Creating Projects

One of the major advantages of Qt Design Studio is that it allows a team of designers and developers to share a project across different development platforms with a common tool for design, development, profiling, and debugging.

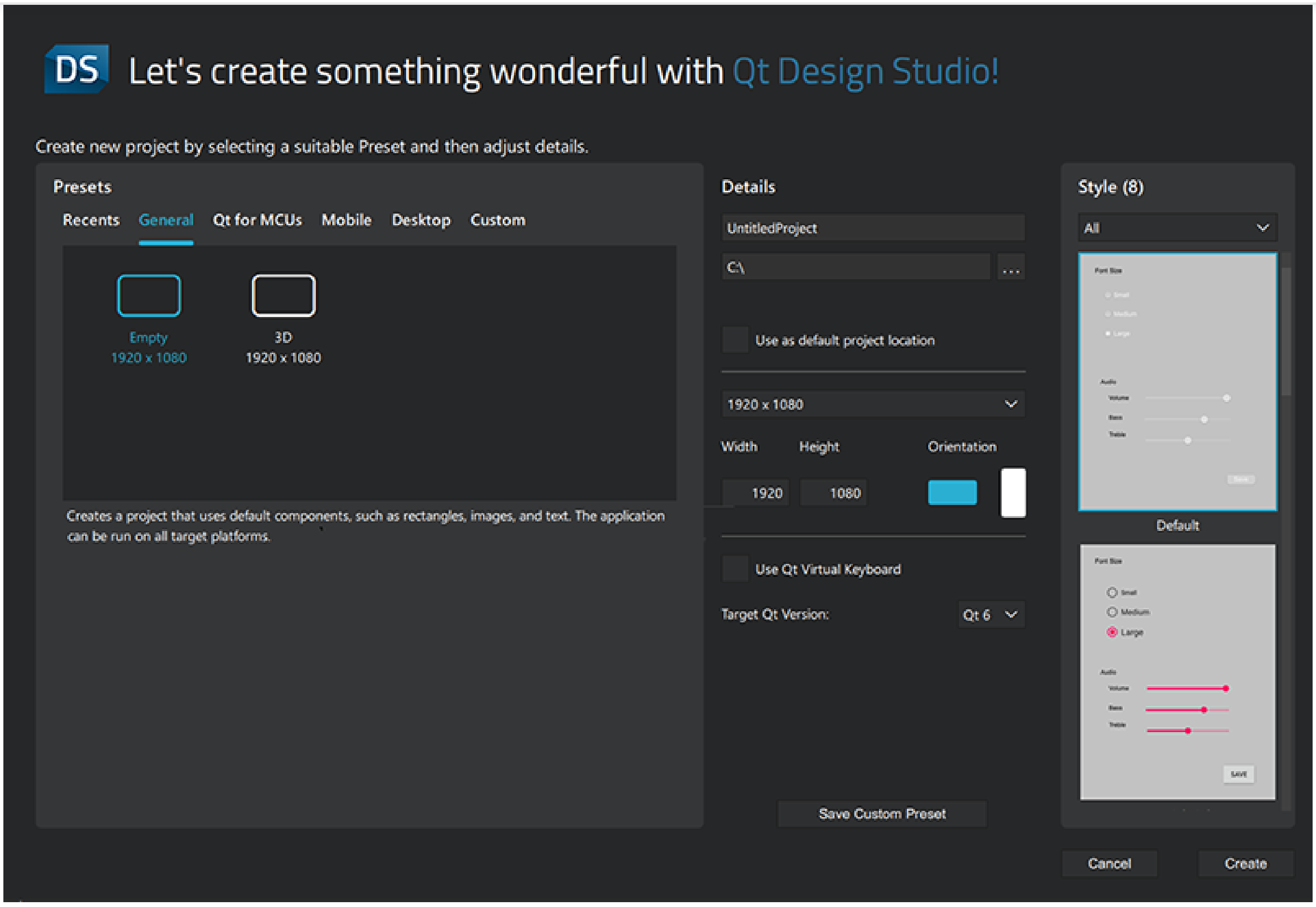
Creating a project enables you to:

- Group files together.
- Include UI files (.ui.qml), component files (.qml), and assets files.
- Specify settings for previewing UIs.

Setting up a new project in Qt Design Studio is aided by a wizard with a set of presets that you can choose from. You can adjust project settings and save custom presets. When you create a project, all necessary files are created.

The following presets are available:

Category	Preset	Purpose
Recents		Lists your most recently used presets.
General	Empty	Creates a project that uses default components such as rectangles, images, and text. You can run the application on all target platforms.
	3D	Creates a project that uses default and 3D components such as cameras, lights, 3D models, and materials.
Qt for MCUs	MCU	Creates an application that uses a subset of default components (as supported by Qt for MCUs) that you can deploy, run, and debug on MCU boards.
Mobile	Scroll	Creates an application that uses Qt Quick controls to implement a scrollable list.
	Stack	Creates an application that uses Qt Quick controls to implement a set of pages with a stack-based navigation model.
	Swipe	Creates an application that uses Qt Quick controls to implement a swipable screen.
Desktop	Launcher	Creates a project that uses default components such as rectangles, images, and text, and defines a launcher application.
Custom		Lists your saved custom presets. **Note:** This tab is not visible if there are no saved custom presets.


DS Let's create something wonderful with Qt Design Studio!

Create new project by selecting a suitable Preset and then adjust details.

Creates a project that uses default components, such as rectangles, images, and text. The application can be run on all target platforms.

Details: UntitledProject, C:\, 1920 x 1080, Width 1920, Height 1080, Target Qt Version: Qt 6

To test how well your designs work, you can preview the UIs on the desktop, embedded Linux devices, or Android devices. For more information, see Validating with Target Hardware.

You can export designs from other design tools and import them to projects. For more information, see Exporting from Design Tools and Importing Designs From Other Design Tools.

Creating a Project

To create a project:

1. Select **File** > **New Project**.
2. In the **Presets** tab, select a preset.
3. In the **Details** tab:
   - Enter a name for the project. Keep in mind that projects cannot be easily renamed later.
   - Select the path for the project files. You can move project folders later.
   - Set the screen resolution for previewing the UI on the desktop or on a device. This determines the screen size. You can change the screen size later in Properties.
   - Select **Use Qt Virtual Keyboard** to enable users to enter text using a virtual keyboard.
   - In **Target Qt Version**, select the Qt version to use for developing the application. While you can change the Qt version later in the **Run Settings** of the project, keep in mind that the two versions are not fully compatible.
4. In the **Style** tab, select one of the predefined UI styles to use.
5. Select **Create** to create the project.

- .qmlproject project file defines that all component, JavaScript, and image files in the project folder belong to the project. Therefore, you do not need to individually list all the files in the project.
- .qml file defines the functionality and appearance of a component.
- *Screen01.ui.qml* defines a custom component that you can edit in the 2D view. For more information, see UI Files.

  By default, this is the main file in the project, but you can change that in the .qmlproject file. While the custom component is a good starting point for new users, you don't have to use it. Specifically, if you export and import designs using Qt Bridge, your main file is most likely called something else. For more information, see Exporting from Design Tools.
- *CMakeLists.txt* project configuration file allowing you to share your project as a fully working C++ application with developers.
- qtquickcontrols2.conf file specifies the preferred style and some style-specific arguments.
- *fonts* folder contains font files that you have added in **Assets**.
- *imports* folder contains a *Constants.qml* file that specifies a font loader for the Arial font and the screen resolution. The size of the default Screen.ui.qml Rectangle should be set as `width: Constants.width & height: Constants.height` so that it inherits the global resolution saved here.
- *qmldir* module definition file declares the Constant component. For more information, see Module Definition qmldir Files.

To use JavaScript and image files in the UI, select **Assets** > + .

Using Custom Presets

You can save project settings as custom presets. All saved custom presets are available on the **Custom** tab in the **Create Project** wizard. You cannot modify custom presets once you have created them.

To create a custom preset:

1. In the **Create Project** wizard, set the details and style that you want to use.
2. Select **Save Custom Preset** and give a name for the custom preset.

Adding Files to Projects

You can use wizard templates to add individual files to projects.

The wizard templates in the **Qt Quick Controls** category create stylable versions of the components in the **Qt Quick Controls** module. For more information, see Creating Custom Controls.

You can create the following types of files:

Category	Wizard Template	Purpose
Qt Quick Files	Flow Item and Flow View	Generate components that you can use to design the application flow.
	Qt Quick File	Generates a component with one of the following default components or positioners as the root component: Item, Rectangle, Image, Border Image, Flickable, Row, Column,

	UI File	
	Qt Quick Views	Generates a Grid View or a List View. For more information, see List and Grid Views.
Qt Quick Controls	Custom Button	Creates a push button with a text label.
	Custom CheckBox	Creates a check box.
	Custom Dial	Creates a dial.
	Custom Slider	Creates a slider.
	Custom SpinBox	Creates a spin box.
	Custom Switch	Creates a switch with on and off states.
	Pane	Provides a background that matches the UI style and theme.
	StackView	Provides a stack-based navigation model.
	SwipeView	Enables users to navigate pages by swiping sideways.
QML Files	ListModel	Adds a list model to the project.
JavaScript	JavaScript File	Generates files that you can use to write the application logic. This is useful for testing the application before the developers implement the application logic in C++, for example. For more information, see Simulating Application Logic.

‹ Keyboard Shortcuts

Use Cases ›

Qt DOCUMENTATION

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login

Contact Us
Customer Success

Community

Contribute to Qt
Forum
Wiki

Downloads
Marketplace

Feedback Sign In