

Using Components

A *component* is a reusable building block for a UI.

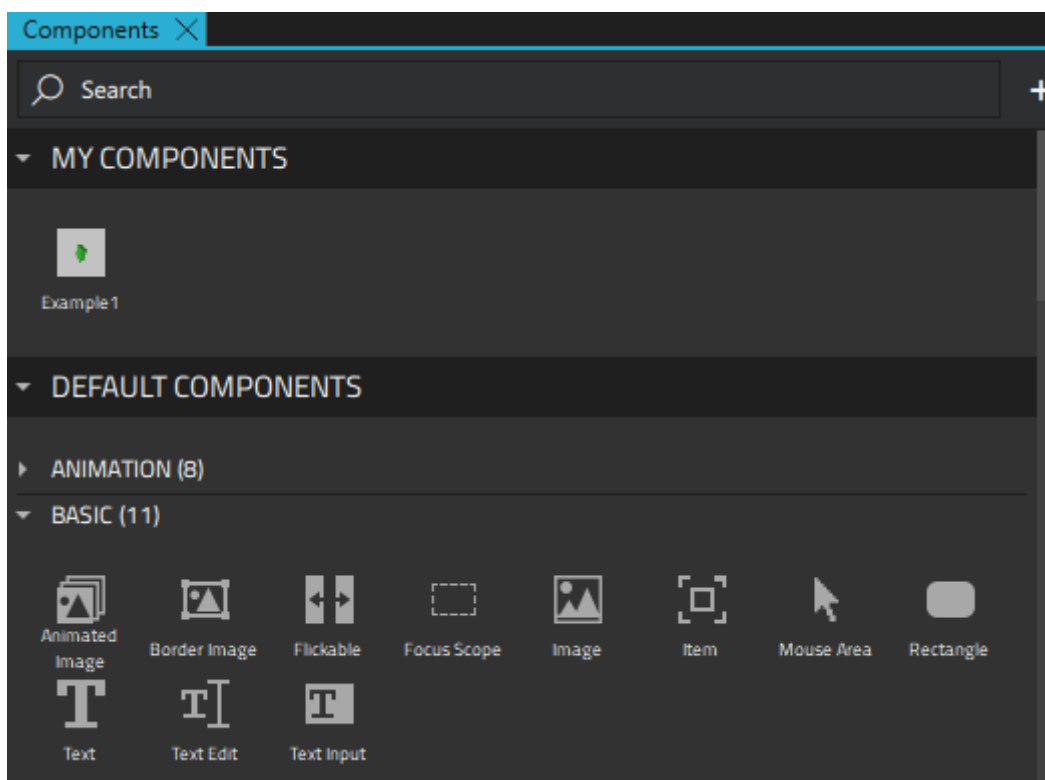
Qt Design Studio comes with *preset components* that you can use in your UI by creating instances of them. The preset components are similar to *Symbols* in Sketch or *Prefab* in Unity.

Some of the preset components represent simple shapes, text, or images, while others represent complex UI controls with full functionality, such as spin boxes or sliders. You can also add instances of preset [3D components](#) to your UIs.

To build your own components, you can modify the *properties* of the component instances and combine them.

A component is specified within one file (with the file extension *ui.qml* or *.qml*). For example, a Button component may be defined in *Button.ui.qml*. Typically, the visual appearance of a component is defined in a *UI file* (ui.qml). To create component files, you can use [wizard templates](#), or [move component instances into separate component files](#).

Select **Components** to view the preset components that have been added to your project. If you cannot find the components you need, add the relevant modules to your project, as described in [Adding and Removing Modules](#).



Read more about components:

[› Creating Custom Components](#)

Using Components Economically

It is important to understand the performance costs associated with using components.

To use components efficiently and economically:

- › Componentize your design by packaging your resources into reusable components that can be conveniently recombined to suit the needs of your UI.
- › Use as few components as necessary. To minimize the number of components, use [alias properties](#) and [states](#) to create the differences in your component instances. We recommend reusing components instead of duplicating them, so the components do not need to be processed as completely new component types. This reduces loading and compilation time as well as the size of the binary.
- › Any content that is data-driven should be exported as a public property (alias property) of the relevant component. For example, a speedometer should have an `int` or `real` property for speed to which the UI is bound.
- › Separate UI from the application logic. Designers should work with the [UI files](#) (.ui.qml), while developers should work on the corresponding implementation files (.qml) to define their programmatic behaviors or JavaScript. This enables iteration from both the design and development side of the process without the risk of overwriting each other's work.

[‹ Reacting to External Events](#)[Preset Components ›](#)[Contact Us](#)

Company

[About Us](#)
[Investors](#)
[Newsroom](#)
[Careers](#)
[Office Locations](#)

Licensing

[Terms & Conditions](#)
[Open Source](#)
[FAQ](#)

Support

For Customers



Partners

Training

Qt Login

Contact Us

Customer Success

Community

Contribute to Qt

Forum

Wiki

Downloads

Marketplace

© 2022 The Qt Company

Feedback Sign In