
qtloader.js	JavaScript API for loading Qt apps
app.js	JS API for loading Qt apps
app.wasm	app binary



- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for QtLoader.js
- Use cases
 - WebAssembly app developer

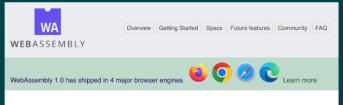


WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

Generated file	Brief Description
app.html	HTML container
qtloader.js	JavaScript API for loading Qt apps
app.js	JS API for loading Qt apps
app.wasm	app binary



- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for QtLoader.js
- Use cases
 - WebAssembly app developer
- Value
 - QtLoader used for landing the wasm app benefits of the new implementation (easier debugging/maintenance)



WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

Generated file	Brief Description
app.html	HTML container
qtloader.js	JavaScript API for loading Qt apps
app.js	JS API for loading Qt apps
app.wasm	app binary



- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for QtLoader.js
- Use cases
 - WebAssembly app developer
- Value
 - QtLoader used for landing the wasm app benefits of the new implementation (easier debugging/maintenance)
- How to find
 - https://doc.qt.io/qt-6/wasm.html



WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

Generated file	Brief Description
app.html	HTML container
qtloader.js	JavaScript API for loading Qt apps
app.js	JS API for loading Qt apps
app.wasm	app binary



- Improve developer experience with Qt on Android
 - Added FileProvider support when using QDesktopServices::openUrl()
 - Use AndroidX by default
 - More Qt examples are tailored for Android
 - Android 13 is max supported version
 - Support SDK level 33 to match Play Store req.

Qt 6.6 > <u>Getting Started with Qt for Android</u>

Getting Started with Qt for Android

The video below is a beginner's guide to using the Qt for Android toolchain, including the Qt Creator IDE, to get you started developing Android apps.



The rest of this page has more detailed getting started information.

To download and install Qt for Android, follow the instructions on the Getting Started with Qt page.



- Improve developer experience with Qt on Android
 - Added FileProvider support when using QDesktopServices::openUrl()
 - Use AndroidX by default
 - More Qt examples are tailored for Android
 - Android 13 is max supported version
 - Support SDK level 33 to match Play Store req.
- Use cases
 - Mobile app dev with cross-platforms in focus

Qt 6.6 > <u>Getting Started with Qt for Android</u>

Getting Started with Qt for Android

The video below is a beginner's guide to using the Qt for Android toolchain, including the Qt Creator IDE, to get you started developing Android apps.



The rest of this page has more detailed getting started information.

To download and install Qt for Android, follow the instructions on the Getting Started with Qt page.



- Improve developer experience with Qt on Android
 - Added FileProvider support when using QDesktopServices::openUrl()
 - Use AndroidX by default
 - More Qt examples are tailored for Android
 - Android 13 is max supported version
 - Support SDK level 33 to match Play Store req.
- Use cases
 - Mobile app dev with cross-platforms in focus
- Value
 - Better API coverage with more use cases supported

Qt 6.6 > Getting Started with Qt for Androic

Getting Started with Qt for Android

The video below is a beginner's guide to using the Qt for Android toolchain, including the Qt Creator IDE, to get you started developing Android apps.



The rest of this page has more detailed getting started information.

To download and install Qt for Android, follow the instructions on the Getting Started with Qt page.



- Improve developer experience with Qt on Android
 - Added FileProvider support when using QDesktopServices::openUrl()
 - Use AndroidX by default
 - More Qt examples are tailored for Android
 - Android 13 is max supported version
 - Support SDK level 33 to match Play Store req.
- Use cases
 - Mobile app dev with cross-platforms in focus
- Value
 - Better API coverage with more use cases supported
- How to find
 - https://doc-snapshots.qt.io/qt6-6.6/android-getting-started.html

Qt 6.6 > Getting Started with Qt for Androic

Getting Started with Qt for Android

The video below is a beginner's guide to using the Qt for Android toolchain, including the Qt Creator IDE, to get you started developing Android apps.



The rest of this page has more detailed getting started information.

To download and install Qt for Android, follow the instructions on the Getting Started with Qt page.



Qt for Python





Asynchronous compatibility

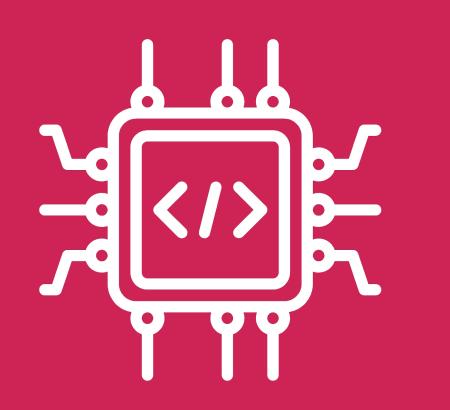


Qt

<pre>from PySide6.QtCore import (Qt, QEvent, QObject, Signal, Slot) from PySide6.QtWidgets import (QApplication, QLabel, QMainWindow, QPushButton,</pre>	QVBoxLayout, QWidget)
import outcome import signal import sys import traceback import trio	
	ifname == "main":
You, vor 7 Monaten 1 author (You) class MainWindow(QMainWindow):	<pre>app = QApplication(sys.argv)</pre>
<pre>start_signal = Signal()</pre>	main_window = MainWindow()
<pre>definit(self): super()init()</pre>	<pre>async_helper = AsyncHelper(main_window, main_window.set_text)</pre>
widget = QWidget() self.setCentralWidget(widget)	<pre>main_window.show()</pre>
layout = QVBoxLayout(widget)	signal signal STCINT signal STC DEL)
<pre>self.text = QLabel("The answer is 42.") layout.addWidget(self.text, alignment=Qt.AlignmentFlag.AlignCenter)</pre>	<pre>signal.signal(signal.SIGINT, signal.SIG_DFL) app.exec()</pre>
<pre>async_trigger = QPushButton(text="What is the question?")</pre>	
async_trigger.clicked_connect[self.async_start) layout.addWidget(async_trigger, alignment=Qt.AlignmentFlag.AlignCenter)	
<pre>@Slot() def async_start(self):</pre>	
self.start_signal.emit()	
<pre>async def set_text(self): await trio.sleep(1) self.text.setText("What do you get if you multiply six by nine?")</pre>	



Embedded wheels





Packages for **aarch64**

Targeting Raspberry Pi devices mainly
To be available on PyPI



Android



View Go Run Terminal Help

i view Go Run Terminal Help		
	$\cdots + \diamond \text{ videosettings.py 9+} \diamond \text{ ui_videosettings.py 9+} \diamond \text{ buildozer.py 9+} \times \diamond \cdots \diamond \diamond$	
	sources > pyside-tools > deploy_lib > android > 🍦 buildozer.py > 😭 BuildozerConfig	
URCECONTROL		
SIDE-SETUP	PROBLEMS 355 OUTPUT DEBUG CONSOLE TERMINAL GITLENS +	×
designer external graphs gui httpserver installer_test location r multimedia	<pre>(venv) shyamnath@shyamnath-ThinkPad-X1-Carbon-Gen-9:~/qt_for_python/pysid e-setup/examples/multimedia/camera\$ pyside6-android-deploywheel-pyside =/home/shyamnath/qt_for_python/pyside-setup/dist/PySide6-6.6.0a1-6.6.0-cp 37-abi3-android_x86_64.whlwheel-shiboken=/home/shyamnath/qt_for_python /pyside-setup/dist/shiboken6-6.6.0a1-6.6.0-cp37-abi3-android_x86_64.whl - -name=camerakeep-deployment-files</pre>	
: COMMIT DETAILS		
JTLINE		
MELINE		
: LINE HISTORY		
: FILE HISTORY		
: VISUAL FILE HISTORY		
: SEARCH & COMPARE		
: REPOSITORIES		
: COMMITS		
: BRANCHES		
: REMOTES		
: STASHES		
: TAGS		
: WORKTREES		
፡ copheeQtoeompany Cris	stián Maureira-Fredes	
	CMake: [Debug]: Ready 🎇 [GCC 9.3.0 x86_64-linux-gnu] 🔀 Build [all] ▷ 🛛 { } Python 3.10.5 ('venv': venv) 🖗 G	o Li



For license holders Installation

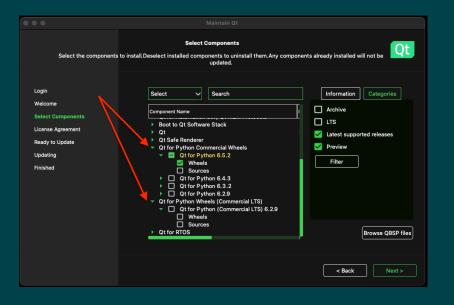
Maintenance Tool

Select the component	Select Components s to install.Deselect installed components to uninstall them.Any components already installed will not be updated.
Login Welcome Select Components License Agreement Ready to Update Updating Finished	Select Search Component Name Archive LTS Qt Started Qt Qt Safe Renderer Qt for Python 6.5.2 Qt for Python 6.4.3 Qt for Python 6.3.2 Qt for Python 8.2.9 Sources Qt for Python 8.2.9 Browse QBSP files
	< Back Next >



For license holders Installation

Maintenance Tool



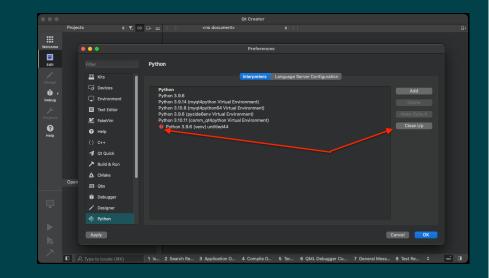
	🕅 Qt — vladimir@man		
→ Qt ls			
5.15.14	6.5.1	MaintenanceTool.app	Tools
5.15.8	6.5.2	MaintenanceTool.dat	components.xml
5.15.9	6.6.0	MaintenanceTool.ini	dist
6.2.9	Docs	Qt Creator.app	installerResources
6.3.0	Examples	QtDesignStudio	licenseInfo.txt
6.4.0	InstallationLog.txt	QtForPython	network.xml
6.5.0	Licenses	README.txt	
→ <mark>Qt</mark> ls QtFo	rPython		
6.5.2			
→ Qt			

qtpip



Towards a Qt flavored Python IDE QtCreator

- Interpreter selection
- Virtual env creation
- Automatic PySide installation
- More to come





Silent features and fixes **PySide** and **Shiboken**



How can Qt contribute to Libre and Open Health initiatives?



Let's be in touch!



Communication channels

- Telegram <u>https://t.me/qtforpython</u>
- IRC/Matrix #qt-pyside on Libera.chat
- Mailing list <u>lists.qt-project.org/mailman/listinfo/pyside</u>

More platforms at wiki.qt.io/Qt_for_Python#Community



Don't forget the documentation



loc.qt.io/qtforpython/

Py.

Qt for Python

earch

- Python Documentation Python Getting Started
- ng Started on Linux
- ig Started on macOS
- ng Started on Windows
- ation Framework
- nimation Framework
- ow and Dialog Widgets
- Overview
- ndable Properties
- etooth Examples
- ra Overview
- atibility Map
- ges to Qt Concurrent
- iner Classes
- inate System

Qt for Python

Qt for Python offers the official Python bindings for Qt, which enables you to use Python to write your Qt applications. The project has two main components:

- PySide6, so that you can use Qt6 APIs in your Python applications, and
- Shiboken6, a binding generator tool, which can be used to expose C++ projects to Python, and a Python
 module with some utility functions.

Porting from PySide2 to PySide6 provides information on porting existing PySide2 applications.

This project is available under the LGPLv3/GPLv3 and the Qt commercial license.

Quick Start

You can obtain the latest stable version by running pip install pyside6. If you want to build it yourself, check the getting started guide.

To learn how to use it, check out write your first application, and to learn what is installed with the pyside6, check the package content, structure, and tools page.

Documentation



0

CONTENTS Quick Start Documentation





Meet Qt 6.6, and more!

Dr. Cristián Maureira-Fredes @cmaureir

@ (?) in У

