

Developer  
/ Documentation  
/ Unreal Engine ▾  
/ Unreal Engine 5.4 Documentation  
/ Programming and Scripting  
/ Programming with C++  
/ Delegates  
/ Dynamic Delegates

# Dynamic Delegates

Delegates that can be serialized and support reflection.



Dynamic delegates can be serialized, their functions can be found by name, and they are slower than regular delegates.

## Declaring Dynamic Delegates

Dynamic delegates are declared in the same manner as you [declare standard delegates](#) except they use the macro variations specific to dynamic delegates.

Declaration Macro	Description
<code>DECLARE_DYNAMIC_DELEGATE[_RetVal, ...] \\ DelegateName \)</code>	Creates a dynamic delegate.
<code>DECLARE_DYNAMIC_MULTICAST_DELEGATE[_RetVal, ...] \\ DelegateName \)</code>	Creates a dynamic multi-cast delegate.

# Dynamic Delegate Binding

Helper Macro	Description
<code>BindDynamic( UserObject, FuncName )</code>	Helper macro for calling BindDynamic() on dynamic delegates. Automatically generates the function name string.
<code>AddDynamic( UserObject, FuncName )</code>	Helper macro for calling AddDynamic() on dynamic multi-cast delegates. Automatically generates the function name string.
<code>RemoveDynamic( UserObject, FuncName )</code>	Helper macro for calling RemoveDynamic() on dynamic multi-cast delegates. Automatically generates the function name string.

## Executing Dynamic Delegates

The function bound to a delegate is executed by calling the delegate's `Execute()` function. You must check if delegates are "bound" before executing them. This is to make the code more safe as there may be cases where delegates have return values and output parameters that are uninitialized and subsequently accessed. Executing an unbound delegate could actually scribble over memory in some instances. You can call `IsBound()` to check if the delegate is safe to execute. Also, for delegates that have no return value, you can call `ExecuteIfBound()`, but be wary of output parameters that may be left uninitialized.

Execution Functions	Description
<code>Execute</code>	Executes a delegate without checking its bindings
<code>ExecuteIfBound</code>	Checks that a delegate is bound, and if so, calls Execute
<code>IsBound</code>	Checks whether or not a delegate is bound, often before code that includes an <code>Execute</code> call

See [Multi-cast Delegates](#) for details on executing multi-cast delegates.