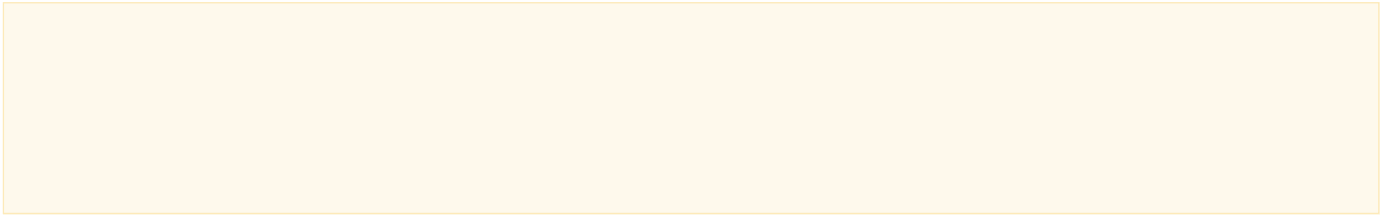
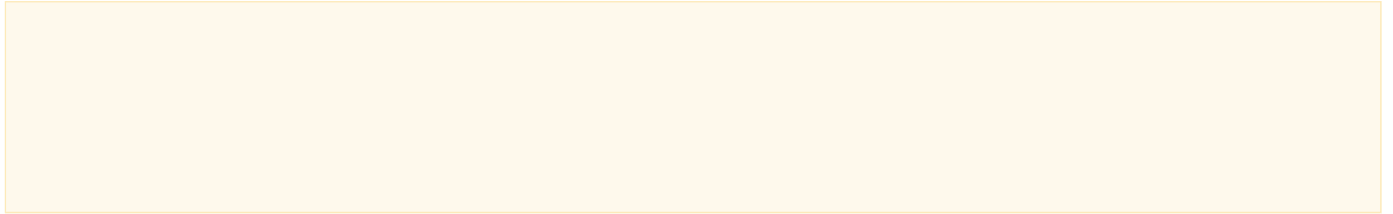




- 
- 
- 
- 
- 
-





- 
- 
- 
- 
- 
- 
- 
- 
-



---

```
{
  "is-c3-addon": true,
  "sdk-version": 2,
  "type": "plugin",
  "name": "My custom plugin",
  "id": "MyCompany_MyAddon",
  "version": "1.0.0.0",
  "author": "Scirra",
  "website": "https://www.construct.net",
  "documentation": "https://www.construct.net",
  "description": "Example custom Construct 3 plugin.",
  "editor-scripts": [
    "plugin.js",
    "type.js",
    "instance.js"
  ],
  "file-list": [
    "c3runtime/plugin.js",
    "c3runtime/type.js",
    "c3runtime/instance.js",
    "c3runtime/conditions.js",
    "c3runtime/actions.js",
    "c3runtime/expressions.js",
    "lang/en-US.json",
    "aces.json",
    "addon.json",
    "icon.svg",
    "instance.js",
    "plugin.js",
    "type.js"
  ]
}
```

-----  
`true`  
-----

-----  
`"plugin"`

`"behavior"`

`"effect"`

`"theme"`  
-----  
-----

`MyCompany_MyAddon`  
-----  
-----  
-----

-----

-----

-----

-----

true

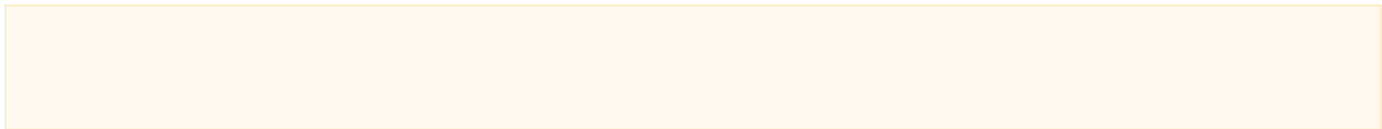
false

-----

-----

-----



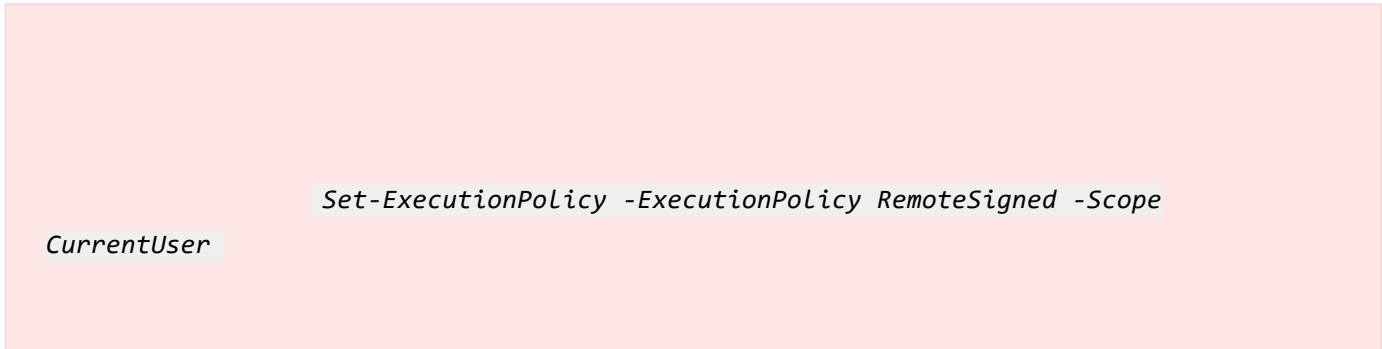


```
npm install -g typescript
```

```
tsc
```

```
tsc --
```

```
version
```



```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope  
CurrentUser
```



Ctrl Shift B





---

```
const PLUGIN_ID = "MyCompany_MyAddon";  
const PLUGIN_VERSION = "1.0.0.0"; // addon SDK v1 only  
const PLUGIN_CATEGORY = "general";
```

MyCompany\_MyAddon

---

"3d" "data-and-storage" "form-controls" "general" "input"  
"media" "monetisation" "platform-specific" "web" "other"

```
const PLUGIN_CLASS = SDK.Plugins.MyCompany_MyAddon = class MyCustomPlugin extends SDK.IPluginBase
```

```
class
```

- `PLUGIN_CLASS`

- `class`

	<code>Type</code>	<code>Instance</code>
<code>AudioPlugin</code>	<code>AudioType</code>	<code>AudioInstance</code>



```
"file-list"
```

```
"editor-scripts"
```

```
this._info.SetC3RuntimeScripts()
```

```
this._info
```

```
IPluginInfo
```

```
this._info.SetProperties
```

```
PluginProperty
```

```
this._info.SetProperties([
    new SDK.PluginProperty("integer", "test-property", 0)
]);
```

---

```
const BEHAVIOR_ID = "MyCompany_MyAddon";  
const BEHAVIOR_VERSION = "1.0.0.0"; // addon SDK v1 only  
const BEHAVIOR_CATEGORY = "general";
```

---

MyCompany\_MyAddon

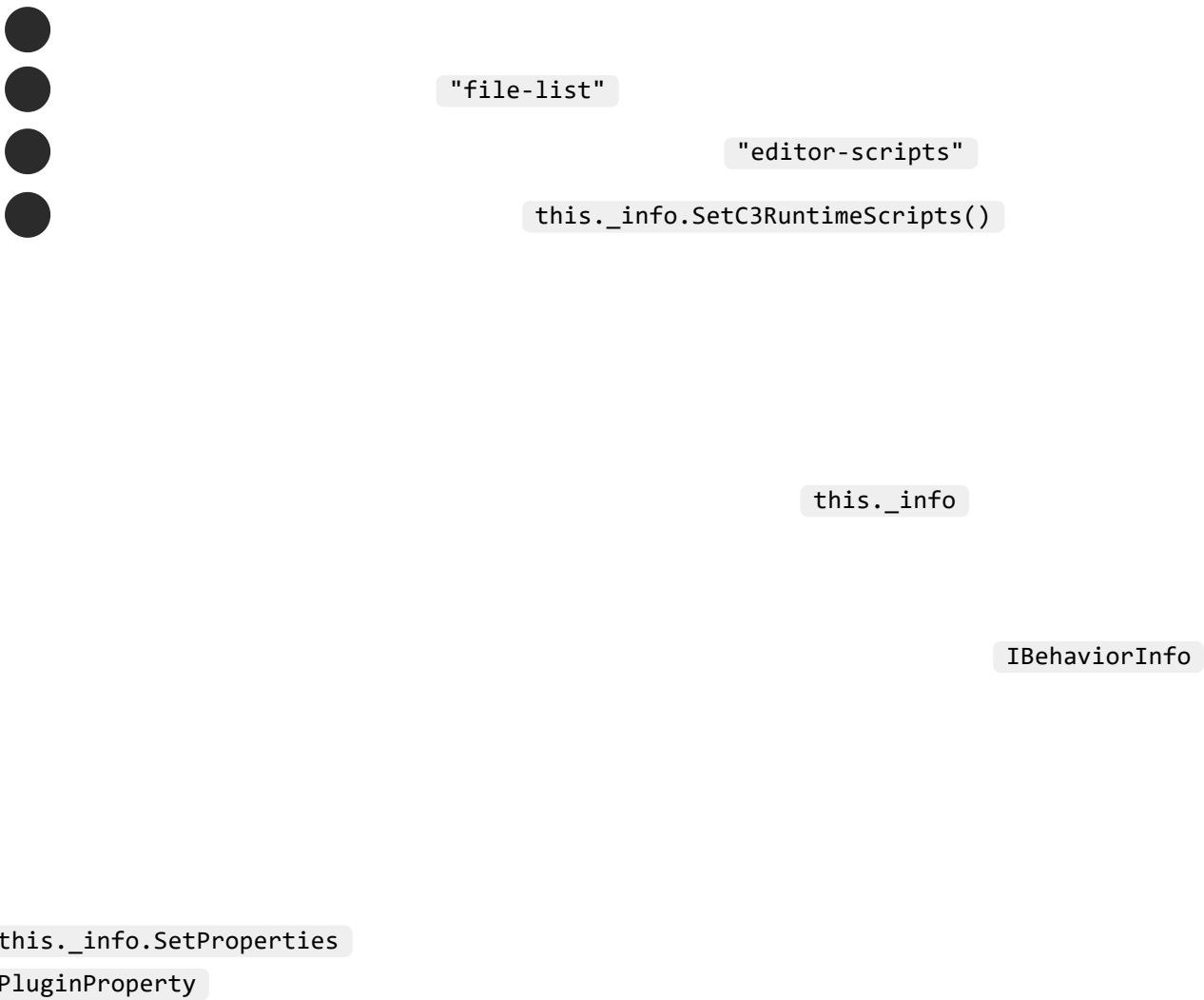
---

"attributes" "general" "movements" "other"

```
const BEHAVIOR_CLASS = SDK.Behaviors.MyCompany_MyAddon = class MyCustomBehavior extends SDK.IBeh
```

- `BEHAVIOR_CLASS`
- `class`

	<code>BulletBehavior</code>	<code>BulletType</code>	<code>BulletInstance</code>
		<code>Type</code>	<code>Instance</code>



```
this._info.SetProperties([  
    new SDK.PluginProperty("integer", "test-property", 0)  
]);
```



-----

-----

"distortion" "mask" "normal-mapping" "tiling" "other" "3d" "blend" "color"

-----

["webgl"] "webgl2" "webgpu"

-----

-----

samplerDepth

-----

cross-sampling false

cross-sampling true

-----

---

seconds

---

---

false

true

"horizontal" "vertical"  
"extend-box": { "horizontal": 30, "vertical": 30 }

---

parameters

---

---

id

"red,green,blue"

---

"float"

"percent"

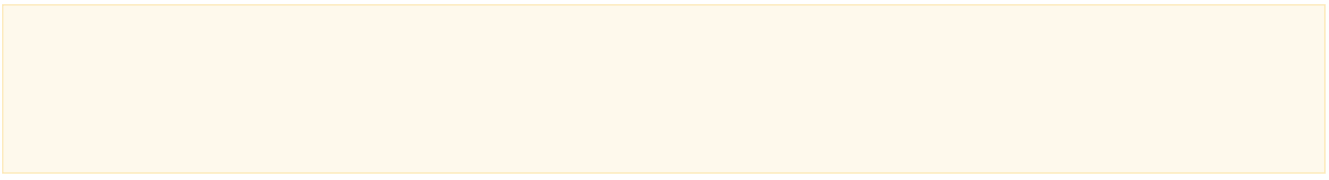
"color"

---

---

vec3

float



---

true



---

`"webgl2"``"supported-renderers"``"supported-renderers": ["webgl", "webgl2"]``effect.fx``effect.webgl2.fx`

- `#version 300 es`

- `varying in vTex`

- `gl_FragColor` `out lowp vec4 outColor;`
- `texture2D()` `texture()`  
`dFdx()` `dFdy()`  
`textureGrad()`

- 
- 
- 

```
#extension GL_EXT_frag_depth : enable
#extension GL_EXT_shader_texture_lod : enable
#extension GL_OES_standard_derivatives : enable

// now you can use gl_FragDepthEXT, dFdx, dFdy, texture2DGradEXT etc.
```

gl\_FragColor

vTex

vTex

background

true

blends-

true

uses-depth

samplerBack

animated

true

highp

`#ifdef GL_FRAGMENT_PRECISION_HIGH`

highp

mediump



```
lowp vec4 front = texture2D(samplerFront, vTex);
```

```
// sample next pixel to the right
lowp vec4 next = texture2D(samplerFront, vTex + vec2(pixelSize.x, 0.0));
```

n vTex

```
mediump vec2 n = (vTex - srcStart) / (srcEnd - srcStart);
lowp vec4 back = texture2D(samplerBack, mix(destStart, destEnd, n));
```

r

zNear zFar

```
mediump vec2 n = (vTex - srcStart) / (srcEnd - srcStart);
mediump float depthSample = texture2D(samplerDepth, mix(destStart, destEnd, n)).r;
mediump float zLinear = zNear * zFar / (zFar + depthSample * (zNear - zFar));
```

```
mediump vec2 srcOriginSize = srcOriginEnd - srcOriginStart;
mediump vec2 n = ((vTex - srcOriginStart) / srcOriginSize);
```

n

```
mediump vec2 srcOriginSize = srcOriginEnd - srcOriginStart;
mediump vec2 n = ((vTex - srcOriginStart) / srcOriginSize);
mediump vec2 l = mix(layoutStart, layoutEnd, n);
```

```
lowp vec4 front = texture2D(samplerFront, vTex);
lowp float a = front.a;

// unpremultiply
if (a != 0.0)
    front.rgb /= a;

// ...modify unpremultiplied front color...

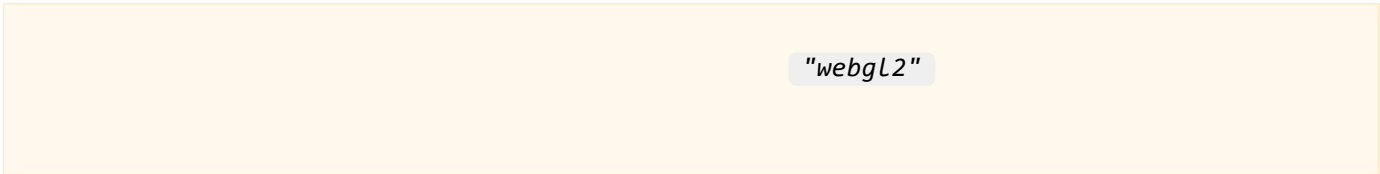
// premultiply again
front.rgb *= a;
```



"webgpu"

"supported-renderers"

"supported-renderers": ["webgl", "webgpu"]



"webgl2"



%%NAME%%



ShaderParams

```
%%NAME%%
```

---

```
@binding @group
```

```
%%SAMPLERFRONT_BINDING%% var samplerFront : sampler;
%%TEXTUREFRONT_BINDING%% var textureFront : texture_2d<f32>;
```

---

```
@binding @group
```

```
%%SAMPLERBACK_BINDING%% var samplerBack : sampler;
%%TEXTUREBACK_BINDING%% var textureBack : texture_2d<f32>;
```

---

```
@binding @group
```

```
%%SAMPLERDEPTH_BINDING%% var samplerDepth : sampler;
%%TEXTUREDEPTH_BINDING%% var textureDepth : texture_depth_2d;
```

---

```
FragmentInput
```

```
fragUV : vec2<f32>
```

```
vTex
```

```
@builtin(position) fragPos :
```

```
vec4<f32>
```

---

```
FragmentOutput
```

```
color : vec4<f32>
```

```
gl_FragColor
```

---

```
@binding @group
```

```

struct ShaderParams {
    setColor : vec3<f32>
};
%%SHADERPARAMS_BINDING%% var<uniform> shaderParams : ShaderParams;

```

### c3Params

```

srcStart          : vec2<f32>,
srcEnd            : vec2<f32>,
srcOriginStart   : vec2<f32>,
srcOriginEnd     : vec2<f32>,
layoutStart      : vec2<f32>,
layoutEnd        : vec2<f32>,
destStart        : vec2<f32>,
destEnd          : vec2<f32>,
devicePixelRatio : f32,
layerScale       : f32,
layerAngle       : f32,
seconds          : f32,
zNear            : f32,
zFar             : f32,
isSrcTexRotated  : u32

```

```
c3_getBackUV(input.fragPos.xy, textureBack)
```

```
c3_getDepthUV(input.fragPos.xy, textureDepth)
```

---

```

input.fragUV  c3_srcToNorm()
              srcStart  srcEnd  c3_normToSrc()
              srcOrigin  srcOriginStart
srcOriginEnd

```

---

```

srcOriginEnd  srcStart  srcEnd  srcOriginStart

```

---

```
input.fragUV
```

---

```

destStart  destEnd  srcStart  srcEnd

```

---

```
destStart  destEnd
```

---

---



---

```

textureDimensions()
    pixelSize

```

---

```

var front : vec4<f32> = textureSample(textureFront, samplerFront, input.fragUV);

```

```

// get width of a pixel in texture co-ordinates
var pixelWidth : f32 = c3_getPixelSize(textureFront).x;

// sample next pixel to the right
var next : vec4<f32> = textureSample(textureFront, samplerFront, input.fragUV + vec2<f32>(pixelW

```

```

c3_getBackUV()

```

```

var back : vec4<f32> = textureSample(textureBack, samplerBack, c3_getBackUV(input.fragPos.xy, t

```

```

c3_getDepthUV()

```

```

c3_linearizeDepth()

```

```
// sample depth buffer
var depthSample : f32 = textureSample(textureDepth, samplerDepth, c3_getDepthUV(input.fragPos.xy));

// linearize depth sample to Z distance
var zLinear : f32 = c3_linearizeDepth(depthSample);
```

c3\_srcOriginToNorm()

```
var n : vec2<f32> = c3_srcOriginToNorm(input.fragUV);
```

c3\_getLayoutPos()

```
var l : vec2<f32> = c3_getLayoutPos(input.fragUV);
```

c3\_unpremultiply()

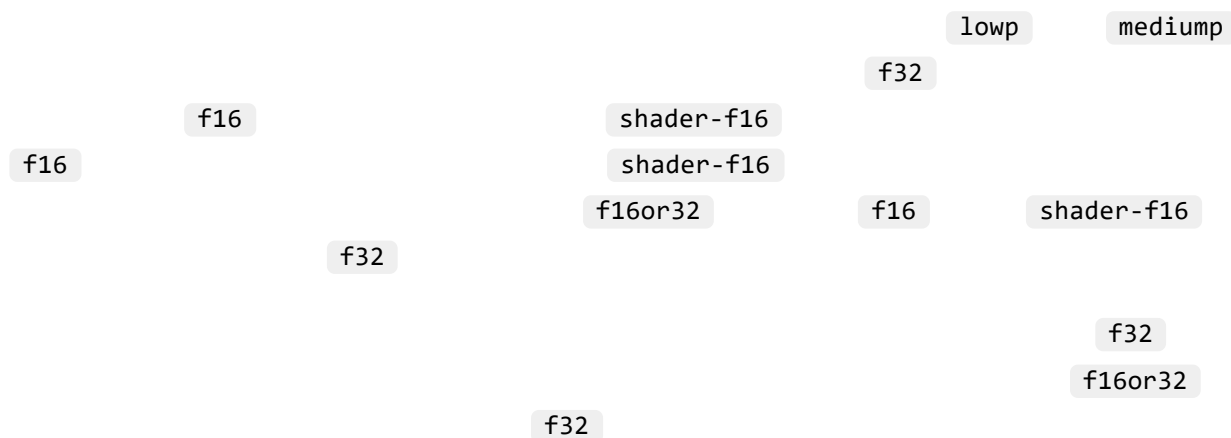
c3\_premultiply()

```
// sample front texture
var front : vec4<f32> = textureSample(textureFront, samplerFront, input.fragUV);

// unpremultiply
front = c3_unpremultiply(front);

// ...modify unpremultiplied front color...

// premultiply again
front = c3_premultiply(front);
```





f32

lowp

mediump

```
// At start of shader: get normalized source co-ordinates
// and then invert Y direction to match WebGL
var tex : vec2<f32> = c3_srcToNorm(input.fragUV);
tex.y = 1.0 - tex.y;

// ... rest of effect ...

// At end of shader: invert Y direction again and then
// calculate background sampling position
p.y = 1.0 - p.y;

var output : FragmentOutput;
output.color = textureSample(textureBack, samplerBack, mix(c3Params.destStart, c3Params.destEnd,
```

---

```
{
  "category1": {
    "conditions": [
      condition1,
      condition2,
      ...
    ],
    "actions": [
      action1,
      action2,
      ...
    ],
    "expressions": [
      expression1,
      expression2,
      ...
    ]
  },
  "category2": {
    "conditions": [
      ...
    ],
    "actions": [
      ...
    ],
    "expressions": [
      ...
    ]
  }
}
```

`"$schema"``"conditions"``"actions"``"expressions"`

```
{  
  "id": "every-tick",  
  "scriptName": "EveryTick"  
}
```

`id``scriptName``id``expressionName``returnType`

-----

expressionName

-----

-----

-----

-----

-----

-----

-----

-----

-----

false

-----

false

-----

true

async

returnType

scriptName

expressionName

-----

"number"

"string"

"any"

"any"

-----

true

"params"

"any"

`"params"`

```

{
  "id": "compare-two-values",
  "scriptName": "Compare",
  "params": [
    { "id": "first-value", "type": "any" },
    { "id": "comparison", "type": "cmp" },
    { "id": "second-value", "type": "any" }
  ]
}

```

`"number"``"string"``"any"``"number"``"string"``"any"`

- `"number"`

- `"string"`

- `"any"`

- `"boolean"`

- `"combo"`

`"items"`

- `"combo-grouped"`

`"itemGroups"`

- "cmp"
- "object"
  - "allowedPluginIds"
- "objectname"
- "layer"
- "layout"
- "keyb"
- "instancevar"
- "instancevarbool"
- "eventvar"
- "eventvarbool"
- "animation"
- "objinstancevar"
  - "object"
    - "object"

---

"false"

"combo"

"number"

"boolean"

"true"

---

"combo"

---

"combo-grouped"

"id"

"items"

```
// example "combo-grouped" parameter definition
{
  "id": "dinosaur",
  "type": "combo-grouped",
  "itemGroups": [{
    "id": "theropods",
    "items": ["tyrannosaurus", "velociraptor", "deinonychus"]
  }, {
    "id": "sauropods",
    "items": ["diplodocus", "saltasaurus", "apatosaurus"]
  }
}
```

---

"object"

["Sprite"]

"string"



---

```
{
  "languageTag": "en-US",
  "fileDescription": "Strings for MyCustomPlugin.",
  "text": {
    "plugins": {
      "mycompany_myaddon": {
        "name": "My Custom Plugin",
        "description": "Description for my custom plugin.",
        "help-url": "https://www.scirra.com",
        "properties": {
          "test-property": {
            "name": "Test property",
            "desc": "A test number property. Displayed by 'A"
          }
        },
        "aceCategories": {
          "custom": "Custom"
        },
        "conditions": {
          "is-large-number": {
            "list-name": "Is large number",
            "display-text": "[i]{0}[/i] is a large number",
            "description": "Test if a number is greater than",
            "params": {
              "number": {
                "name": "Number",
```



`"name"``"description"``"help-url"``"name"``"description"``"help-url"``"properties"``"parameters"`

- `"name"`
- `"desc"`

```
new SDK.PluginProperty("integer", "test-property", 0)
```

`"properties"`

```
"test-property": {
    "name": "Test property",
    "desc": "A test number property. Displayed by 'Alert' action."
}
```

`"combo"``"items"`

```
new SDK.PluginProperty("combo", "timescale-audio", {
    initialValue: "off",
    items: ["off", "sounds-only", "sounds-and-music"]
})
```

```
"timescale-audio": {
    "name": "Timescale audio",
```

```

"desc": "Choose whether the audio playback rate changes with the time scale.",
"items": {
  "off": "Off",
  "sounds-only": "On (sounds only)",
  "sounds-and-music": "On (sounds and music)"
}
}

```

"link"

"link-text"

```

"make-original-size": {
  "name": "Size",
  "desc": "Click to set the object to the same size as its image.",
  "link-text": "Make 1:1"
}

```

"aceCategories"

"customCategory"

"My

custom category"

```

"aceCategories": {
  "customCategory": "My custom category"
}

```

"conditions"

"actions"

"expressions"

- "list-name"
- "display-text"
  - [b]
  - [i]
  - {0}
  - {1}
  - {my}
- "description"
- "description"
- "translated-name"

"params"

"name"

"desc"

"combo"

"items"

"combo"

"combo-grouped"

"itemGroups"

```
{
  "id": "is-large-number",
  "scriptName": "IsLargeNumber",
  "highlight": true,
  "params": [
    {
      "id": "number",
      "type": "number"
    }
  ]
}
```

```

"is-large-number": {
  "list-name": "Is large number",
  "display-text": "[i]{0}[/i] is a large number",
  "description": "Test if a number is greater than 100.",
  "params": {
    "number": {
      "name": "Number",
      "desc": "Number to test if greater than 100."
    }
  }
}

```

"combo-grouped"

```

{
  "id": "dinosaur",
  "type": "combo-grouped",
  "itemGroups": [{
    "id": "theropods",
    "items": ["tyrannosaurus", "velociraptor", "deinonychus"]
  }, {
    "id": "sauropods",
    "items": ["diplodocus", "saltasaurus", "apatosaurus"]
  }
}

```

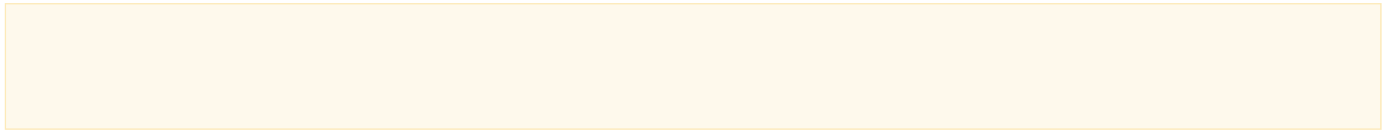
"params"

```

"dinosaur": {
  "name": "Dinosaur",
  "desc": "Choose a dinosaur",
  "itemGroups": {
    "theropods": {
      "name": "Theropods",
      "items": {
        "tyrannosaurus": "Tyrannosaurus",
        "velociraptor": "Velociraptor",
        "deinonychus": "Deinonychus"
      }
    },

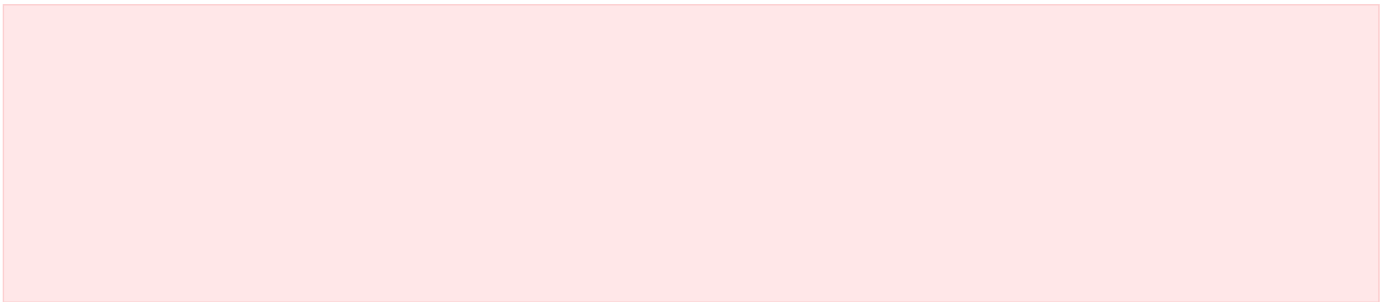
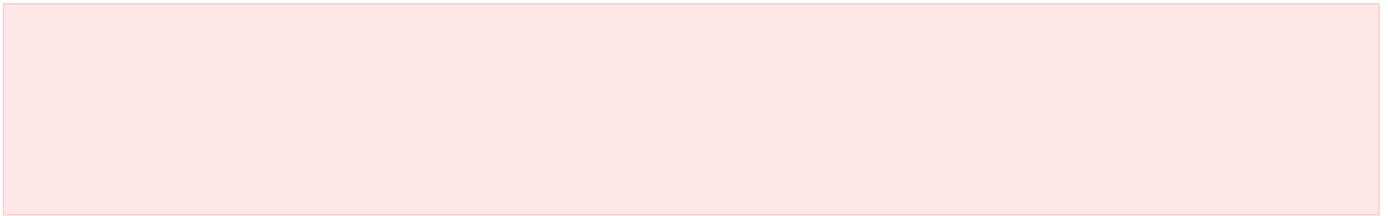
```

```
    "sauropods": {  
        "name": "Sauropods",  
        "items": {  
            "diplodocus": "Diplodocus",  
            "saltasaurus": "Saltasaurus",  
            "apatosaurus": "Apatosaurus"  
        }  
    }  
}
```



- 
-





fetch()

```
this._info.SetDOMSideScripts(["c3runtime/domSide.js"]);
```

```
addon.json
```

```
<button>
```

```
GetDebuggerProperties()
```

```
{ title, properties }
```

```
title
```

```
properties
```

```
{ name, value
```

```
}
```

```
onedit
```

```
onedit
```

\$

"plugins.sprite.debugger.foo"

foo

"\$foo"

- 
- 
- 
- 

```

GetDebuggerProperties()
{
    const prefix = "plugins.sprite.debugger.animation-properties";
    return [{
        title: prefix + ".title",
        properties: [
            {name: prefix + ".current-animation", value: this._currentAnimation.Get},
            {name: prefix + ".current-frame", value: this._currentFrame},
            {name: prefix + ".is-playing", value: this._isAnimationPlaying},
            {name: prefix + ".speed", value: this._currentAnimation.Speed},
            {name: prefix + ".repeats", value: this._currentAnimation.Repeats}
        ]
    }];
}

```



import

export



`this._info.SetRuntimeModuleMainScript("c3runtime/main.js")`



`import "./instance.js";`



`file-list`

`SetRuntimeModuleMainScript()`

`import`

`export``this._info.AddC3RuntimeScript("c3runtime/mymodule.js");``"file-list"``import * as MyModule from "./mymodule.js";``import``fetch()`

```
this._info.SetDOMSideScripts(["c3runtime/domSide.js"]);
```

```
addon.json
```

```
<button>
```

```
_getDebuggerProperties()
    { title, properties }
properties
}
onedit
    { name, value
```

```
onedit
```

```
$
```

```
"plugins.sprite.debugger.foo"
```

```
foo
```

```
"$foo"
```

- 
- 
- 
- 

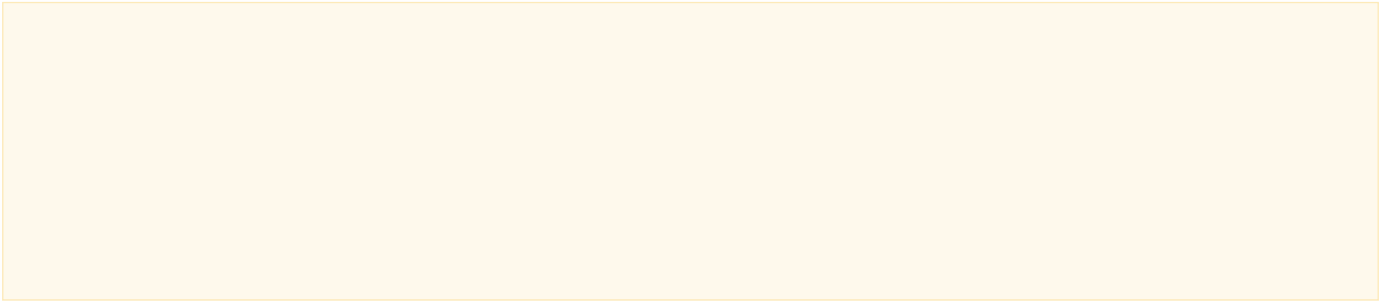
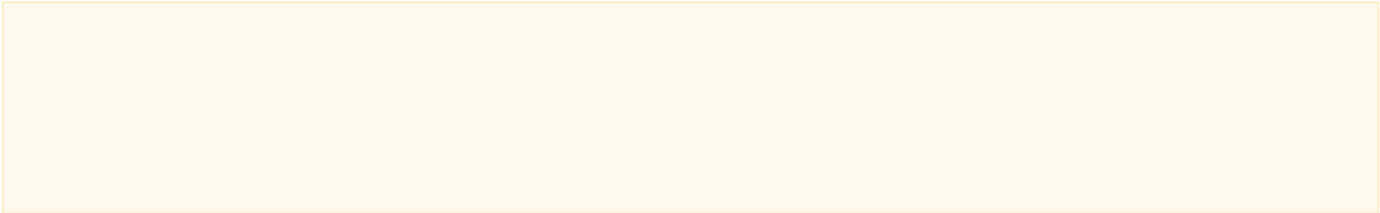
```
_getDebuggerProperties()
{
    const prefix = "plugins.sample-plugin.debugger";
    return [{
        title: prefix + ".title",
        properties: [
            {name: prefix + ".speed",      value: this.speed,      onedit: v => this
            {name: prefix + ".angle",     value: this.angle,     onedit: v => this
        ]
    }];
}
```

---



---

---



---

---



=

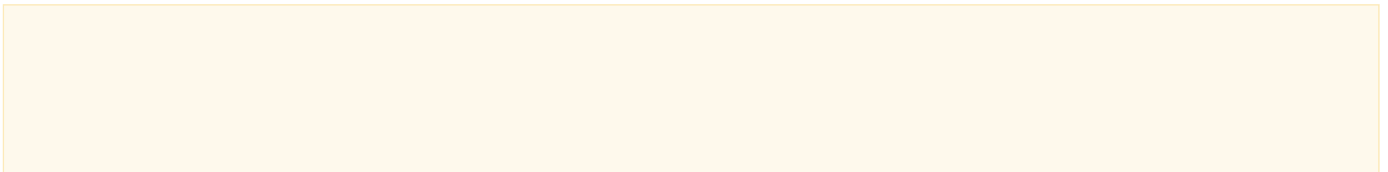
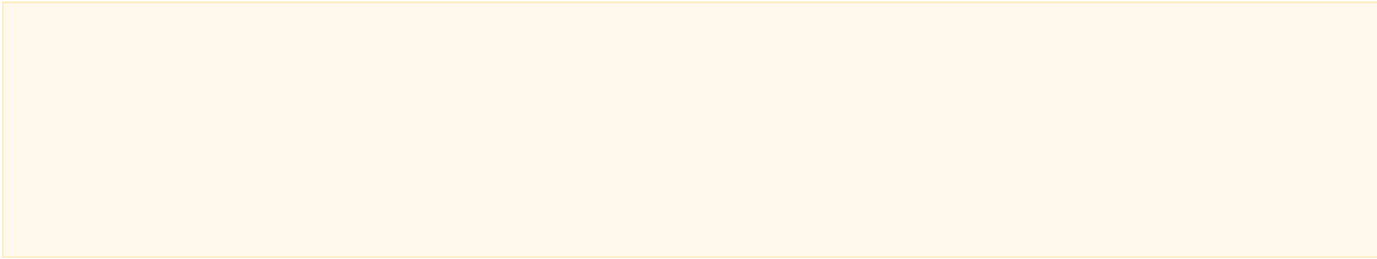
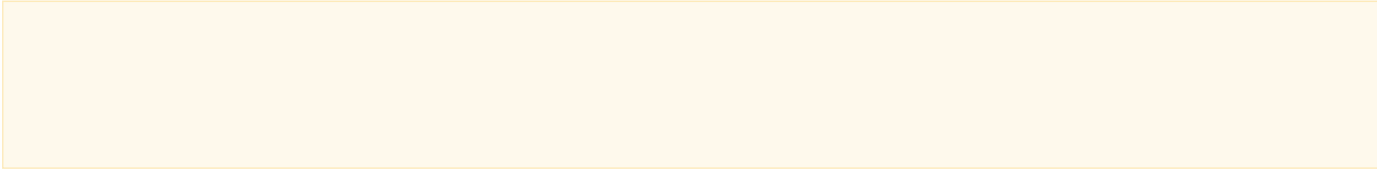
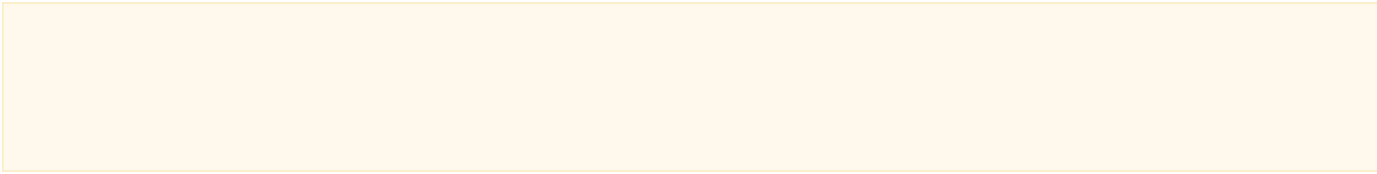
-----



-----

-----

-----





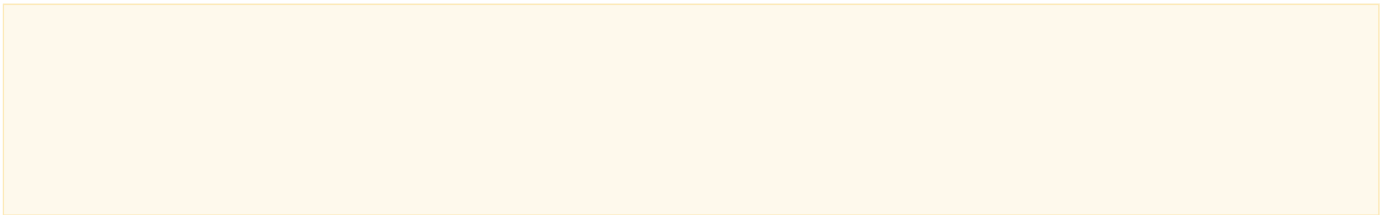


- 
- 
- 
- 

- 
-

- `AddFileDependency()`  
`"wrapper-extension"`

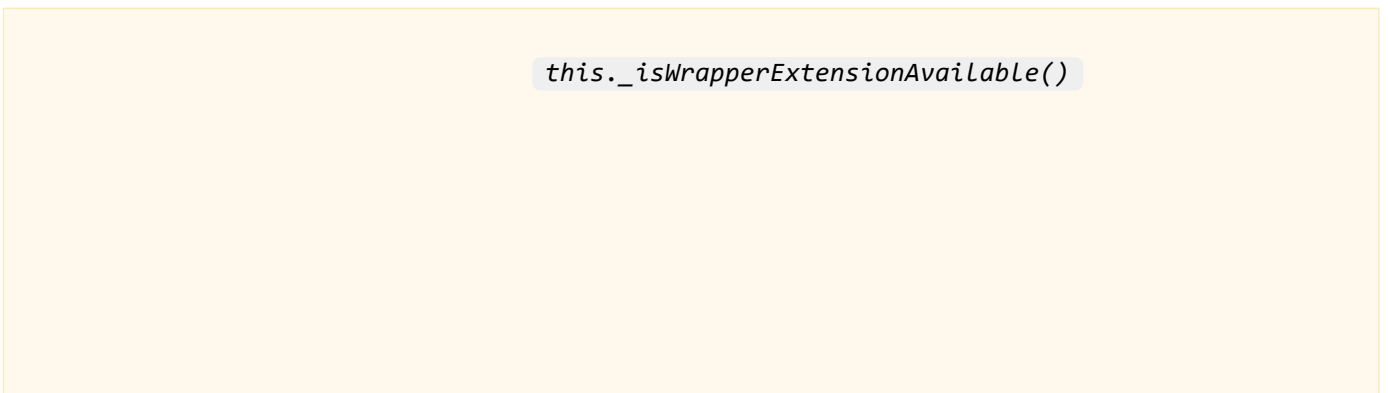
- 



```
iApplication->RegisterComponentId()
```

```
WrapperExtension
```

```
this.SetWrapperExtensionComponentId()
```



```
this._isWrapperExtensionAvailable()
```

```
this._sendWrapperExtensionMessage("message-id", [params...])
```

```

                                this._addWrapperExtensionMessageHandler("message-
id", handlerFunc)

```

```
// C++
SendWebMessage("message-id", {
    { "sampleString1", "Hello world!" },
    { "sampleString2", "Foo bar baz" },
});
```

```
double
```

```
std::string
```

```
char*
```

```
result["sampleString1"]
```

```
HandleWebMessage()
```

```
this._sendWrapperExtensionMessageAsync()
```

```
this._sendWrapperExtensionMessage()
```

```
asyncId
```

SendAsyncResponse()

asyncId

```
SendAsyncResponse({
    { "sampleString1",          "Hello world!" },
    { "sampleString2",          "Foo bar baz" },
}, asyncId);
```

SendWebMessage()

*\_sendWrapperExtensionMessageAsync()**\_sendWrapperExtensionMessageAsync()*

wchar\_t

std::wstring

LPCWSTR

char

std::string

char\*

Utf8ToWide()

std::string

std::wstring

`c_str()``WideToUtf8()``std::wstring``std::string``MessageBoxA()``#import``[alert``setMessageText:nsTitle]``std::string``NSString``NSString``NSString``std::string``NSString``std::string`

```
std::string message = "Hello world";
```

```
// Convert std::string to NSString
```

```
NSString* nsMessage = [NSString stringWithUTF8String:message.c_str()];
```

```
// Convert NSString back to std::string
```

```
message = std::string([nsMessage UTF8String]);
```



- `_WIN32`
- `__APPLE__`
- `__linux__`



- 
- 
- 

```
this._info.SetIcon("icon.png", "image/png");
```



```
this._info.SetIsSingleGlobal(true);
```



○



id

○



c2id





`"sdk-version"`

`"version"`

`extends`

C3.SDKInstanceBase

globalThis.ISDKInstanceBase

properties

this.\_getInitProperties()

```

// Instance constructor in Addon SDK v1
constructor(inst, properties)
{
    super(inst);

    if (properties)
    {
        // ... read properties ...
    }

    // ... rest of constructor ...
}

// Instance constructor in Addon SDK v2
constructor()
{
    super();

    const properties = this._getInitProperties();
    if (properties)
    {
        // ... read properties ...
    }

    // ... rest of constructor ...
}

```

super()

```

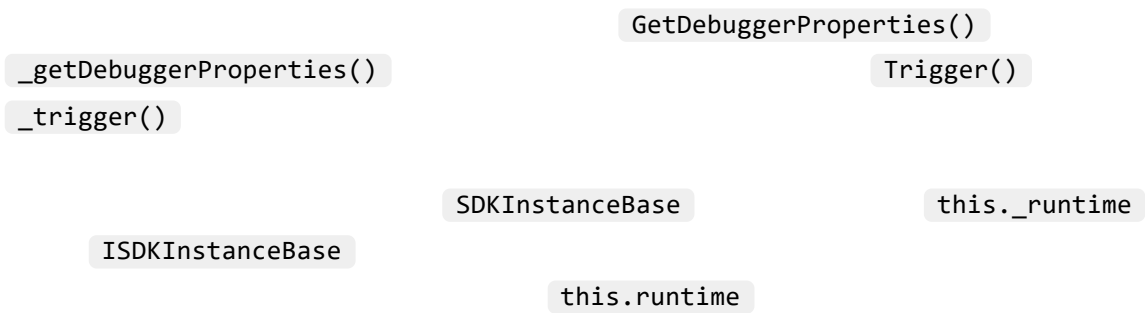
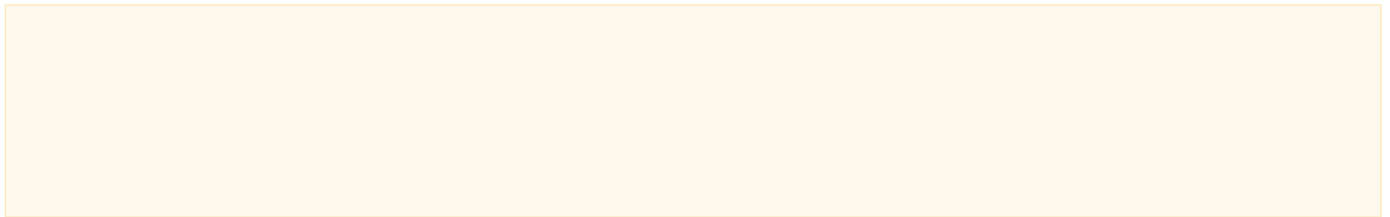
// In ISDKDOMPluginBase or ISDKDOMInstanceBase constructor:
super({ domComponentId: DOM_COMPONENT_ID });

// In ISDKInstanceBase constructor for a wrapper extension:

```

```
super({ wrapperComponentId: "my-extension" });
// (instead of calling SetWrapperExtensionComponentId())
```

```
GetScriptInterfaceClass()
```



PLUGIN\_VERSION

```
this._info.SetVersion(PLUGIN_VERSION);
```

window

self

globalThis

import

export

import

export

- 
-

- 
- 
- 
- 
- 
-



stylesheets

- 

- 

- 

-









```
Access-Control-Allow-Origin: *
```

```
add_header Access-Control-Allow-Origin *;
```

```
http://localhost
```



---

`IBehaviorInstanceBase`

`IBehaviorInstanceBase`

---

---

---

---

---

---

---

---

IInstanceBase

IInstanceBase

IInstanceBase

---

---

---

---

---

---

---

---

---

---

-----

-----



---

IWorldInstanceBase

IWorldInstanceBase

---

iRenderer

iDrawParams

---

null

---

```
                                false    IsOriginalSizeKnown()  
                                true     IsOriginalSizeKnown()  
GetOriginalWidth()    GetOriginalHeight()
```

---

```
                                false    HasDoubleTapHandler()  
true    HasDoubleTapHandler()    OnDoubleTap()
```

---

Color

unpremultiply()

```
new SDK.Color();  
new SDK.Color(r, g, b, a);
```

Color

---

---

---

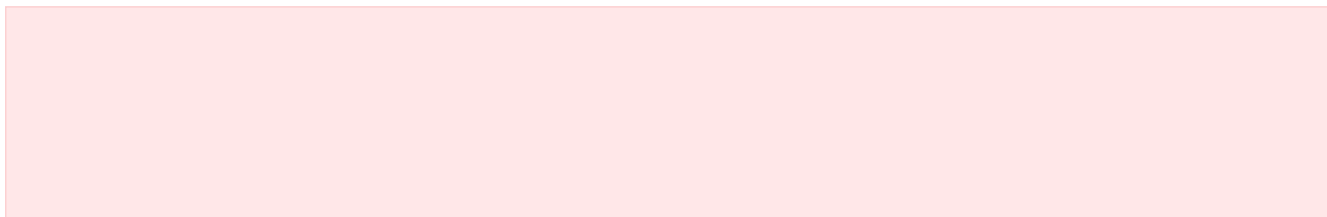
SDK.Color

SDK.Color

SDK.Color

SDK.Color

SDK.Color



---

Quad

```
new SDK.Quad();  
new SDK.Quad(tlx, tly, trx, try_, brx, bry, blx, bly);
```

Quad

try

try\_

---

---

SDK.Quad

---

angle



---

Rect

```
new SDK.Rect();  
new SDK.Rect(left, top, right, bottom);
```

Rect

---

---

SDK.Rect

SDK.Rect

---



x

y

SDK.Rect

---

IDrawParams

Draw()

Draw()

---

---

---

IWebGLRenderere

Draw()



SetColor()

SetColorRgba()



SetTexture()

Draw()

Rect()

Quad()

Rect2()

Quad3()

SDK.Quad

SDK.Rect

SDK.Quad

PopLineWidth()

"square"

PopLineCap()

"butt"

UpdateTexture()

height

opts

width

- wrapX "clamp-to-edge" "repeat" "mirror-repeat"
- wrapY wrapX
- sampling "nearest" "bilinear" "trilinear"
- pixelFormat "rgba8" "rgb8" "rgba4" "rgb5\_a1" "rgb565"
- mipMap
- mipMapQuality mipMap "default" "low" "high"

```
        CreateDynamicTexture()  
            HTMLImageElement HTMLVideoElement  
HTMLCanvasElement ImageBitmap OffscreenCanvas ImageData  
            HTMLImageElement HTMLVideoElement  
HTMLCanvasElement ImageBitmap OffscreenCanvas
```

- `premultiplyAlpha`

---

```
CreateDynamicTexture()
```

---

IWebGLText

CreateWebGLText()

---

IWebGLText

Release()

---

fillStyle



"center"

"right"

"left"

"center"

"bottom"

"top"

"character"

"word"

zoomScale

SetTextUpdateCallback()

null

GetTexture()

---

`GetTexture()`

---

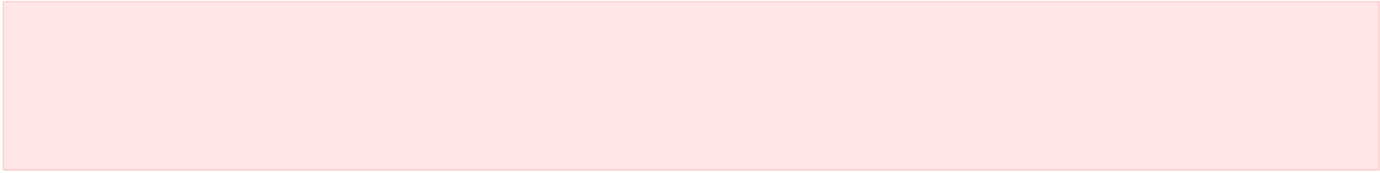
---

IWebGLTexture

---

---

ILang



---

Get(".bar")

PushContext("foo")

Get("foo.bar")

PopContext()

---

---

*self.Lang()*

---

IZipFile

---

---

---

---

null

---

null

---

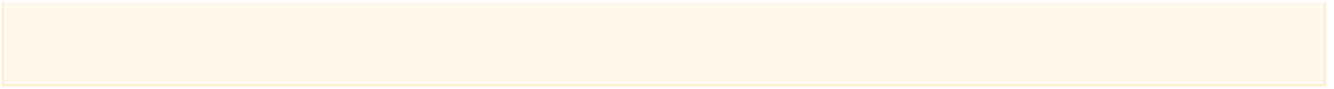
---

---

Blob

*name*

*LastModified*



---

`IZipFileEntry`

---

**IEventBlock**

```
// Note: this code is assumed to be in an async function
// First get the associated event sheet for the layout view
const eventSheet = layoutView.GetLayout().GetEventSheet();
if (eventSheet) // note the layout may not have an event sheet
{
    // Get the IObjectType for the System plugin
    const systemType = eventSheet.GetProject().GetSystemType();

    // Create an empty event block at the root level of the event sheet
    const eventBlock = await eventSheet.GetRoot().AddEventBlock();

    // Add an 'On start of layout' condition
    eventBlock.AddCondition(systemType, null, "on-start-of-layout");

    // Example code for adding a 'Set position' action
    //eventBlock.AddAction(iObjectType, null, "set-position", [100, 200]);
}
```



null

"on-start-of-layout"

"1+1"

---

IEventParentRow

---

---

`IEventSheet`

`GetRoot()`

---

ILayer

---

---

---

ILayout

---

---

---

---

---

null

---

IProject

---

null

---

null



---

name

GetName()

---

null

---

null

---

null

---

null

---

null

---

Blob

"general"

"sound"

"music"

"video"

"font"

"icon"

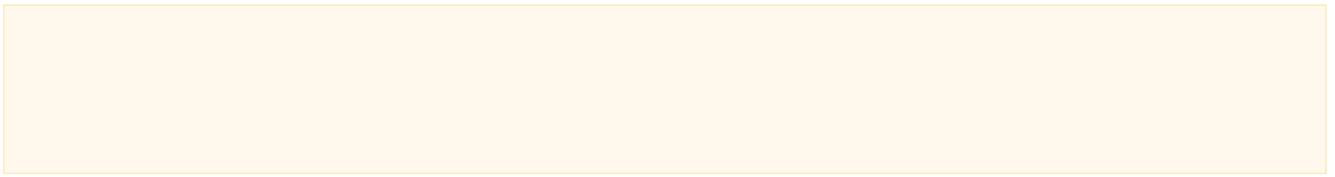
---

fileList

Blob

File

name



propertyId

instances



---

`IProjectFile`

---

---

---

---

---

IAnimation

---

---

---

---

---

- - 
  - 
  -
-

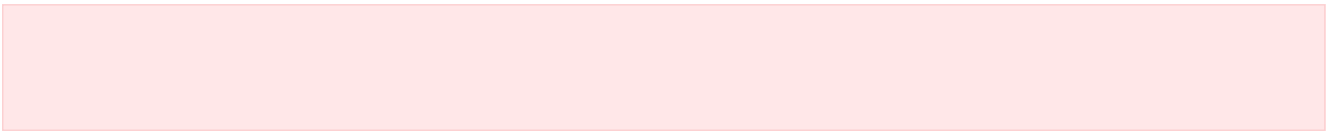
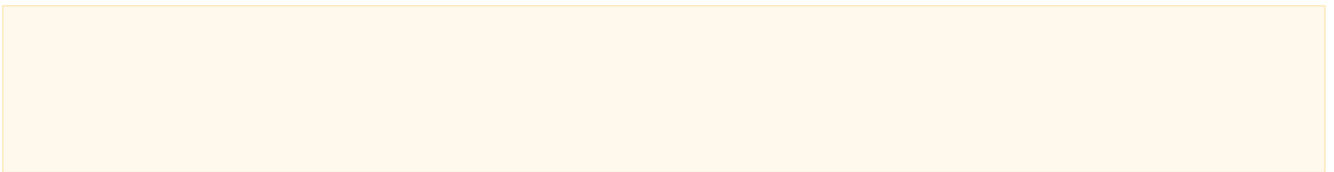
---

---

---

---

---



---

IAnimationFrame

IAnimationFrame

*IAnimationFrame*

GetTexture()

---

---

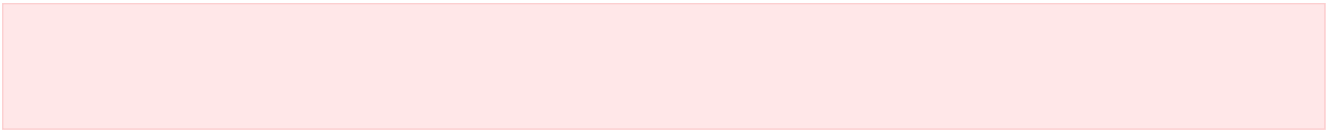
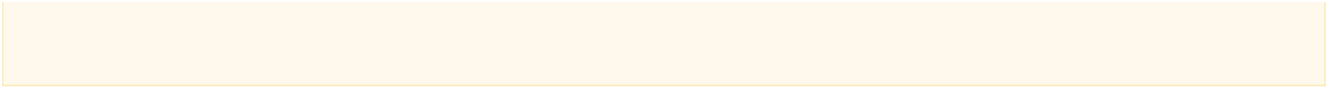
null

GetCachedWebGLTexture()

GetCachedWebGLTexture()

null





---

`IBehaviorInstance`

---

---

---

---

---

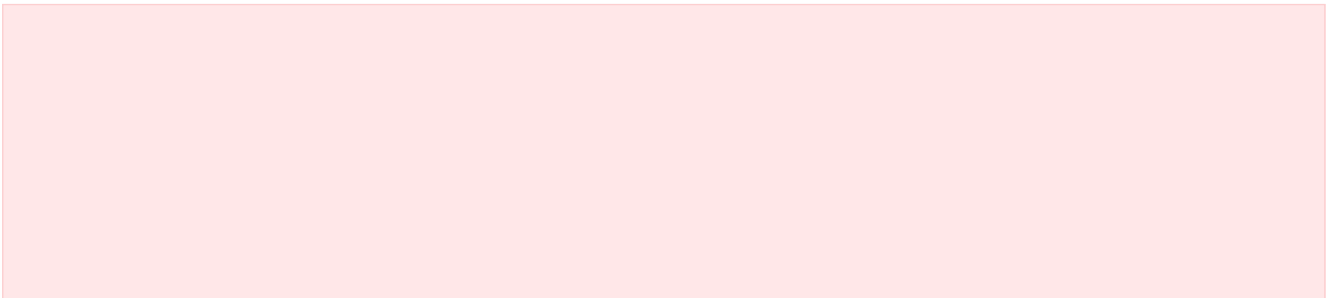
---

---

---

`MyCustomBehaviorInstance`

`null`



---

IBehaviorType

---

---



---

ICollisionPoly

---

---

---

---

---

---

IContainer

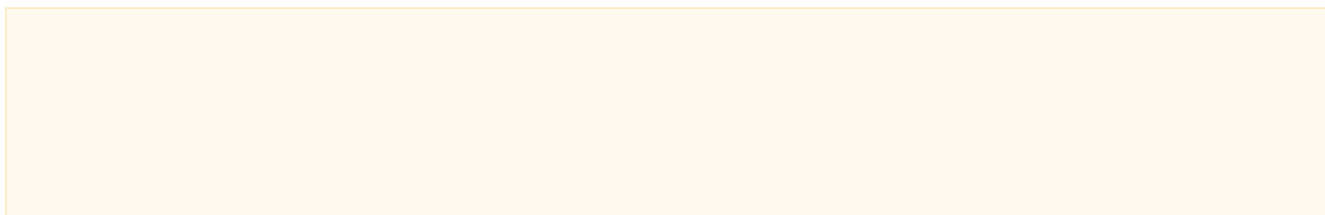
---

"normal"

"all"

"wrap"

---



---

IFamily

CreateFamily()

---

---

---

IImagePoint

---

---

---

---

---

IObjectClass

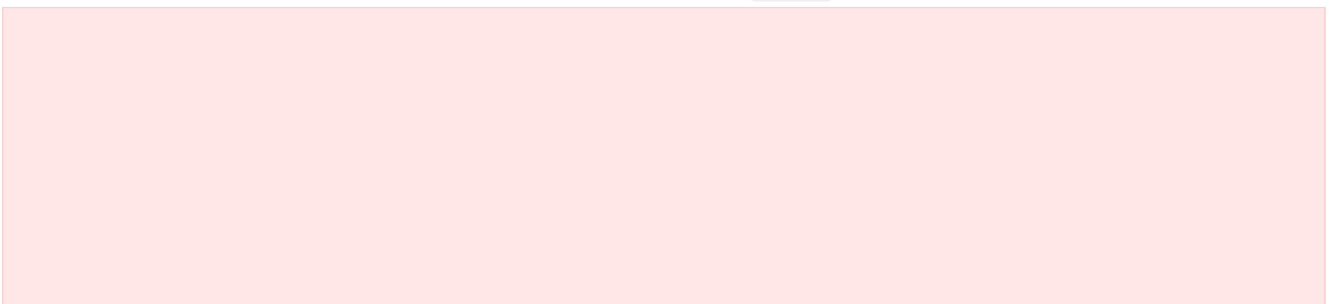
IObjectClass

IObjectClass

-----  
`IObjectInstance`  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----

`MyDrawingInstance`

`null`





"world"

---

---

---

null

---



---

IWorldInstance

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Draw()





---

SDK.UI.Util

---

true

false

opts

AddDragDropFileHandler()

- isZipFormat

true

file

false

file

Blob

- toLayoutView

true

opts

false

callback

async function(filename, file, opts)

file

isZipFormat

opts

toLayoutView

- layoutView

- `clientX`     `clientY`

- `layoutX`     `layoutY`



`longtext`

`null`

---

```
C3SDK_ListAddonIDs(addonType)
```

```
"plugin" "behavior"
```

```
C3SDK_ListAddonIDs("plugin")
```

```
C3SDK_ListACEIDs(addonType, addonId, aceType)
```

- ```
"plugin" "behavior"
```
- ```
C3SDK_ListAddonIDs()
```
- ```
"actions" "conditions" "expressions"
```

```
C3SDK_ListACEIDs("plugin", "Sprite", "actions")
```

---

`this._info`

---

---

---

---

`BEHAVIOR_VERSION`

---

`"attributes"`

`"general"`

`"movements"`

`"other"`

`BEHAVIOR_CATEGORY`

`"image/svg+xml"`

`"icon.svg"`

`SetIcon("icon.png", "image/png")`

---



false  
true

---

SetIsDeprecated(true)

---

false

---

---

---

---

---

---

---

---

---

---

`instance` `behaviorType` `behavior` `opts`

```
this._info.SetScriptInterfaceNames({  
    instance: "IBulletBehaviorInstance"  
});
```

---

```
this._info.SetTypeScriptDefinitionFiles(["c3runtime/IBulletBehaviorInstance.d.ts"]);
```

---

`this._info`

---

`PLUGIN_VERSION`

---

`PLUGIN_CATEGORY`

`"data-and-storage"`

`"form-controls"`

`"general"`

`"input"`

`"media"`

`"monetisation"`

`"platform-specific"`

`"web"`

`"other"`

---

`"object"`

`"world"`

`SDK.IWorldInstanceBase`

`SDK.IInstanceBase`

`Draw()`

---

"image/svg+xml"

"icon.svg"

setIcon("icon.png", "image/png")

---

"world"

true

---

"world"

true

---

"world"

true

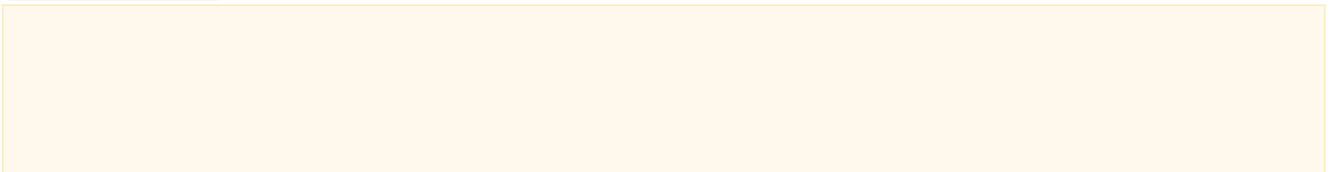
---

"world"

true

---

"default.png"



"world"

true

---

SetIsDeprecated(true)

true

"object"

true

"world"

Draw()

AddCommonZOrderACEs()

true

"world"

Draw()

AddCommonAppearanceACEs()

true

"world"

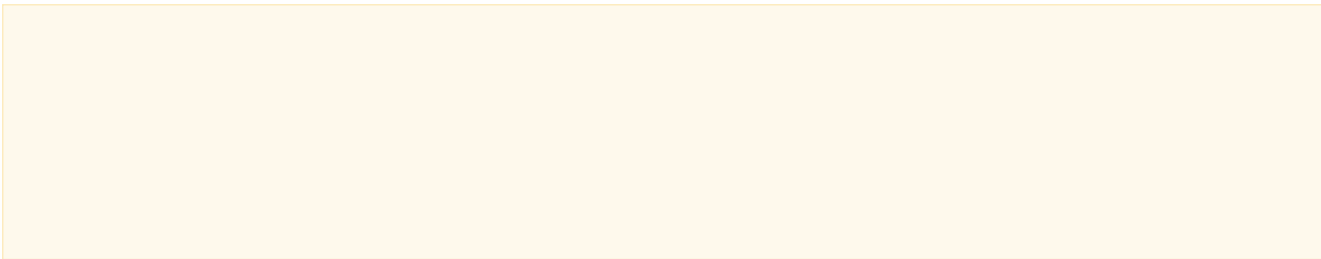
SetMustPreDraw(true)

true

---

false

---



---

"module"

type

---

true

<preference

name="GradlePluginGoogleServicesEnabled" value="true" />

---

opts

`instance``objectType``plugin`

```
this._info.SetScriptInterfaceNames({  
    instance: "ISpriteInstance"  
});
```

---

```
this._info.SetTypeScriptDefinitionFiles(["c3runtime/ISpriteInstance.d.ts"]);
```



---

```
this._info.SetProperties()
```

```
new SDK.PluginProperty(type, id, initialValue_or_options)
```

---

- "integer"
- "float"
- "percent"
- "text"
- "longtext"                      "text"
- "check"
- "font"

- "combo"

items

- "color"

[1, 0, 0]

- "object"

IObjectClass

- "group"

- "link"

linkCallback

- "info"

infoCallback

---

---

"float"

"combo"

---

"color"

"combo"

[1, 0, 0]

"combo"

"object"

"<world>"

"link"

callbackType

"link"

- "for-each-instance"

SDK.IWorldInstanceBase

- "once-for-type"

SDK.ITypeBase

---

"info"

---

"integer" "float" "percent" "text" "longtext" "check" "combo"  
"color" false true

---

```
AddFileDependency(opts)
```

```
AddCordovaPluginReference(opts)
```

```
AddRemoteScriptDependency
```

```
type
```

---

---

```
scriptType
```

```
<script>
```

```
"module"
```

---

```
<script>
```

```
importScripts()
```

---

```
rel="stylesheet">
```

```
<link
```

## AddFileDependency

```

this._info.AddFileDependency({
    filename: "mydependency.js",
    type: "external-dom-script"
});

```

*"file-list"*

"external-dom-script"

type "copy-to-output"

"image.png"

"copy-to-output"

fileType

"image/png"

type "external-dom-script"

"module"

<script src="filename.js"

type="module"></script>

```

    type      "wrapper-extension"
             "windows-x86"             "windows-x64" "windows-
arm64"      "xbox-uwp-x64"           "macos-universal"
             "linux-x64"  "linux-arm"  "linux-arm64"

```

#### AddCordovaPluginReference

```

this._info.AddCordovaPluginReference({
    id: "cordova-plugin-inappbrowser"
});

```

#### variables

```

[variableName, pluginProperty]
plugin

```

```

const property = new SDK.PluginProperty("integer", "test-property", 0);

this._info.SetProperties([
    property
]);

this._info.AddCordovaPluginReference({
    id: "cordova-plugin-inappbrowser",
    plugin: this,
    variables: [
        ["MY_VAR", property]
    ]
});

```

---

---

```
"1.0.4"
```

---

```
"ios"
```

```
"all"
```

```
"android"
```

---

```
pluginProperty]
```

```
[variableName,
```

```
SDK.PluginProperty
```

```
plugin
```

---

```
this
```

```
<resource file src="..." target="...">
```

```
AddCordovaResourceFile
```

```
this._info.AddCordovaResourceFile({  
    src: "myfile.txt"
```



```
});
```

```
<resource-file src="myfile.txt">
```

---

```
src
```

```
resource-file
```

---

```
target
```

```
resource-file
```

---

```
"ios"
```

```
"all"
```

```
"android"
```

```
https://example.com/api.js
```

- 
- 
-

`AddRemoteScriptDependency``"module"`

```
// Load remote "classic" script
this._info.AddRemoteScriptDependency("https://example.com/api.js");

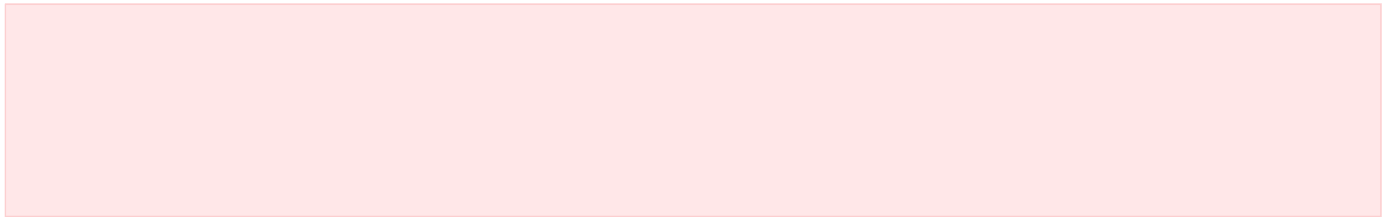
// Load remote module script
this._info.AddRemoteScriptDependency("https://example.com/api-module.js", "module");
```

```
<!-- Classic script -->
<script src="https://example.com/api.js"></script>

<!-- Module script -->
<script src="https://example.com/api-module.js" type="module"></script>
```

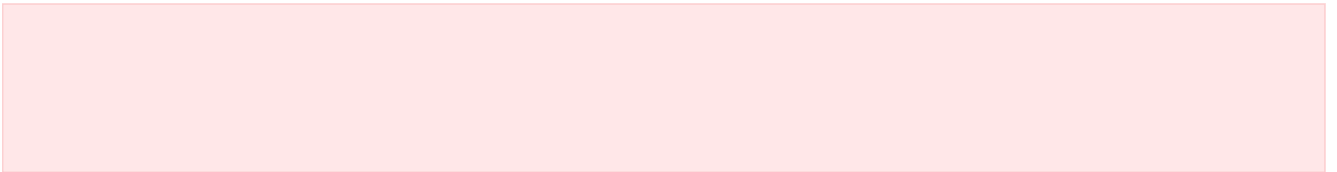
`http:``//example.com/api.js`

-----



-----

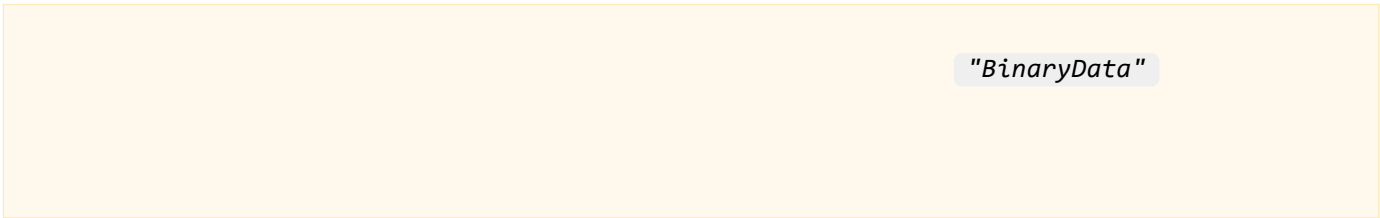
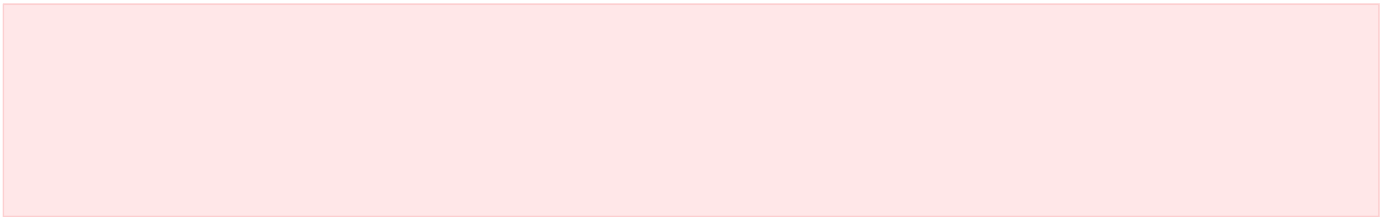
-----



-----

-----

---



---

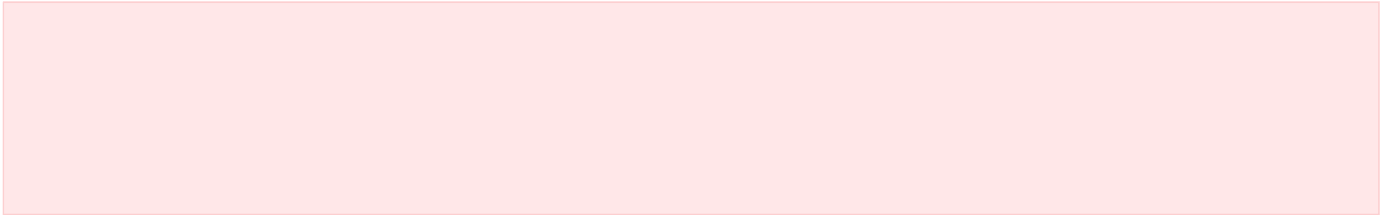
`viewOrBuffer`

---

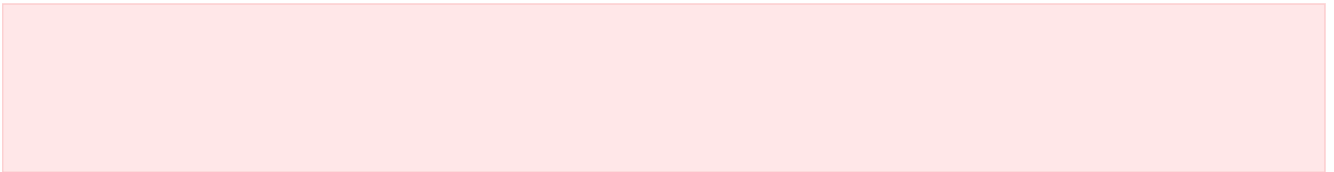
---

---

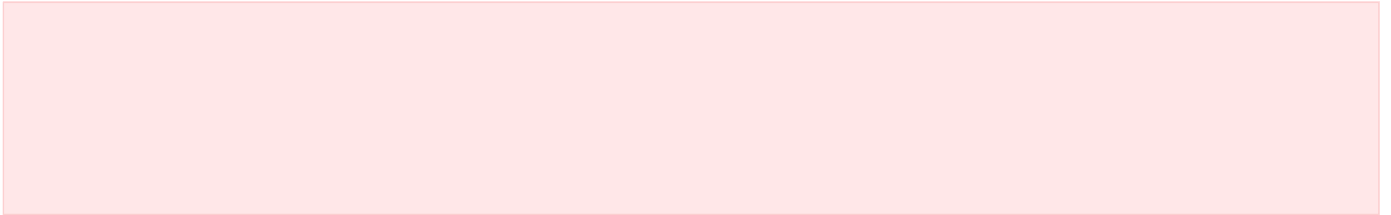
-----



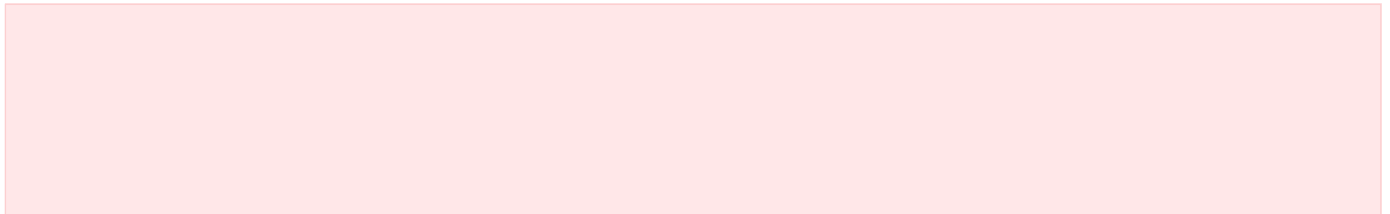
-----



---



---



ObjectClass

GetSingleGlobalObjectClassByCtor(C3.Plugins.Facebook)

GetSingleGlobalObjectClassByCtor()

null

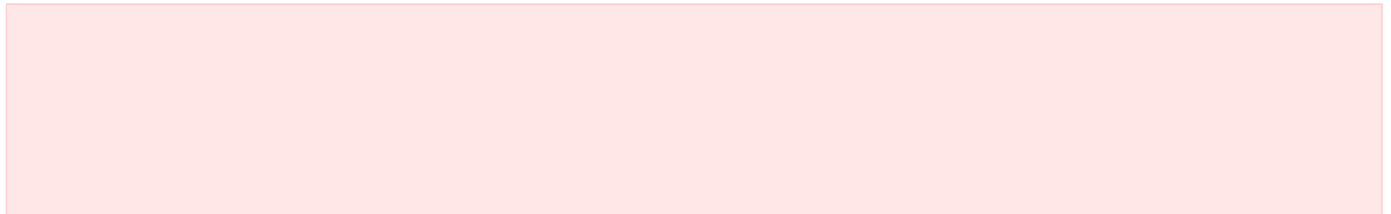
```
const facebookObjectClass = this._runtime.GetSingleGlobalObjectClassByCtor(C3.Plugins.Facebook);
if (facebookObjectClass)
{
    const facebookSdkInst = facebookObjectClass.GetSingleGlobalInstance().GetSdkInstance();
    // can now make calls, e.g.: facebookSdkInst.GetAccessToken()
}
```

---

```
if (self.c2_callFunction)
    self.c2_callFunction("name", ["param1", "param2"]);
```

```
c3_callFunction
```





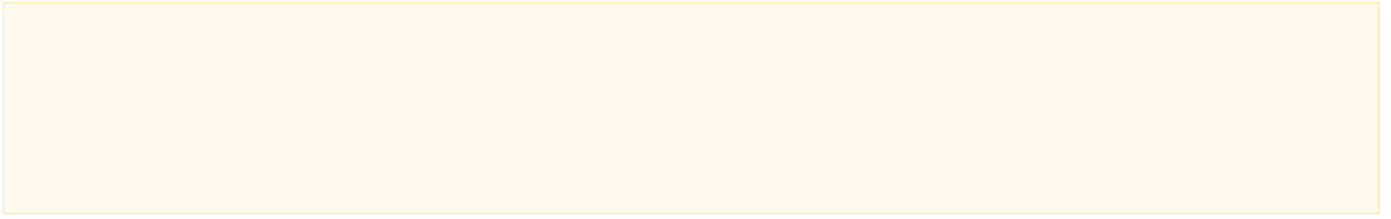
- 
- 

```
C3.Plugins.Tilemap.TILE_FLIPPED_HORIZONTAL = -0x80000000;  
C3.Plugins.Tilemap.TILE_FLIPPED_VERTICAL = 0x40000000;  
C3.Plugins.Tilemap.TILE_FLIPPED_DIAGONAL = 0x20000000;  
C3.Plugins.Tilemap.TILE_FLAGS_MASK = 0xE0000000;  
C3.Plugins.Tilemap.TILE_ID_MASK = 0x1FFFFFFF;
```

```
2 | C3.Plugins.Tilemap.TILE_FLIPPED_HORIZONTAL  
tile & C3.Plugins.Tilemap.TILE_FLIPPED_HORIZONTAL
```

```
tile &
```

```
C3.Plugins.Tilemap.TILE_ID_MASK
```



-----

-----

-----

-----

---

DOMElementHandler

*document*

---

PostToDOMElement()/PostToDOMElementAsync()

handler

func

(elem, e)

data

PostToDOMElementAsync()

func

"click"

AddElementMessageHandler()

handler

elementId

CreateElement()

data

elementId

---

---

`GetElementState()`

---

DOMHandler

document

---

async  
AddRuntimeMessageHandlers

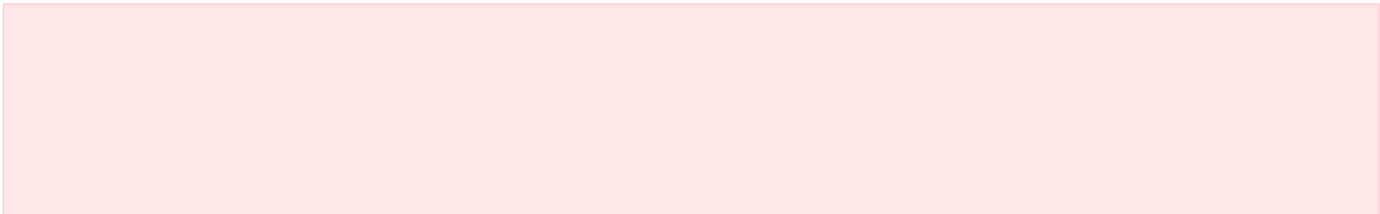
PostToDOMAsync

[handler, callback]

---

data

---



SDKBehaviorBase

SDKBehaviorBase

---

---

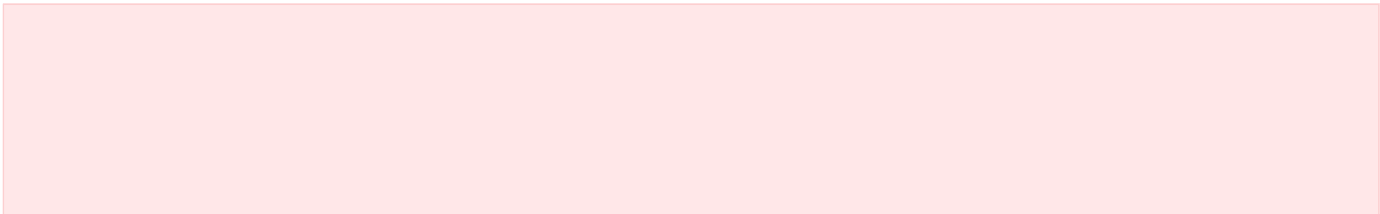
this.\_runtime

---

---

---

-----



SDKBehaviorInstance

SDKBehaviorInstance

-----

-----

-----

-----

-----

---

---

`this._behInst`

---

`this._inst`

---

`this._inst.GetWorldInfo()`

`null`

---

`this._runtime`

---

`this._objectClass`

---

`this._behaviorType`

---

`this._sdkType`

---

---

---



`this.Trigger(C3.Behaviors.Bullet.Cnds.OnStep)`

---

`Tick()` `Tick2()` `PostTick()`

---

`_StartTicking()`

---

`_StartTicking2()`

---

`Tick()`

`Tick()`

`PostTick()`

---

SaveToJson()

```
this.CallAction(C3.Behaviors.MyAddon.Acts.MyAction, "foo", "bar")
```

```
const value =  
this.CallExpression(C3.Behaviors.MyAddon.Exps.MyExpression)
```

name

cancelable

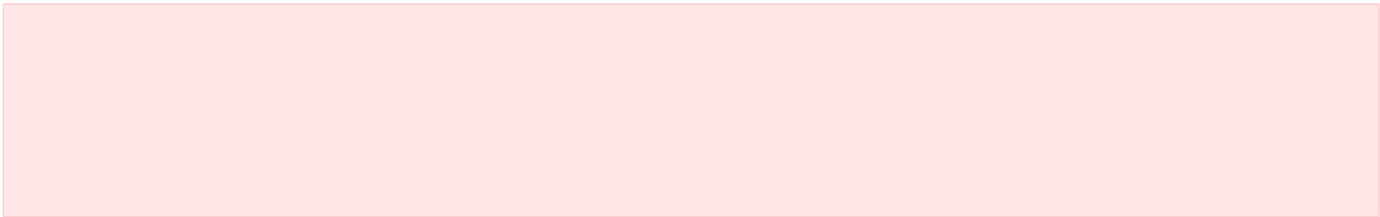
dispatchEvent()

preventDefault()

`additionalProperties`

```
DispatchScriptEvent("myevent", false, { extraProperty: 5 })
```

---



SDKBehaviorTypeBase

SDKBehaviorTypeBase

---

---

---

---

---

---

`this._objectClass`

---

`this._behaviorType`

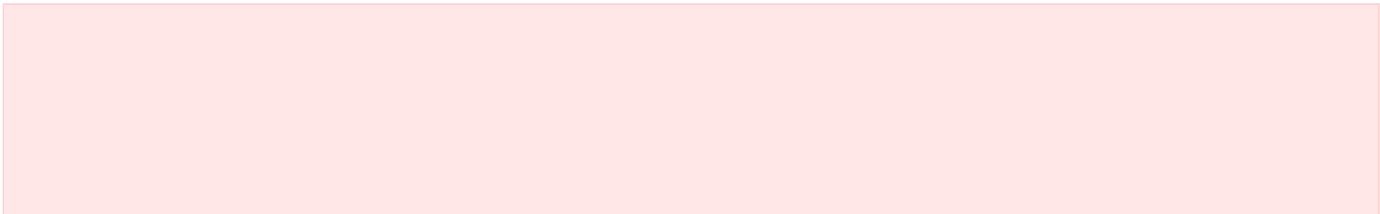
---

`this._runtime`

---

`this._behavior`

---



SDKDOMInstanceBase

SDKDOMInstanceBase

---

AddDOMELEMENTMessageHandler()

data

handler

---

CreateElement()

data

---

focus()

blur()

---

```
SetElementCSSStyle("font-family", "sans-serif")
```

```
elem.style.fontFamily = "sans-serif"
```

---

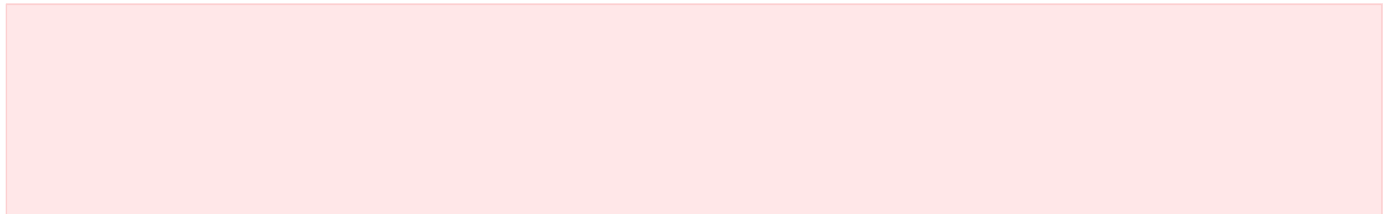
```
CreateElement()
```

```
UpdateElementState()
```

---

```
GetElementState()
```

```
UpdateState(elem, e)
```

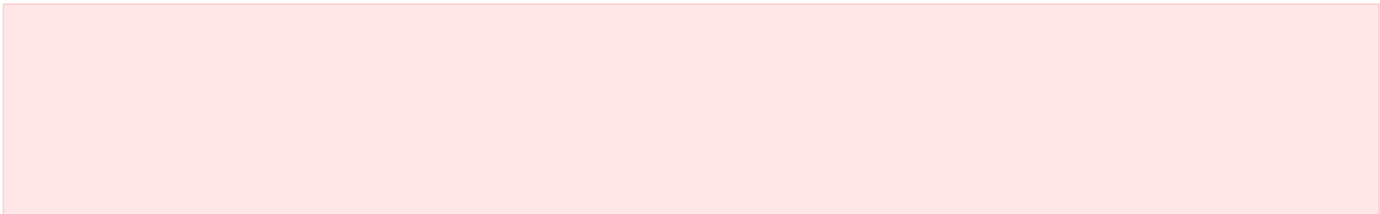
`SDKDOMPluginBase``DOM_COMPONENT_ID``SDKDOMPluginBase``SDKDOMPluginBase`

---

```
func
    handler
        PostToRuntimeElement()
        PostToRuntimeElement()
    PostToRuntimeElement()
    this.AddElementMessageHandler("click", (sdkInst, e) =>
sdkInst._OnClick(e));
```



-----



SDKInstanceBase

SDKInstanceBase

SDKInstanceBase

-----

-----

-----

-----

-----

-----

`this._inst`

---

`this._runtime`

---

`this._objectClass`

---

`this._sdkType`

---

`this.Trigger(C3.Plugins.Sprite.Cnds.OnAnimFinished)`

---

`async`

`AddDOMMessageHandlers`

`PostToRuntimeAsync`

`[handler, callback]`

---

`data`

---

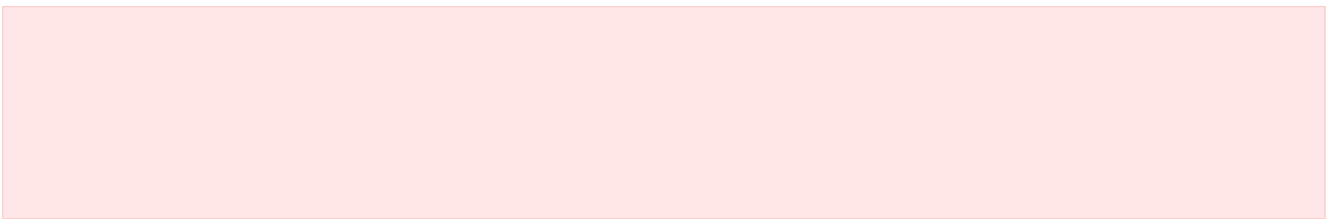
Tick()

Tick2()

-----  
\_StartTicking()

-----  
\_StartTicking2()

-----  
SaveToJson()

-----  
  
this.CallAction(C3.Plugins.MyAddon.Acts.MyAction, "foo", "bar")

---

```
const value =  
this.CallExpression(C3.Plugins.MyAddon.Exps.MyExpression)
```

---

---

```
name  
preventDefault()  
cancelable  
dispatchEvent()  
additionalProperties  
DispatchScriptEvent("myevent", false, {  
extraProperty: 5 })
```

---

---

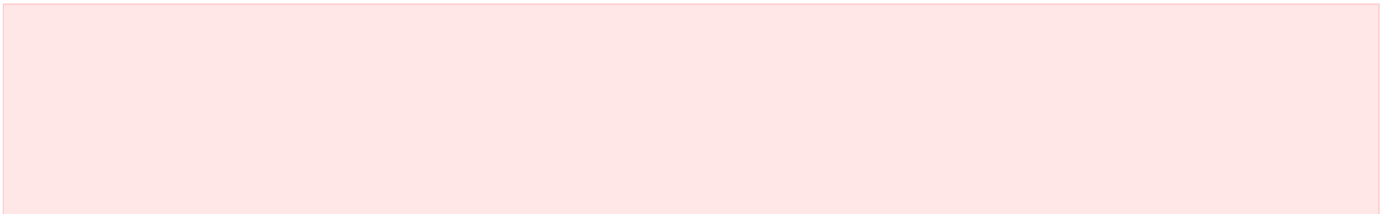


`callback]` `AddWrapperMessageHandlers` `[messageId,`



`params`

---



SDKPluginBase

SDKPluginBase

---

---

this.\_runtime

---

---

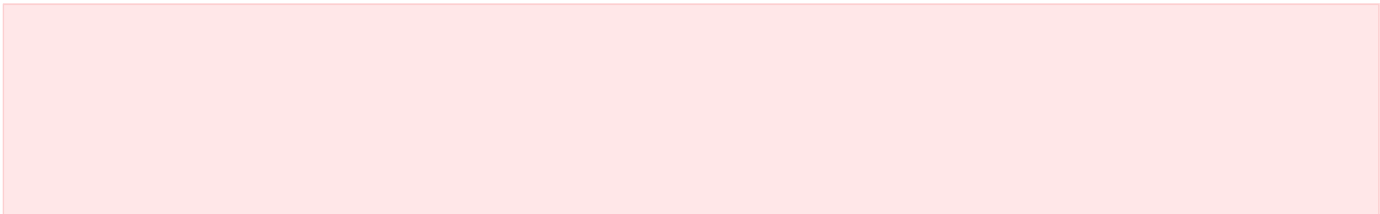
---

---

-----

-----

---



SDKTypeBase

SDKTypeBase

---

---

---

---

this.\_objectClass

---

this.\_runtime

---



`this._plugin`

---

---

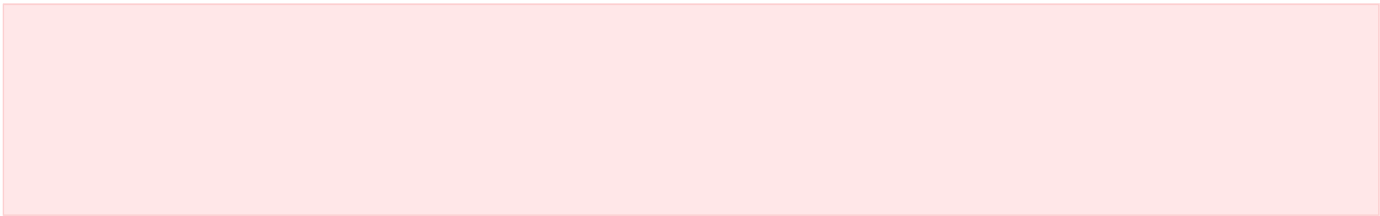
`renderer`

---

---

`LoadTextures()`

---



SDKWorldInstanceBase

SDKWorldInstanceBase

---

this.\_worldInfo

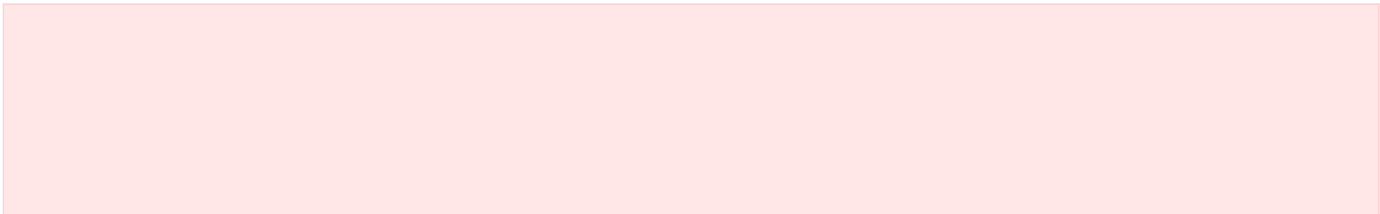
---

OnWebGLContextLost()

OnWebGLContextRestored()

---

```
HandleWebGLContextLoss()
```



EventBlock

Retrigger()



```
MyLoopingCondition()
{
    // Get necessary references
    const runtime = this._runtime;
    const eventSheetManager = runtime.GetEventSheetManager();
    const currentEvent = runtime.GetCurrentEvent();
    const solModifiers = currentEvent.GetSolModifiers();
    const eventStack = runtime.GetEventStack();

    // Get current stack frame and push new one
    const oldFrame = eventStack.GetCurrentStackFrame();
    const newFrame = eventStack.Push(currentEvent);

    for (const item of myArray)
    {
        // ... optionally update state here ...

        // Push a copy of the current SOL
        eventSheetManager.PushCopySol(solModifiers);

        // Retrigger the current event, running a single loop iteration
        currentEvent.Retrigger(oldFrame, newFrame);

        // Pop the current SOL
        eventSheetManager.PopSol(solModifiers);
    }

    // Pop the event stack frame
    eventStack.Pop();

    // Return false since event already executed
    return false;
}
```

null

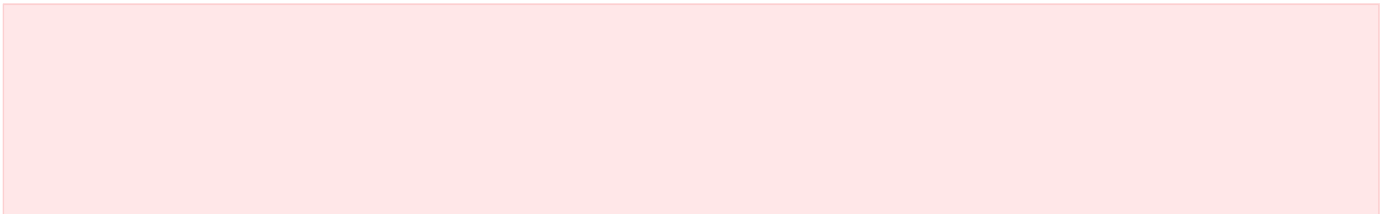
---

true

---

---

---



EventSheetManager

GetEventSheetManager()

---

---

---

GetEventStack().GetCurrentStackFrame()

---

GetCurrentEventStackFrame().GetCurrentEvent()

---

---

---

PushCleanSol

PushCopySol

PopSol

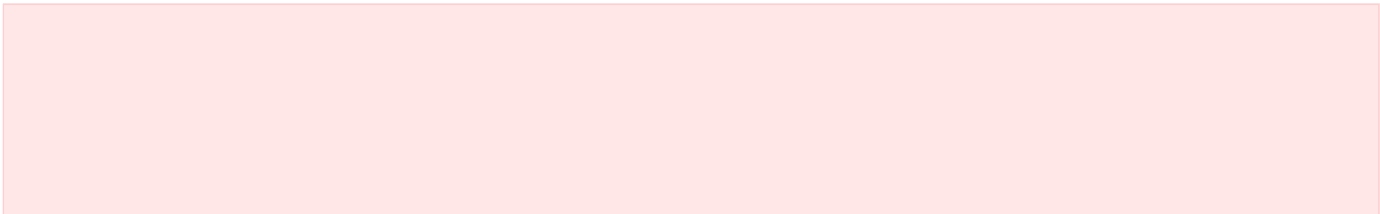
---

PushCleanSol

PushCopySol



---



EventStack

GetEventStack()

---

---

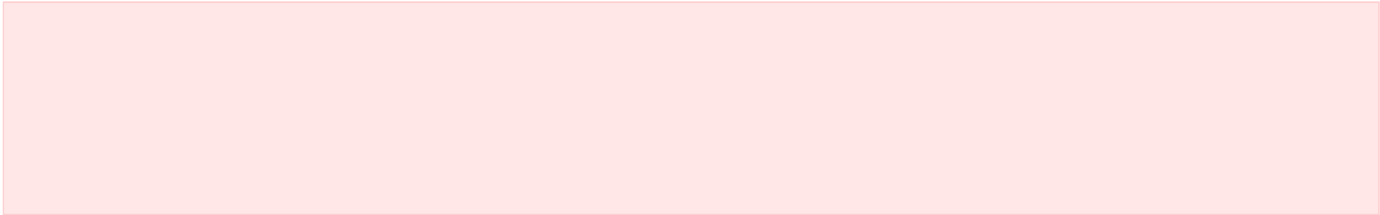
---

---

Pop

---

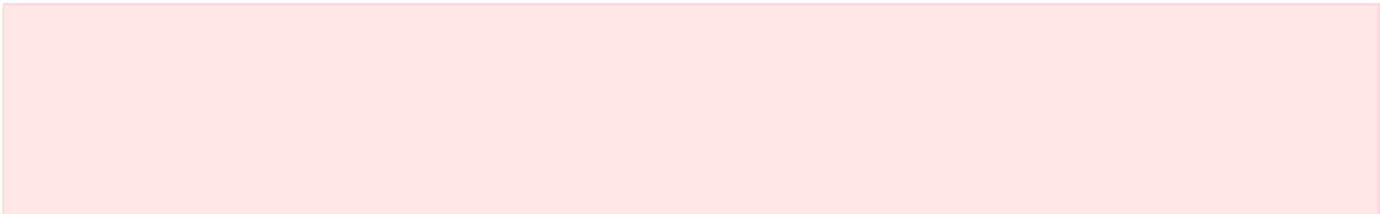
Push



EventStackFrame



---



AssetManager

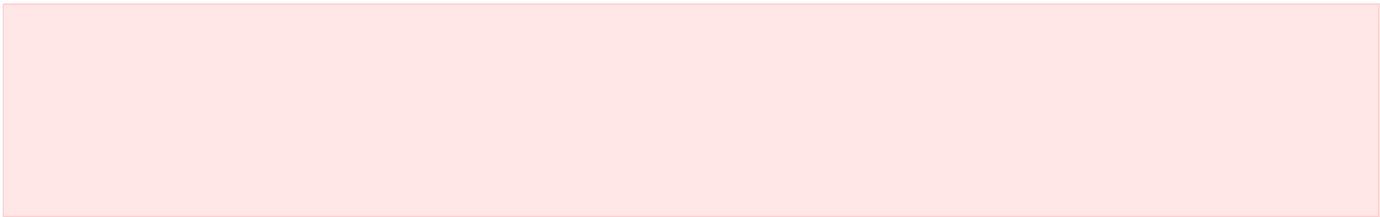
---

---

---

FetchBlob()

---



Layer





---

---

---

---

`inst`                      `inst`                      `otherInst`                      `isAfter`  
`inst`                      `otherInst`                      `inst`

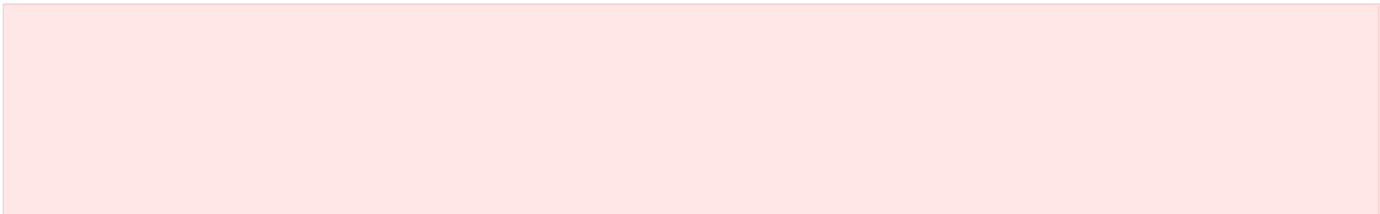
---

`[x, y]`

---

`[x, y]`

-----



Layout

-----

-----

-----

-----

-----

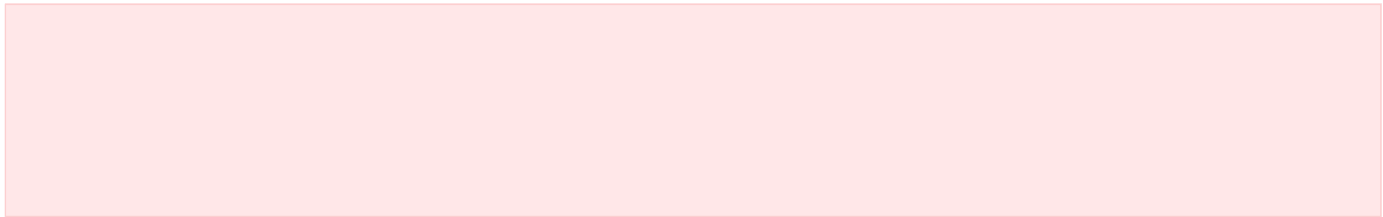
C3.toDegrees()

C3.toRadians()





GetLayer()



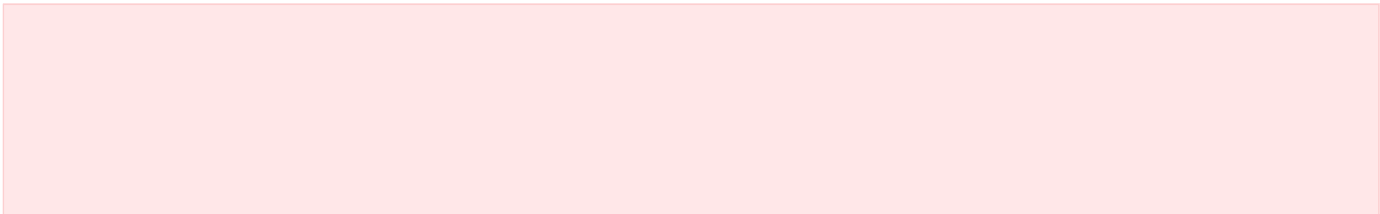
BehaviorInstance

BehaviorInstance

BehaviorInstance



-----



BehaviorType

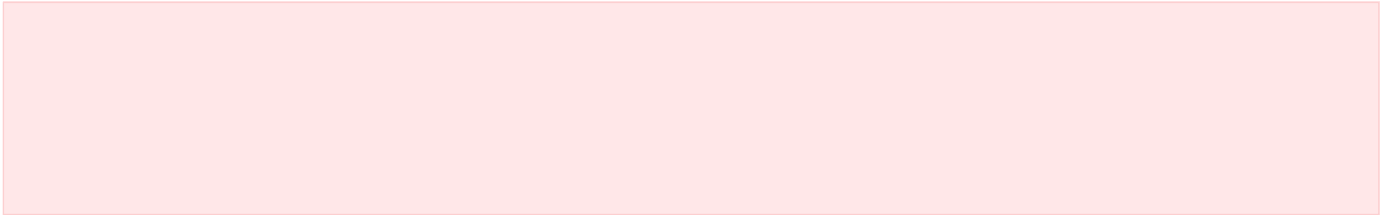
-----

-----

-----

-----

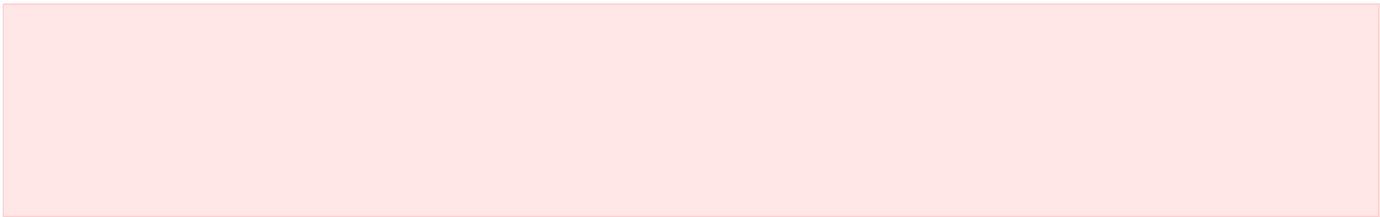
-----



Container



-----



Instance

-----

-----

-----

-----

-----

-----

`GetBehaviorInstanceFromCtor(C3.Behaviors.Platform)`

`null`

`GetBehaviorInstanceFromCtor(ctor)`

---

Map

*WeakMap*

*Instance*

---

null

---

null

---

allChildren()

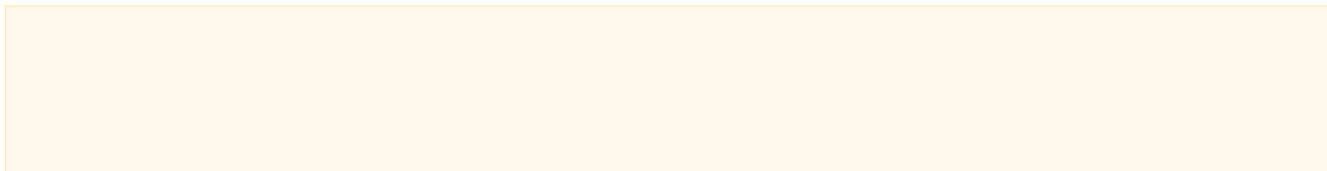
---

`opts`

- `transformX`
- `transformY`
- `transformWidth`
- `transformHeight`
- `transformAngle`
- `transformZElevation`
- `destroyWithParent`

`false`

`true`

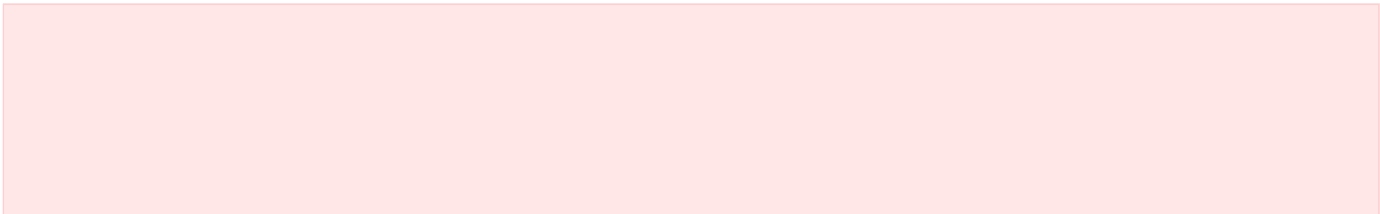


---

`AddChild()`



-----



ObjectClass

ObjectClass

-----

-----

-----

-----

-----

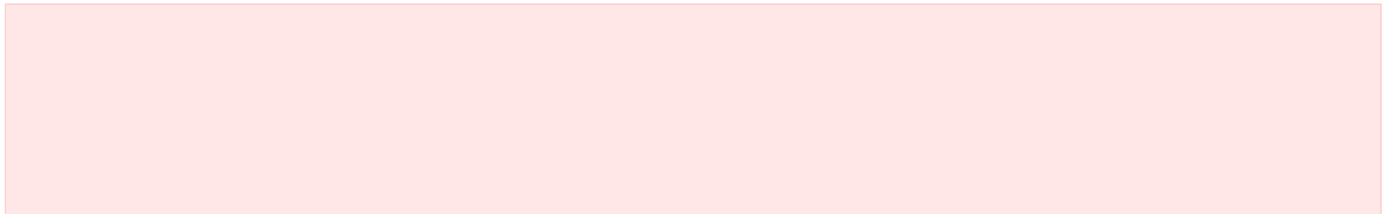
-----

ObjectClass

ObjectClass

null

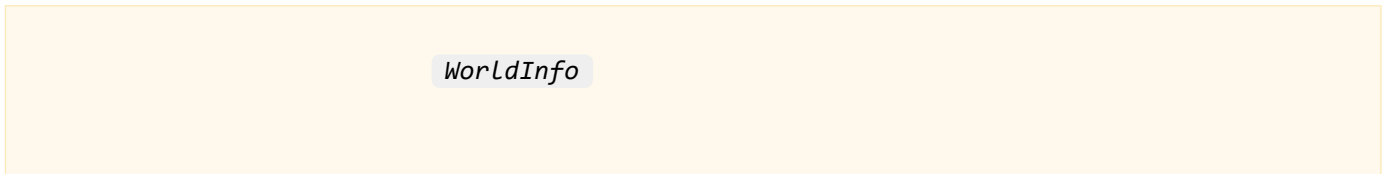




WorldInfo

WorldInfo

GetWorldInfo()



*WorldInfo*

SetBboxChanged()



WorldInfo



---

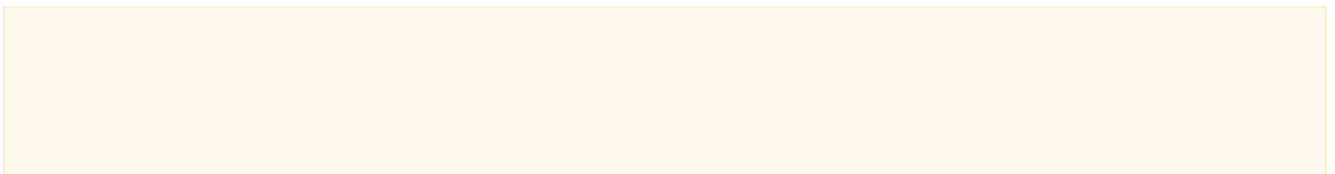
`SetX(GetX() + 5)`

`OffsetX(5)`

---

`C3.toDegrees()`

`C3.toRadians()`



true

false

---

---

---

---

---

---

---

---

opts

- mode "absolute" "relative"  
x y u v
- x y

- `u` `v`

`SetBboxChanged()`

`SetBboxChanged()`

-----

-----

-----

`otherInst`

`Instance`

`otherInst`

`WorldInfo`

`isOnTop`

`otherInst`



---

Runtime

runtime.Dispatcher()

runtime.Dispatcher().addEventListener("instancedestroy", callback)

---

instance

---

LoadFromJson

LoadFromJson()

GetInstanceByUID()

---

"beforefirstlayoutstart"

"beforelayoutstart"

"afterlayoutstart"

"beforefirstlayoutstart"

"afterfirstlayoutstart"



C3.Plugins.Facebook

null

null

GetEventSheetManager().GetEventStack()

GetEventSheetManager().GetCurrentEventStackFrame()

GetEventSheetManager().GetCurrentEvent()

"nearest"

"bilinear"

"trilinear"

