

Overview

The Flowgraph is the implementation of visual scripting, so allowing non-programmers to interact with different parts of CRYENGINE.

For more information about the Flowgraph in the Sandbox Manual see: [Flow Graph Editor](#).

Adding the FlowEvents Table

The FlowEvents table is needed to define the custom inputs and outputs supported by the Script Entity.

```
NewEntity.FlowEvents =
{
    Inputs =
    {
        PowerOn = { NewEntity.Event_PowerOn, "bool" },
        PowerOff = { NewEntity.Event_PowerOff, "bool" },
        Position = { NewEntity.Event_Position, "Vec3" },
    },
    Outputs =
    {
        PowerOn = "bool",
        PowerOff = "bool",
    },
}
```

Value types supported for inputs and outputs:

- **string**
- **bool**
- **entityid**
- **int**
- **float**
- **vec3**

Adding a Flow Node Input

The following line, taken from the Inputs table contained inside the FlowEvents, outlines the addition of a flow node input.

```
PowerOn = { NewEntity.Event_PowerOn, "bool" },
```

Each input element is defined by its name and a table which contains two elements:

1. A reference to the script function to call when the input is activated
2. The value type accepted by the input.

The following code snippet represents a skeleton of the implementation of a function that would receive the flow input.

```
function NewEntity:Event_Position(sender, position)
-- read the position value and set it somewhere
end
```

The following code snippet represents the implementation of a function that would read the input value.

```
function NewEntity:Event_PowerOn(sender)
-- code to implement a reaction to the flow node input
end
```

Adding a Flow Node Output

A new element needs to be added to the Outputs table with a name and the value type of the output. Example:

```
PowerOn = "bool",
```

An output is activated by calling the function `BroadcastEvent` with a string representing the name of the output to be activated.

```
BroadcastEvent(self, "PowerOn");
```

If output value matters, use function `ActivateOutput`:

```
self:ActivateOutput("ActiveCount", self.nActiveCount);
```