

OBJECT CLASS SCRIPT INTERFACE

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

The `IObjectClass` script interface represents an object class in the project, which can be either an object type or a family. It is the base class for both `IObjectType` and `IFamily`.

For the most part, object types and families work the same and can be treated equally, hence most APIs being available on the `IObjectClass` base class. However there are some differences, such as only being able to create an instance of a specific object type, hence the `createInstance()` method only being available for `IObjectType`.

Getting an IObjectClass

References to the project's object classes are typically accessed through the `IRuntime` interface `objects` property. This covers both object types and families, hence everything under that property is an `IObjectClass`. For example `runtime.objects.Sprite` would refer to the `IObjectType` interface for the *Sprite* object type, and `runtime.objects.Family1` would refer to the `IFamily` interface for the family *Family1*, assuming they were added to the project.

*Try not to confuse object classes with object instances. A common mistake is to try to use something like `runtime.objects.Sprite.x` to get the X co-ordinate of a *Sprite* instance. However `runtime.objects.Sprite` is an `IObjectClass`, which does not have a position. First add another call to get an instance before trying to read instance properties, for example `runtime.objects.Sprite.getFirstInstance().x`.*

Object class events

The following events can be listened for using the `addEventListener` method.

"instancecreate"

Fired whenever a new instance belonging to this object class is created. The event object has an `instance` property referring to the `IInstance` (or derivative) that was created.

"hierarchyready"

Fired for the root instance in a hierarchy after all instances have finished creating - see the `IWorldInstance` event for more details. When fired on an `IObjectClass`, the event object has an `instance` property referring to the `IWorldInstance` (or derivative) that was created.

"instancedestroy"

Fired whenever any instance belonging to this object class is destroyed. After this event, all references to the instance are now invalid, so any remaining references to the instance should be removed or cleared to `null` in this event. Accessing an instance after it is destroyed will throw exceptions or return invalid data. The event object has an `instance` property referring to the `IInstance` (or derivative) that was destroyed. It also has an `isEndingLayout` property to indicate if the object is being destroyed because it's the end of a layout, or destroyed for other reasons.

Object class APIs**runtime**

A reference back to the `IRuntime` interface.

plugin

A reference to the `IPlugin` interface (or derivative) this object class was created from.

name

A read-only string of the object class's name.

addEventListener(eventName, callback)**removeEventListener(eventName, callback)**

Add or remove a callback function for an event. See *Object class events* above for more information.

getAllInstances()

Return an array of all instances of this object class.

getFirstInstance()

Return the first instance in the array returned by `getAllInstances()`, or `null` if no instances exist.

***instances()**

Iterates over all the object class's instances.

getPickedInstances()

Return an array of instances that have been picked by the event's conditions. This is only useful with scripts in event sheets.

getFirstPickedInstance()

Return the first instance that has been picked by the event's conditions, or `null` if none. This is only useful with scripts in event sheets.

***pickedInstances()**

Iterates over the instances that have been picked by the event's conditions. This is only useful with scripts in event sheets.

getPairedInstance(inst)

Get an instance of this object class at the same index (IID) as the given instance. If there are fewer instances of this object class than the given instance's object class, the index is allowed to wrap around. This allows identifying the same instance that the event system would pair with the given instance.

callCustomAction(name, instances, ...params)

Call one of this object class's custom actions in an event sheet by a case-insensitive string of its name. `instances` is an iterable (such as an array or `Set`) of instances of the object class that will be picked when the custom action runs (since custom actions may run with multiple instances picked).

It is more efficient to call a custom action once with multiple instances picked, than to call the custom action repeatedly with one instance picked each time.

At least one instance must be provided, and all instances must belong to either the object type or family. Each subsequent parameter is passed to the custom action. There must be at least as many parameters as the custom action uses, although any additional parameters will be ignored.