

Developer  
/ Documentation  
/ Unreal Engine ▾  
/ Unreal Engine 5.4 Documentation  
/ Programming and Scripting  
/ Programming with C++  
/ Delegates  
/ Multi-cast Delegates

# Multi-cast Delegates

Delegates that can be bound to multiple functions and execute them all at once.



Multi-cast delegates have most of the same features as single-cast delegates. They only have weak references to objects, can be used with structs, can be copied around easily, etc. Just like regular delegates, multi-cast delegates can be loaded/saved, and triggered remotely; however, multi-cast delegate functions cannot use return values. They are best used to easily pass a collection of delegates around.

## Declaring Multi-Cast Delegates

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

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

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

## Binding Multi-Cast Delegates

Multicast delegates can have multiple functions bound that all get called when the delegate fires. As a result, the binding functions are more array-like in semantics.

Function	Description
<code>Add()</code>	Adds a function delegate to this multi-cast delegate's invocation list.
<code>AddStatic()</code>	Adds a raw C++ pointer global function delegate.
<code>AddRaw()</code>	Adds a raw C++ pointer delegate. Raw pointer does not use any sort of reference, so may be unsafe to call if the object was deleted out from underneath your delegate. Be careful when calling <code>Execute()</code> !
<code>AddSP()</code>	Adds a shared pointer-based (fast, not thread-safe) member function delegate. Shared pointer delegates keep a weak reference to your object.
<code>AddUObject()</code>	Adds a UObject-based member function delegate. UObject delegates keep a weak reference to your object.
<code>Remove()</code>	Removes a function from this multi-cast delegate's invocation list (performance is $O(N)$ ). Note that the order of the delegates may not be preserved!
<code>RemoveAll()</code>	Removes all functions from this multi-cast delegate's invocation list that are bound to the specified UserObject. Note that the order of the delegates may not be preserved!



`RemoveAll()` will remove all registered delegates bound to the provided pointer. Keep in mind that Raw delegates that are not bound to an object pointer will not be removed by this function!

See `DelegateSignatureImpl.inl` (located in `..\Engine\Source\Runtime\Core\Public\Delegates\`) for the variations, arguments, and implementations of these functions.

## Multi-Cast Execution

Multi-cast delegates allow you to attach multiple function delegates, then execute them all at once by calling the multi-cast delegate's `Broadcast()` function. Multi-cast delegate signatures are not allowed to use a return value.

It is always safe to call `Broadcast()` on a multi-cast delegate, even if nothing is bound. The only time you need to be careful is if you are using a delegate to initialize output variables, which is generally very bad to do.

The execution order of bound functions when calling `Broadcast()` is not defined. It may not be in the order the functions were added.

Function	Description
<code>Broadcast()</code>	Broadcasts this delegate to all bound objects, except to those that may have expired.