

Dependency property

In WPF, dependency properties are typically exposed as standard .NET [properties](#). At a basic level, you could interact with these properties directly and never know that they're implemented as a dependency property.

The purpose of dependency properties is to provide a way to compute the value of a property based on the value of other inputs. These other inputs might include system properties such as themes and user preferences, or just-in-time property from data binding and animations.

A dependency property can be implemented to provide validation, default values, and callbacks that monitor changes to other properties. Derived classes can also change some specific characteristics of an existing property by overriding dependency property metadata, rather than creating a new property or overriding an existing property.

Dependency object

Another type that is key to the WPF property system is the [DependencyObject](#). This type defines the base class that can register and own a dependency property. The [GetValue](#) and [SetValue](#) methods provide the backing implementation of the dependency property for the dependency object instance.

The following example shows a dependency object that defines a single dependency property identifier named `ValueProperty`. The dependency property is created with the `Value` .NET property.

