Shared function

AddEventHandler

> **Important Note:** Do **NOT** use the same name for your handler function as the event name, as this can lead to confusion if multiple handler functions are used. On the same note, for multiple reasons, it is **NOT** a good idea to export the same functions that you use locally as remote event handlers.

This function will add an event handler. An event handler is a function that will be called when the event it's attached to is triggered. See event system for more information on how the event system works.

> **Important Note:** See code for this note below

Event handlers are functions that are called when a particular event happens. Each event specifies a specific set of variables that are passed to the event handler and can be read by your function. The following global variables are available for use in handler functions:

- **source**: the element that triggered the event
- **this**: the element that the event handler is attached to
- **sourceResource**: the resource that triggered the event.
- **sourceResourceRoot**: the root element (dynamic element root on client) of the resource that triggered the event.
- **client**: the client that triggered the event using triggerServerEvent. Not set if the event was not triggered from a client.

- **eventName**: the name of the event which triggered the handler function.

It is important to remember that events pass up and down the element tree. An event triggered on the root element is triggered on every element in the tree. An event triggered on any other element is triggered on its ancestors (its parent element and its parent's parent etc) and its children, grandchildren and great-grandchildren. You can use the *propagate* argument to specify if you wish your handler to receive events that have propagated up or down the tree.

The order in which event handlers are triggered is undefined, you should not rely on one event handler being executed before another.

> **Note:** See Script security for tips on preventing cheaters when using events and element data

> **Note:** See Event Source Element for a descriptive visualization of the event system handling an event trigger.

Each function closure can only be added once to each event. On the second attempt to add the function closure to the same event a warning will be emitted to the debug console and the call to addEventHandler will fail.

Syntax

```
bool addEventHandler ( string eventName, element attachedTo, function handlerFunction [, bool propagate = true, string priority = "normal" ] )
```

Required Arguments

- **eventName:** The name of the event you want to attach the handler function to. **Note: The maximum allowed length is 100 ASCII characters (that is, English letters and numerals)**
- **attachedTo:** The element you wish to attach the handler to. The handler will only be called when the event it is attached to is triggered for this element, or one of its children. Often, this can be the root element (meaning the handler will be called when the event is triggered for *any* element).
- **handlerFunction:** The handler function you wish to call when the event is triggered. This function will be passed all of the event's parameters as arguments, but it isn't required that it takes all of them.

Optional Arguments

- **propagate:** A boolean representing whether the handler will be triggered if the event was propagated down or up the element tree (starting from the source), and not triggered directly on attachedTo (that is, handlers attached with this argument set to *false* will only be triggered if *source == this*). In GUI events you will probably want to set this to *false*.

- **priority :** A string representing the trigger order priority relative to other event handlers of the same name. Possible values are:
  - **"high"**
  - **"normal"**

  - **"low"**

It is also possible to add finer priority control by appending a positive or negative number to the priority string. For example (in priority order for reference): "high+4" "high" "high-1" "normal-6" "normal-7" "low+1" "low" "low-1"

> **Important Note:** Anything bound to a specific element will be run before other handlers that are bound to something higher in the element tree (like root) This means that "high+10" bound to root **won't** trigger before "normal" bound directly to an element.

Returns

Returns *true* if the event handler was attached successfully. Returns *false* if the specified event could not be found or any parameters were invalid.

Remarks

Due to the additional set of global variables, the event-trigger specific variables it is **NOT a good idea to use the same function locally as well as directly as an event handler**. Event handlers often make use of the source element variable which would often find no use in generic functions. Inside of server-side remote event handlers it is important to add protections against exploits due to unexpected client event triggers or network-based load situations while generic functions, due to being part of a controlled call-stack, do not in general face the same issues. It is recommended to adapt a **good-natured distancing** principle between code meant to run from local logic in separation to code meant to run from remote logic.