

Qt for Python

5.9.2024



Qt for Python Team

- Dr. Cristián Maureira-Fredes cristian.maureira-fredes@qt.io (Team Lead, Build system)
- Friedemann Kleint Friedemann.Kleint@qt.io (Shiboken bindings generator)
- Shyamnath Premnadh Shyamnath.Premnadh@qt.io (Android, deployment)
- (External) Christian Tismer tismer@stackless.com (Python)
- *New Hire starting in October*

Larger features since the last QtCS

AsyncIO support (Tech Preview)

- A Pure Python module (no Qt-binding) that allows programs to be written with Qt API-alike code and the asyncio module.
- **Currently supporting:** features, tasks, handles, executors, and event loop management functions.

Larger features since the last QtCS

Lazy loading for improving startup time

- Startup time hit caused by Python enums
- <https://bugreports.qt.io/browse/PYSIDE-2404>
- Based on the PEP690 idea of delaying module imports.
- In PySide's case, we load the classes on demand.

[Python](#) » [PEP Index](#) » [PEP 690](#)



PEP 690 – Lazy Imports

Author: Germán Méndez Bravo <german.mb at gmail.com>, Carl Meyer <carl at oddbird.net>

Sponsor: Barry Warsaw <barry at python.org>

Discussions-To: [Discourse thread](#)

Status: [Rejected](#)

Type: [Standards Track](#)

Created: 29-Apr-2022

Python-Version: 3.12

Post-History: [03-May-2022](#), [03-May-2022](#)

Resolution: [Discourse message](#)

► [Table of Contents](#)

[Abstract](#)

This PEP proposes a feature to transparently defer the finding and execution of imported modules until the moment when an imported object is first used. Since Python programs commonly import many more modules than a single invocation of the program is likely to use in practice, lazy imports can greatly reduce the overall number of modules loaded, improving startup time and memory usage. Lazy imports also mostly eliminate the risk of import cycles.

[Motivation](#)

Larger features since the last QtCS

Flatpak support for KDE applications

- <https://develop.kde.org/docs/getting-started/python/>
- Base Application available in <https://github.com/flathub/io.qt.PySide.BaseApp/>

The screenshot shows the KDE documentation website for 'Python with Kirigami'. The navigation bar includes 'Developer', 'Documentation', 'API', 'KDE Human Interface Guidelines', and 'English'. The breadcrumb trail is 'Documentation / Getting started / Python with Kirigami'. The main heading is 'Python with Kirigami' with the subtext 'Create KDE applications using Python.' Below this, there is an introductory paragraph: 'Linux applications with QML and Python? Why not? Python is a popular programming language. QML offers an intuitive way to create user interfaces. Kirigami extends QML to provide useful UI components and it implements UI/UX patterns for mobile and desktop. We will fit these technologies together and create a simple application.' Three feature boxes are displayed: 'Your first Python + Kirigami application' (Learn how to write an application with PyQt/PySide), 'Creating a Python package' (Understand the requirements to create your own Python package), and 'Publishing your Python app as a Flatpak' (Ship your app easily to users). A small Qt logo is visible in the bottom right corner of the page.

Larger features since the last QtCS

Reduction of binary size

- Still under development.
- First steps: strings in generated code compressed/extracted.
- Each package includes the Qt binaries, and tools, but we are trying to reduce the binding-generated code.

- 📄 [PySide6_Essentials-6.7.2-cp39-abi3-win_amd64.whl](#) (78.9 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `Windows x86-64`
- 📄 [PySide6_Essentials-6.7.2-cp39-abi3-manylinux_2_31_aarch64.whl](#) (88.0 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `manylinux: glibc 2.31+ ARM64`
- 📄 [PySide6_Essentials-6.7.2-cp39-abi3-manylinux_2_28_x86_64.whl](#) (87.7 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `manylinux: glibc 2.28+ x86-64`
- 📄 [PySide6_Essentials-6.7.2-cp39-abi3-macosx_11_0_universal2.whl](#) (154.0 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `macOS 11.0+ universal2 (ARM64, x86-64)`

- 📄 [PySide6_Addons-6.7.2-cp39-abi3-win_amd64.whl](#) (123.0 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `Windows x86-64`
- 📄 [PySide6_Addons-6.7.2-cp39-abi3-manylinux_2_31_aarch64.whl](#) (122.7 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `manylinux: glibc 2.31+ ARM64`
- 📄 [PySide6_Addons-6.7.2-cp39-abi3-manylinux_2_28_x86_64.whl](#) (137.5 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `manylinux: glibc 2.28+ x86-64`
- 📄 [PySide6_Addons-6.7.2-cp39-abi3-macosx_11_0_universal2.whl](#) (258.2 MB [view hashes](#))
Uploaded Jun 18, 2024 `CPython 3.9+` `macOS 11.0+ universal2 (ARM64, x86-64)`

Larger features since the last QtCS

Android support improvements

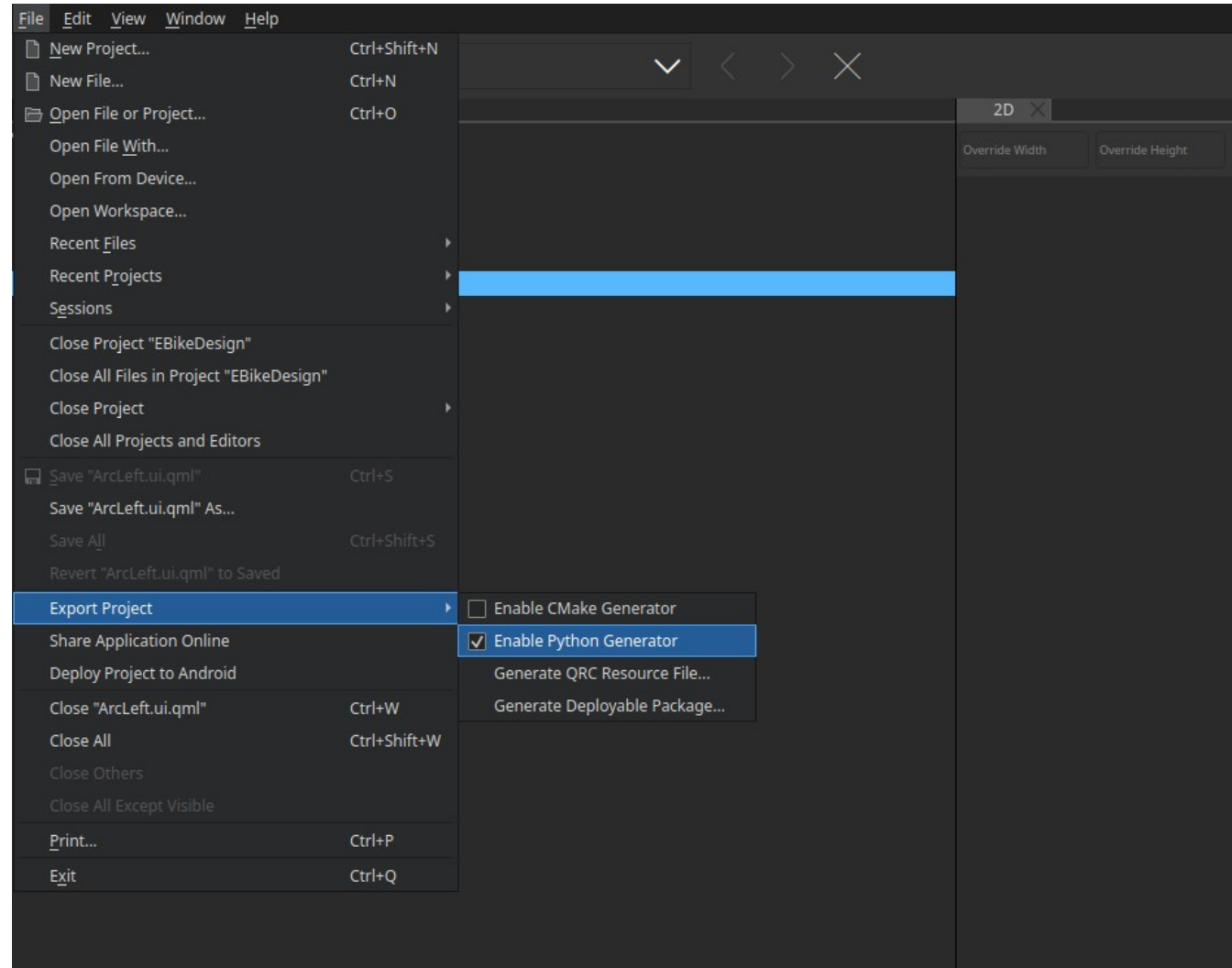
- Cross-compilation and `pyside6-android-deploy` now works on macOS. Earlier, this was Linux-only.
- Android deployment compatible with 6.7 and 6.8 (once dev switches)
- Improved the usage of `pyside6-android-deploy`
- Added multi-architecture support (aarch64, armv7a, i686, x86_64)
- Added caching of NDK, SDK, cross-compiled Python, toolchain files.



Larger features since the last QtCS

Qt Design Studio support

- Special package to include the special DS Quick Components: **PySide6_DS**
- QtDS 4.6 will have a menu option "Export Project" → "Enable Python Generator"



Larger features since the last QtCS

Research work on QtScript (Dynamic bindings)

- Initially, provide QtScript functionalities in Python
- **A separate session will happen on this topic**

Discussion topics

- AsyncIO support needs a new Maintainer and needs contributions
- Python is dropping the mandatory GIL in 3.13: NoGIL support
- Stable API support and planned CPython development
- Other pain-points?