

# IBehaviorInstance Script Interface

**View online:** <https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/object-interfaces/ibehaviorinstance>

The `IBehaviorInstance` interface represents a behavior on an `IInstance`.

Many behaviors return a more specific class deriving from `IBehaviorInstance` to add APIs specific to the behavior. See the [Behavior instances reference](#) for more information.

## Getting an IBehaviorInstance

Behavior instances are typically accessed via the `behaviors` property of `IInstance`, followed by the name of the behavior. Some example code is shown below.

```
const mySpriteInst = runtime.objects.Sprite.getFirstInstance();
const myBehaviorInst = mySpriteInst.behaviors.Platform;
// ... do something with myBehaviorInst ...
```

## IBehaviorInstance APIs

### `addEventListener(type, func, capture)`

### `removeEventListener(type, func, capture)`

Add or remove an event handler for a particular type of event fired by an addon's script interface. An event object is passed as a parameter to the handler function. See [behavior instance event](#) for standard event object properties. For information on which events are fired by specific addons and which additional event object properties are available, see the documentation on each addon's script interfaces.

### `dispatchEvent(e)`

Dispatch an event, firing any handler functions that have been added for the event type. You can use `new C3.Event(eventName, isCancelable)` to create an event object that can be dispatched (e.g. `new C3.Event("arrived", true)`), and add any extra properties relevant to your event to that object. This can also be used by the [addon SDK](#) to cause your addon to fire an event in the script interface, e.g.:

```
const e = new C3.Event("arrived", true);
this.GetScriptInterface().dispatchEvent(e);
```

## instance

A reference to the [IInstance](#) representing the object instance this behavior instance is affecting.

---

## **behavior**

A reference to the [IBehavior](#) representing the kind of behavior, e.g. Solid or Physics.

---

## **behaviorType**

A reference to the [IBehaviorType](#) the behavior instance belongs to.

---

## **runtime**

A reference back to the [IRuntime](#) interface.